
Amazon Elastic Compute Cloud

User Guide for Linux Instances



Amazon Elastic Compute Cloud: User Guide for Linux Instances

Copyright © 2020 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is Amazon EC2	1
Features of Amazon EC2	1
How to get started with Amazon EC2	1
Related services	2
Accessing Amazon EC2	3
Pricing for Amazon EC2	3
PCI DSS compliance	4
Instances and AMIs	4
Instances	5
AMIs	6
Regions and Zones	7
Regions	7
Availability Zones	12
Local Zones	14
Wavelength Zones	17
AWS Outposts	19
Root device volume	20
Root device storage concepts	20
Choosing an AMI by root device type	22
Determining the root device type of your instance	22
Changing the root volume to persist	23
Setting up	26
Sign up for AWS	26
Create a key pair	26
Create a security group	27
Getting started tutorial	30
Overview	30
Prerequisites	31
Step 1: Launch an instance	31
Step 2: Connect to your instance	32
Step 3: Clean up your instance	32
Next steps	33
Best practices	34
Tutorials	36
Install a LAMP server (Amazon Linux 2)	36
Step 1: Prepare the LAMP server	37
Step 2: Test your LAMP server	40
Step 3: Secure the database server	41
Step 4: (Optional) Install phpMyAdmin	42
Troubleshooting	45
Related topics	45
Install a LAMP server (Amazon Linux AMI)	46
Step 1: Prepare the LAMP server	46
Step 2: Test your Lamp server	50
Step 3: Secure the database server	51
Step 4: (Optional) Install phpMyAdmin	52
Troubleshooting	55
Related topics	56
Tutorial: Hosting a WordPress blog	56
Prerequisites	57
Install WordPress	57
Next steps	64
Help! My public DNS name changed and now my blog is broken	64
Tutorial: Configure SSL/TLS on Amazon Linux 2	65

Prerequisites	66
Step 1: Enable TLS on the server	66
Step 2: Obtain a CA-signed certificate	68
Step 3: Test and harden the security configuration	73
Troubleshooting	75
Certificate automation: Let's Encrypt with Certbot on Amazon Linux 2	76
Tutorial: Configure SSL/TLS on Amazon Linux	80
Prerequisites	80
Step 1: Enable TLS on the server	81
Step 2: Obtain a CA-signed certificate	82
Step 3: Test and harden the security configuration	87
Troubleshooting	89
Certificate automation: Let's Encrypt with Certbot on Amazon Linux	89
Tutorial: Increase the availability of your application	93
Prerequisites	94
Scale and load balance your application	94
Test your load balancer	96
Amazon Machine Images	97
Using an AMI	97
Creating your own AMI	98
Buying, sharing, and selling AMIs	98
Deregistering your AMI	99
Amazon Linux 2 and Amazon Linux AMI	99
AMI types	99
Launch permissions	99
Storage for the root device	100
Virtualization types	102
Finding a Linux AMI	104
Finding a Linux AMI using the Amazon EC2 console	104
Finding an AMI using the AWS CLI	105
Finding the latest Amazon Linux AMI using Systems Manager	105
Using a Systems Manager parameter to find an AMI	106
Finding a Quick Start AMI	108
Shared AMIs	109
Finding shared AMIs	110
Making an AMI public	112
Sharing an AMI with specific AWS accounts	113
Using bookmarks	115
Guidelines for shared Linux AMIs	115
Paid AMIs	119
Selling your AMI	120
Finding a paid AMI	120
Purchasing a paid AMI	121
Getting the product code for your instance	121
Using paid support	122
Bills for paid and supported AMIs	122
Managing your AWS Marketplace subscriptions	122
Creating an Amazon EBS-backed Linux AMI	123
Overview of creating Amazon EBS-backed AMIs	123
Creating a Linux AMI from an instance	124
Creating a Linux AMI from a snapshot	126
Launching an instance from an AMI you created	126
Creating an instance store-backed Linux AMI	127
Overview of the creation process for instance store-backed AMIs	127
Prerequisites	128
Setting up the AMI tools	128
Creating an AMI from an instance store-backed instance	131

Converting to an Amazon EBS-Backed AMI	138
AMI tools reference	141
Using encryption with EBS-backed AMIs	157
Instance-launching scenarios	158
Image-copying scenarios	161
Copying an AMI	163
Permissions for copying an instance store-backed AMI	164
Cross-Region copying	165
Cross-account copying	166
Encryption and copying	166
Copying an AMI	167
Stopping a pending AMI copy operation	168
Obtaining billing information	169
AMI billing information fields	169
Platform details and usage operation values	169
Viewing platform details and usage operation values	170
Confirm billing information on your bill	171
Deregistering your Linux AMI	172
Cleaning up your Amazon EBS-backed AMI	172
Cleaning up your instance store-backed AMI	174
Amazon Linux	175
Amazon Linux availability	175
Connecting to an Amazon Linux instance	176
Identifying Amazon Linux images	176
AWS command line tools	177
Package repository	178
Extras library (Amazon Linux 2)	180
Accessing source packages for reference	181
cloud-init	181
Subscribing to Amazon Linux notifications	183
Running Amazon Linux 2 on premises	184
Kernel Live Patching	188
User provided kernels	193
HVM AMIs (GRUB)	193
Paravirtual AMIs (PV-GRUB)	193
Using the MATE desktop environment	198
Instances	200
Instance types	200
Available instance types	201
Hardware specifications	204
AMI virtualization types	204
Instances built on the Nitro System	205
Networking and storage features	205
Instance limits	208
General purpose	208
Compute optimized	253
Memory optimized	260
Storage optimized	272
Accelerated computing	278
Finding an instance type	293
Changing the instance type	294
Getting recommendations	300
Instance purchasing options	303
Determining the instance lifecycle	304
On-Demand Instances	305
Reserved Instances	308
Scheduled Instances	347

Spot Instances	351
Dedicated Hosts	438
Dedicated Instances	469
On-Demand Capacity Reservations	474
Instance lifecycle	493
Instance launch	494
Instance stop and start (Amazon EBS-backed instances only)	494
Instance hibernate (Amazon EBS-backed instances only)	495
Instance reboot	495
Instance retirement	496
Instance termination	496
Differences between reboot, stop, hibernate, and terminate	496
Launch	497
Connect	563
Stop and start	587
Hibernate	591
Reboot	603
Retire	604
Terminate	607
Recover	612
Configure instances	613
Common configuration scenarios	614
Managing software	614
Managing users	620
Processor state control	622
Setting the time	628
Optimizing CPU options	632
Changing the hostname	648
Setting up dynamic DNS	651
Running commands at launch	653
Instance metadata and user data	658
Elastic Inference	691
Identify instances	692
Inspecting the instance identity document	692
Inspecting the system UUID	692
Monitoring	694
Automated and manual monitoring	695
Automated monitoring tools	695
Manual monitoring tools	696
Best practices for monitoring	696
Monitoring the status of your instances	697
Instance status checks	697
Scheduled events	704
Monitoring your instances using CloudWatch	715
Enable detailed monitoring	715
List available metrics	717
Get statistics for metrics	728
Graph metrics	736
Create an alarm	736
Create alarms that stop, terminate, reboot, or recover an instance	738
Automating Amazon EC2 with CloudWatch Events	747
Monitoring memory and disk metrics	747
Collecting metrics using the CloudWatch agent	748
Deprecated: Collecting metrics using the CloudWatch monitoring scripts	748
Logging API calls with AWS CloudTrail	755
Amazon EC2 and Amazon EBS information in CloudTrail	755
Understanding Amazon EC2 and Amazon EBS log file entries	756

Auditing users that connect via EC2 Instance Connect	757
Networking	759
Instance IP addressing	759
Private IPv4 addresses and internal DNS hostnames	759
Public IPv4 addresses and external DNS hostnames	760
Elastic IP addresses (IPv4)	761
Amazon DNS server	761
IPv6 addresses	761
Working with the IPv4 addresses for your instances	762
Working with the IPv6 addresses for your instances	765
Multiple IP addresses	767
Bring your own IP addresses	775
Requirements	775
Prepare to bring your address range to your AWS account	776
Provision the address range for use with AWS	778
Advertise the address range through AWS	779
Work with your address range	779
Deprovision the address range	780
Elastic IP addresses	781
Elastic IP address basics	781
Working with Elastic IP addresses	782
Using reverse DNS for email applications	788
Elastic IP address limit	788
Network interfaces	789
Network interface basics	789
IP addresses per network interface per instance type	791
Working with network interfaces	802
Scenarios for network interfaces	808
Best practices for configuring network interfaces	810
Requester-managed network interfaces	812
Enhanced networking	813
Enhanced networking support	813
Enabling enhanced networking on your instance	813
Enhanced networking: ENA	814
Enhanced networking: Intel 82599 VF	826
Troubleshooting ENA	832
Elastic Fabric Adapter	838
EFA basics	839
Supported interfaces and libraries	840
Supported instance types	840
Supported AMIs	840
EFA limitations	841
Getting started with EFA and MPI	841
Getting started with EFA and NCCL	849
Working with EFA	865
Monitoring an EFA	867
Verifying the EFA installer using a checksum	868
Placement groups	869
Cluster placement groups	870
Partition placement groups	871
Spread placement groups	871
Placement group rules and limitations	872
Creating a placement group	873
Tagging a placement group	874
Launching instances in a placement group	876
Describing instances in a placement group	877
Changing the placement group for an instance	879

Deleting a placement group	880
Network MTU	881
Jumbo frames (9001 MTU)	881
Path MTU Discovery	882
Check the path MTU between two hosts	882
Check and set the MTU on your Linux instance	883
Troubleshooting	883
Virtual private clouds	884
Amazon VPC documentation	884
EC2-Classic	884
Detecting supported platforms	884
Instance types available in EC2-Classic	886
Differences between instances in EC2-Classic and a VPC	886
Sharing and accessing resources between EC2-Classic and a VPC	891
ClassicLink	892
Migrating from EC2-Classic to a VPC	904
Security	914
Infrastructure security	914
Network isolation	915
Isolation on physical hosts	915
Controlling network traffic	915
Interface VPC endpoints	916
Create an interface VPC endpoint	916
Create an interface VPC endpoint policy	916
Resilience	917
Data protection	918
Encryption at rest	918
Encryption in transit	918
Identity and access management	919
Network access to your instance	919
Amazon EC2 permission attributes	919
IAM and Amazon EC2	920
IAM policies	921
IAM roles	974
Network access	983
Key pairs	985
Creating or importing a key pair	986
Tagging a key pair	989
Retrieving the public key for your key pair	991
Retrieving the public key for your key pair through instance metadata	991
Locating the public key on an instance	992
Identifying the key pair that was specified at launch	992
(Optional) Verifying your key pair's fingerprint	993
Adding or replacing a key pair for your instance	993
Connecting to your Linux instance if you lose your private key	994
Deleting your key pair	998
Security groups	999
Security group rules	1000
Default security groups	1003
Custom security groups	1003
Working with security groups	1004
Security group rules reference	1011
Update management	1017
Compliance validation	1017
Storage	1018
Amazon EBS	1019
Features of Amazon EBS	1020

EBS volumes	1021
EBS snapshots	1060
EBS data services	1109
EBS volumes and NVMe	1136
EBS optimization	1139
EBS performance	1156
EBS CloudWatch metrics	1172
EBS CloudWatch events	1178
EBS quotas	1188
Instance store	1188
Instance store lifetime	1189
Instance store volumes	1190
Add instance store volumes	1196
SSD instance store volumes	1199
Instance store swap volumes	1201
Optimizing disk performance	1203
File storage	1204
Amazon S3	1204
Amazon EFS	1206
Instance volume limits	1209
Nitro System volume limits	1209
Linux-specific volume limits	1209
Bandwidth versus capacity	1210
Device naming	1210
Available device names	1210
Device name considerations	1211
Block device mapping	1211
Block device mapping concepts	1212
AMI block device mapping	1215
Instance block device mapping	1217
Resources and tags	1222
Resource locations	1222
Resource IDs	1223
Listing and filtering your resources	1224
Listing and filtering resources using the console	1224
Listing and filtering using the CLI and API	1227
Tagging your resources	1229
Tag basics	1230
Tagging your resources	1231
Tag restrictions	1233
Tagging your resources for billing	1234
Working with tags using the console	1234
Working with tags using the command line	1237
Adding tags to a resource using CloudFormation	1240
Service quotas	1241
Viewing your current limits	1241
Requesting an increase	1243
Limits on email sent using port 25	1243
Usage reports	1243
Troubleshooting	1244
Troubleshooting launch issues	1244
Instance limit exceeded	1244
Insufficient instance capacity	1245
Instance terminates immediately	1245
Connecting to your instance	1246
Common causes for connection issues	1247
Error connecting to your instance: Connection timed out	1247

Error: unable to load key ... Expecting: ANY PRIVATE KEY	1249
Error: User key not recognized by server	1250
Error: Permission denied or connection closed by [instance] port 22	1251
Error: Unprotected private key file	1252
Error: Private key must begin with "----BEGIN RSA PRIVATE KEY----" and end with "----END RSA PRIVATE KEY----"	1253
Error: Server refused our key or No supported authentication methods available	1253
Cannot ping instance	1253
Error: Server unexpectedly closed network connection	1254
Stopping your instance	1254
Creating a replacement instance	1254
Terminating your instance	1255
Delayed instance termination	1256
Terminated instance still displayed	1256
Instances automatically launched or terminated	1256
Failed status checks	1256
Review status check information	1257
Retrieve the system logs	1257
Troubleshooting system log errors for Linux-based instances	1258
Out of memory: kill process	1259
ERROR: mmu_update failed (Memory management update failed)	1260
I/O error (block device failure)	1260
I/O ERROR: neither local nor remote disk (Broken distributed block device)	1261
request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)	1262
"FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" (Kernel and AMI mismatch)	1263
"FATAL: Could not load /lib/modules" or "BusyBox" (Missing kernel modules)	1263
ERROR Invalid kernel (EC2 incompatible kernel)	1265
fsck: No such file or directory while trying to open... (File system not found)	1266
General error mounting filesystems (failed mount)	1267
VFS: Unable to mount root fs on unknown-block (Root filesystem mismatch)	1269
Error: Unable to determine major/minor number of root device... (Root file system/device mismatch)	1270
XENBUS: Device with no driver..	1271
... days without being checked, check forced (File system check required)	1271
fsck died with exit status... (Missing device)	1272
GRUB prompt (grubdom>)	1273
Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address)	1275
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration)	1276
XENBUS: Timeout connecting to devices (Xenbus timeout)	1277
Troubleshooting an unreachable instance	1277
Instance reboot	1278
Instance console output	1278
Capture a screenshot of an unreachable instance	1278
Instance recovery when a host computer fails	1279
Booting from the wrong volume	1280
EC2Rescue for Linux	1281
Installing EC2Rescue for Linux	1281
(Optional) Verify the signature of EC2Rescue for Linux	1282
Working with EC2Rescue for Linux	1285
Developing EC2Rescue modules	1287
Sending a diagnostic interrupt	1291
Supported instance types	1291
Prerequisites	1292

Sending a diagnostic interrupt	1294
Document history	1295
History for previous years	1300

What is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

For more information about cloud computing, see [What is Cloud Computing?](#)

Features of Amazon EC2

Amazon EC2 provides the following features:

- Virtual computing environments, known as *instances*
- Preconfigured templates for your instances, known as *Amazon Machine Images (AMIs)*, that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as *instance types*
- Secure login information for your instances using *key pairs* (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as *instance store volumes*
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as *Amazon EBS volumes*
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as *Regions and Availability Zones*
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using *security groups*
- Static IPv4 addresses for dynamic cloud computing, known as *Elastic IP addresses*
- Metadata, known as *tags*, that you can create and assign to your Amazon EC2 resources
- Virtual networks you can create that are logically isolated from the rest of the AWS Cloud, and that you can optionally connect to your own network, known as *virtual private clouds (VPCs)*

For more information about the features of Amazon EC2, see the [Amazon EC2 product page](#).

For more information about running your website on AWS, see [Web Hosting](#).

How to get started with Amazon EC2

First, you need to get set up to use Amazon EC2. After you are set up, you are ready to complete the Getting Started tutorial for Amazon EC2. Whenever you need more information about an Amazon EC2 feature, you can read the technical documentation.

Get up and running

- [Setting up with Amazon EC2 \(p. 26\)](#)
- [Tutorial: Getting started with Amazon EC2 Linux instances \(p. 30\)](#)

Basics

- [Instances and AMIs \(p. 4\)](#)
- [Regions and Zones \(p. 7\)](#)
- [Instance types \(p. 200\)](#)
- [Tags \(p. 1229\)](#)

Networking and security

- [Key pairs \(p. 985\)](#)
- [Security groups \(p. 999\)](#)
- [Elastic IP addresses \(p. 781\)](#)
- [Virtual private clouds \(p. 884\)](#)

Storage

- [Amazon EBS \(p. 1019\)](#)
- [Instance store \(p. 1188\)](#)

Working with Linux instances

- [AWS Systems Manager Run Command](#) in the [AWS Systems Manager User Guide](#)
- [Tutorial: Install a LAMP web server on Amazon Linux 2 \(p. 36\)](#)
- [Tutorial: Configure SSL/TLS on Amazon Linux 2 \(p. 65\)](#)

If you have questions about whether AWS is right for you, [contact AWS Sales](#). If you have technical questions about Amazon EC2, use the [Amazon EC2 forum](#).

Related services

You can provision Amazon EC2 resources, such as instances and volumes, directly using Amazon EC2. You can also provision Amazon EC2 resources using other services in AWS. For more information, see the following documentation:

- [Amazon EC2 Auto Scaling User Guide](#)
- [AWS CloudFormation User Guide](#)
- [AWS Elastic Beanstalk Developer Guide](#)
- [AWS OpsWorks User Guide](#)

To automatically distribute incoming application traffic across multiple instances, use Elastic Load Balancing. For more information, see the [Elastic Load Balancing User Guide](#).

To get a managed relational database in the cloud, use Amazon Relational Database Service (Amazon RDS) to launch a database instance. Although you can set up a database on an EC2 instance, Amazon

RDS offers the advantage of handling your database management tasks, such as patching the software, backing up, and storing the backups. For more information, see the [Amazon Relational Database Service Developer Guide](#).

To make it easier to manage Docker containers on a cluster of EC2 instances, use Amazon Elastic Container Service (Amazon ECS). For more information, see the [Amazon Elastic Container Service Developer Guide](#) or the [Amazon Elastic Container Service User Guide for AWS Fargate](#).

To monitor basic statistics for your instances and Amazon EBS volumes, use Amazon CloudWatch. For more information, see the [Amazon CloudWatch User Guide](#). To detect potentially authorized or malicious use of your EC2 instances, use Amazon GuardDuty. For more information see the [Amazon GuardDuty User Guide](#).

Accessing Amazon EC2

Amazon EC2 provides a web-based user interface, the Amazon EC2 console. If you've signed up for an AWS account, you can access the Amazon EC2 console by signing into the AWS Management Console and selecting **EC2** from the console home page.

If you prefer to use a command line interface, you have the following options:

AWS Command Line Interface (CLI)

Provides commands for a broad set of AWS products, and is supported on Windows, Mac, and Linux. To get started, see [AWS Command Line Interface User Guide](#). For more information about the commands for Amazon EC2, see `ec2` in the [AWS CLI Command Reference](#).

AWS Tools for Windows PowerShell

Provides commands for a broad set of AWS products for those who script in the PowerShell environment. To get started, see the [AWS Tools for Windows PowerShell User Guide](#). For more information about the cmdlets for Amazon EC2, see the [AWS Tools for PowerShell Cmdlet Reference](#).

Amazon EC2 supports creating resources using AWS CloudFormation. You create a template, in JSON or YAML, that describes your AWS resources, and AWS CloudFormation provisions and configures those resources for you. You can reuse your CloudFormation templates to provision the same resources multiple times, whether in the same Region and account or in multiple Regions and accounts. For more information about the resource types and properties for Amazon EC2, see [EC2 resource type reference](#) in the [AWS CloudFormation User Guide](#).

Amazon EC2 provides a Query API. These requests are HTTP or HTTPS requests that use the HTTP verbs GET or POST and a Query parameter named `Action`. For more information about the API actions for Amazon EC2, see [Actions](#) in the [Amazon EC2 API Reference](#).

If you prefer to build applications using language-specific APIs instead of submitting a request over HTTP or HTTPS, AWS provides libraries, sample code, tutorials, and other resources for software developers. These libraries provide basic functions that automate tasks such as cryptographically signing your requests, retrying requests, and handling error responses, making it easier for you to get started. For more information, see [AWS SDKs and Tools](#).

Pricing for Amazon EC2

When you sign up for AWS, you can get started with Amazon EC2 for free using the [AWS Free Tier](#).

Amazon EC2 provides the following purchasing options for instances:

On-Demand Instances

Pay for the instances that you use by the second, with no long-term commitments or upfront payments.

Savings Plans

You can reduce your Amazon EC2 costs by making a commitment to a consistent amount of usage, in USD per hour, for a term of 1 or 3 years.

Reserved Instances

You can reduce your Amazon EC2 costs by making a commitment to a specific instance configuration, including instance type and Region, for a term of 1 or 3 years.

Spot Instances

Request unused EC2 instances, which can reduce your Amazon EC2 costs significantly.

For a complete list of charges and prices for Amazon EC2, see [Amazon EC2 Pricing](#).

To calculate the cost of a sample provisioned environment, see [Cloud Economics Center](#).

To see your bill, go to the **Billing and Cost Management Dashboard** in the [AWS Billing and Cost Management console](#). Your bill contains links to usage reports that provide details about your bill. To learn more about AWS account billing, see [AWS Account Billing](#).

If you have questions concerning AWS billing, accounts, and events, [contact AWS Support](#).

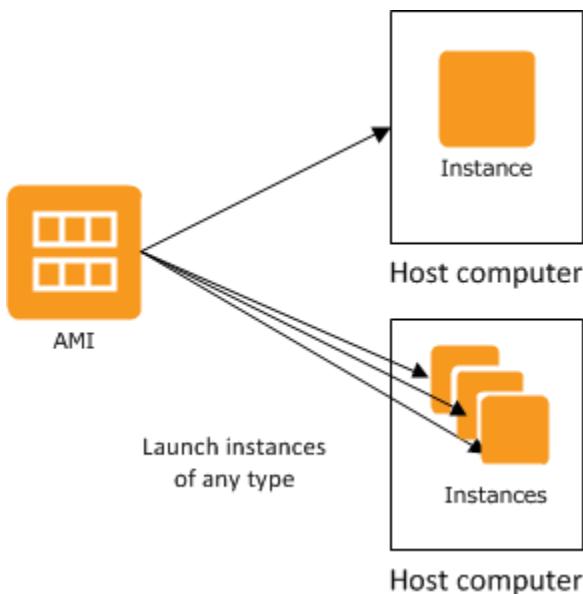
For an overview of Trusted Advisor, a service that helps you optimize the costs, security, and performance of your AWS environment, see [AWS Trusted Advisor](#).

PCI DSS compliance

Amazon EC2 supports the processing, storage, and transmission of credit card data by a merchant or service provider, and has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS). For more information about PCI DSS, including how to request a copy of the AWS PCI Compliance Package, see [PCI DSS Level 1](#).

Instances and AMIs

An *Amazon Machine Image (AMI)* is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch an *instance*, which is a copy of the AMI running as a virtual server in the cloud. You can launch multiple instances of an AMI, as shown in the following figure.



Your instances keep running until you stop or terminate them, or until they fail. If an instance fails, you can launch a new one from the AMI.

Instances

An instance is a virtual server in the cloud. Its configuration at launch is a copy of the AMI that you specified when you launched the instance.

You can launch different types of instances from a single AMI. An *instance type* essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory capabilities. Select an instance type based on the amount of memory and computing power that you need for the application or software that you plan to run on the instance. For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instance Types](#).

After you launch an instance, it looks like a traditional host, and you can interact with it as you would any computer. You have complete control of your instances; you can use **sudo** to run commands that require root privileges.

Your AWS account has a limit on the number of instances that you can have running. For more information about this limit, and how to request an increase, see [How many instances can I run in Amazon EC2](#) in the Amazon EC2 General FAQ.

Storage for your instance

The root device for your instance contains the image used to boot the instance. For more information, see [Amazon EC2 root device volume \(p. 20\)](#).

Your instance may include local storage volumes, known as instance store volumes, which you can configure at launch time with block device mapping. For more information, see [Block device mapping \(p. 1211\)](#). After these volumes have been added to and mapped on your instance, they are available for you to mount and use. If your instance fails, or if your instance is stopped or terminated, the data on these volumes is lost; therefore, these volumes are best used for temporary data. To keep important data safe, you should use a replication strategy across multiple instances, or store your persistent data in Amazon S3 or Amazon EBS volumes. For more information, see [Storage \(p. 1018\)](#).

Security best practices

- Use AWS Identity and Access Management (IAM) to control access to your AWS resources, including your instances. You can create IAM users and groups under your AWS account, assign security credentials to each, and control the access that each has to resources and services in AWS. For more information, see [Identity and access management for Amazon EC2 \(p. 919\)](#).
- Restrict access by only allowing trusted hosts or networks to access ports on your instance. For example, you can restrict SSH access by restricting incoming traffic on port 22. For more information, see [Amazon EC2 security groups for Linux instances \(p. 999\)](#).
- Review the rules in your security groups regularly, and ensure that you apply the principle of *least privilege*—only open up permissions that you require. You can also create different security groups to deal with instances that have different security requirements. Consider creating a bastion security group that allows external logins, and keep the remainder of your instances in a group that does not allow external logins.
- Disable password-based logins for instances launched from your AMI. Passwords can be found or cracked, and are a security risk. For more information, see [Disable password-based remote logins for root \(p. 116\)](#). For more information about sharing AMIs safely, see [Shared AMIs \(p. 109\)](#).

Stopping and terminating instances

You can stop or terminate a running instance at any time.

Stopping an instance

When an instance is stopped, the instance performs a normal shutdown, and then transitions to a stopped state. All of its Amazon EBS volumes remain attached, and you can start the instance again at a later time.

You are not charged for additional instance usage while the instance is in a stopped state. A minimum of one minute is charged for every transition from a stopped state to a running state. If the instance type was changed while the instance was stopped, you will be charged the rate for the new instance type after the instance is started. All of the associated Amazon EBS usage of your instance, including root device usage, is billed using typical Amazon EBS prices.

When an instance is in a stopped state, you can attach or detach Amazon EBS volumes. You can also create an AMI from the instance, and you can change the kernel, RAM disk, and instance type.

Terminating an instance

When an instance is terminated, the instance performs a normal shutdown. The root device volume is deleted by default, but any attached Amazon EBS volumes are preserved by default, determined by each volume's `deleteOnTermination` attribute setting. The instance itself is also deleted, and you can't start the instance again at a later time.

To prevent accidental termination, you can disable instance termination. If you do so, ensure that the `disableApiTermination` attribute is set to `true` for the instance. To control the behavior of an instance shutdown, such as `shutdown -h` in Linux or `shutdown` in Windows, set the `instanceInitiatedShutdownBehavior` instance attribute to `stop` or `terminate` as desired. Instances with Amazon EBS volumes for the root device default to `stop`, and instances with `instance-store` root devices are always terminated as the result of an instance shutdown.

For more information, see [Instance lifecycle \(p. 493\)](#).

AMIs

Amazon Web Services (AWS) publishes many [Amazon Machine Images \(AMIs\)](#) that contain common software configurations for public use. In addition, members of the AWS developer community have

published their own custom AMIs. You can also create your own custom AMI or AMIs; doing so enables you to quickly and easily start new instances that have everything you need. For example, if your application is a website or a web service, your AMI could include a web server, the associated static content, and the code for the dynamic pages. As a result, after you launch an instance from this AMI, your web server starts, and your application is ready to accept requests.

All AMIs are categorized as either *backed by Amazon EBS*, which means that the root device for an instance launched from the AMI is an Amazon EBS volume, or *backed by instance store*, which means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.

The description of an AMI indicates the type of root device (either `ebs` or `instance store`). This is important because there are significant differences in what you can do with each type of AMI. For more information about these differences, see [Storage for the root device \(p. 100\)](#).

You can deregister an AMI when you have finished using it. After you deregister an AMI, you can't use it to launch new instances. Existing instances launched from the AMI are not affected. Therefore, if you are also finished with the instances launched from these AMIs, you should terminate them.

Regions and Zones

Amazon EC2 is hosted in multiple locations world-wide. These locations are composed of Regions, Availability Zones, Local Zones, and Wavelength Zones. Each *Region* is a separate geographic area.

- Availability Zones are multiple, isolated locations within each Region.
- Local Zones provide you the ability to place resources, such as compute and storage, in multiple locations closer to your end users.
- AWS Outposts brings native AWS services, infrastructure, and operating models to virtually any data center, co-location space, or on-premises facility.
- Wavelength Zones allow developers to build applications that deliver ultra-low latencies to 5G devices and end users. Wavelength deploys standard AWS compute and storage services to the edge of telecommunication carriers' 5G networks.

AWS operates state-of-the-art, highly available data centers. Although rare, failures can occur that affect the availability of instances that are in the same location. If you host all of your instances in a single location that is affected by a failure, none of your instances would be available.

To help you determine which deployment is best for you, see [AWS Wavelength FAQs](#).

Contents

- [Regions \(p. 7\)](#)
- [Availability Zones \(p. 12\)](#)
- [Local Zones \(p. 14\)](#)
- [Wavelength Zones \(p. 17\)](#)
- [AWS Outposts \(p. 19\)](#)

Regions

Each Amazon EC2 Region is designed to be isolated from the other Amazon EC2 Regions. This achieves the greatest possible fault tolerance and stability.

The following diagram illustrates multiple AWS Regions.



When you view your resources, you see only the resources that are tied to the Region that you specified. This is because Regions are isolated from each other, and we don't automatically replicate resources across Regions.

When you launch an instance, you must select an AMI that's in the same Region. If the AMI is in another Region, you can copy the AMI to the Region you're using. For more information, see [Copying an AMI \(p. 163\)](#).

Note that there is a charge for data transfer between Regions. For more information, see [Amazon EC2 Pricing - Data Transfer](#).

Contents

- [Available Regions \(p. 8\)](#)
- [Regions and endpoints \(p. 9\)](#)
- [Describing your Regions \(p. 9\)](#)
- [Getting the Region name \(p. 11\)](#)
- [Specifying the Region for a resource \(p. 11\)](#)

Available Regions

Your account determines the Regions that are available to you. For example:

- An AWS account provides multiple Regions so that you can launch Amazon EC2 instances in locations that meet your requirements. For example, you might want to launch instances in Europe to be closer to your European customers or to meet legal requirements.
- An AWS GovCloud (US-West) account provides access to the AWS GovCloud (US-West) Region and the AWS GovCloud (US-East) Region. For more information, see [AWS GovCloud \(US\)](#).
- An Amazon AWS (China) account provides access to the Beijing and Ningxia Regions only. For more information, see [AWS in China](#).

The following table lists the Regions provided by an AWS account. You can't describe or access additional Regions from an AWS account, such as AWS GovCloud (US) Region or the China Regions. To use a Region introduced after March 20, 2019, you must enable the Region. For more information, see [Managing AWS Regions](#) in the *AWS General Reference*.

For information about available Wavelength Zones, see [Available Wavelength Zones](#) in the *AWS Wavelength Developer Guide*.

Code	Name	Opt-in Status	Local Zone
us-east-2	US East (Ohio)	Not required	Not available
us-east-1	US East (N. Virginia)	Not required	Not available
us-west-1	US West (N. California)	Not required	Not available
us-west-2	US West (Oregon)	Not required	us-west-2-lax-1a us-west-2-lax-1b

Code	Name	Opt-in Status	Local Zone
af-south-1	Africa (Cape Town)	Required	Not available
ap-east-1	Asia Pacific (Hong Kong)	Required	Not available
ap-south-1	Asia Pacific (Mumbai)	Not required	Not available
ap-northeast-3	Asia Pacific (Osaka-Local)	Not required	Not available
ap-northeast-2	Asia Pacific (Seoul)	Not required	Not available
ap-southeast-1	Asia Pacific (Singapore)	Not required	Not available
ap-southeast-2	Asia Pacific (Sydney)	Not required	Not available
ap-northeast-1	Asia Pacific (Tokyo)	Not required	Not available
ca-central-1	Canada (Central)	Not required	Not available
eu-central-1	Europe (Frankfurt)	Not required	Not available
eu-west-1	Europe (Ireland)	Not required	Not available
eu-west-2	Europe (London)	Not required	Not available
eu-south-1	Europe (Milan)	Required	Not available
eu-west-3	Europe (Paris)	Not required	Not available
eu-north-1	Europe (Stockholm)	Not required	Not available
me-south-1	Middle East (Bahrain)	Required	Not available
sa-east-1	South America (São Paulo)	Not required	Not available

For more information, see [AWS Global Infrastructure](#).

The number and mapping of Availability Zones per Region may vary between AWS accounts. To get a list of the Availability Zones that are available to your account, you can use the Amazon EC2 console or the command line interface. For more information, see [Describing your Regions \(p. 9\)](#).

Regions and endpoints

When you work with an instance using the command line interface or API actions, you must specify its Regional endpoint. For more information about the Regions and endpoints for Amazon EC2, see [Amazon EC2 endpoints and quotas](#) in the *Amazon Web Services General Reference*.

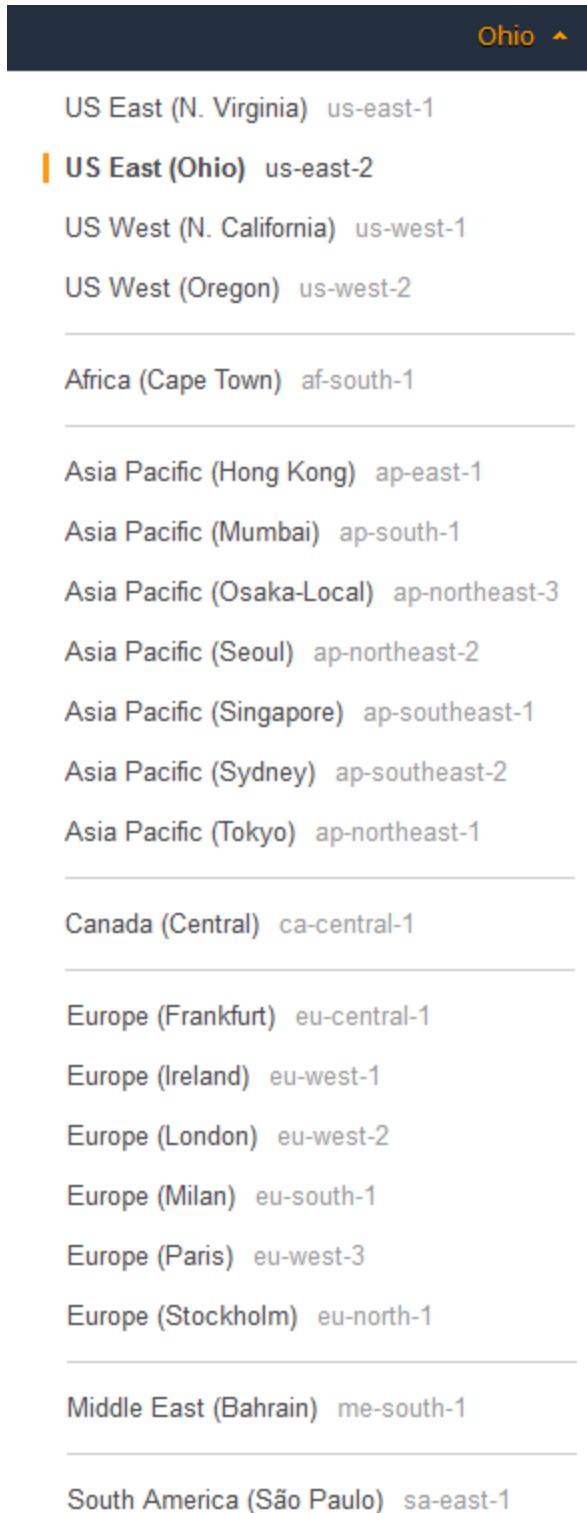
For more information about endpoints and protocols in AWS GovCloud (US-West), see [AWS GovCloud \(US-West\) Endpoints](#) in the *AWS GovCloud (US) User Guide*.

Describing your Regions

You can use the Amazon EC2 console or the command line interface to determine which Regions are available for your account. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

To find your Regions using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, view the options in the Region selector.



3. Your EC2 resources for this Region are displayed on the **EC2 Dashboard** in the **Resources** section.

To find your Regions using the AWS CLI

- Use the [describe-regions](#) command as follows to describe the Regions that are enabled for your account.

```
aws ec2 describe-regions
```

To describe all Regions, including any Regions that are disabled for your account, add the --all-regions option as follows.

```
aws ec2 describe-regions --all-regions
```

To find your Regions using the AWS Tools for Windows PowerShell

- Use the [Get-EC2Region](#) command as follows to describe the Regions for your account.

```
PS C:\> Get-EC2Region
```

Getting the Region name

You can use the Amazon Lightsail API to view the name of a Region.

To view the Region name using the AWS CLI

- Use the [get-regions](#) command as follows to describe the name of the specified Region.

```
aws lightsail get-regions --query "regions[?name=='region-name'].displayName" --output text
```

The following example returns the name of the us-east-2 Region.

```
aws lightsail get-regions --query "regions[?name=='us-east-2'].displayName" --output text
```

The following is the output:

```
Ohio
```

Specifying the Region for a resource

Every time you create an Amazon EC2 resource, you can specify the Region for the resource. You can specify the Region for a resource using the AWS Management Console or the command line.

Considerations

Some AWS resources might not be available in all Regions. Ensure that you can create the resources that you need in the desired Regions before you launch an instance.

To specify the Region for a resource using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Use the Region selector in the navigation bar.

To specify the default Region using the command line

You can set the value of an environment variable to the desired Regional endpoint (for example, <https://ec2.us-east-2.amazonaws.com>):

- `AWS_DEFAULT_REGION` (AWS CLI)
- `Set-AWSDefaultRegion` (AWS Tools for Windows PowerShell)

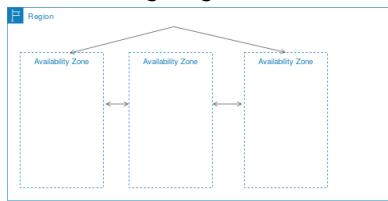
Alternatively, you can use the `--region` (AWS CLI) or `-Region` (AWS Tools for Windows PowerShell) command line option with each individual command. For example, `--region us-east-2`.

For more information about the endpoints for Amazon EC2, see [Amazon Elastic Compute Cloud Endpoints](#).

Availability Zones

Each Region has multiple, isolated locations known as *Availability Zones*. When you launch an instance, you can select an Availability Zone or let us choose one for you. If you distribute your instances across multiple Availability Zones and one instance fails, you can design your application so that an instance in another Availability Zone can handle requests.

The following diagram illustrates multiple Availability Zones in an AWS Region.



You can also use Elastic IP addresses to mask the failure of an instance in one Availability Zone by rapidly remapping the address to an instance in another Availability Zone. For more information, see [Elastic IP addresses \(p. 781\)](#).

An Availability Zone is represented by a Region code followed by a letter identifier; for example, `us-east-1a`. To ensure that resources are distributed across the Availability Zones for a Region, we independently map Availability Zones to names for each AWS account. For example, the Availability Zone `us-east-1a` for your AWS account might not be the same location as `us-east-1a` for another AWS account.

To coordinate Availability Zones across accounts, you must use the *AZ ID*, which is a unique and consistent identifier for an Availability Zone. For example, `use1-az1` is an AZ ID for the `us-east-1` Region and it has the same location in every AWS account.

You can view AZ IDs to determine the location of resources in one account relative to the resources in another account. For example, if you share a subnet in the Availability Zone with the AZ ID `use-az2` with another account, this subnet is available to that account in the Availability Zone whose AZ ID is also `use-az2`. The AZ ID for each VPC and subnet is displayed in the Amazon VPC console. For more information, see [Working with Shared VPCs](#) in the *Amazon VPC User Guide*.

As Availability Zones grow over time, our ability to expand them can become constrained. If this happens, we might restrict you from launching an instance in a constrained Availability Zone unless you

already have an instance in that Availability Zone. Eventually, we might also remove the constrained Availability Zone from the list of Availability Zones for new accounts. Therefore, your account might have a different number of available Availability Zones in a Region than another account.

Contents

- [Describing your Availability Zones \(p. 13\)](#)
- [Launching instances in an Availability Zone \(p. 13\)](#)
- [Migrating an instance to another Availability Zone \(p. 14\)](#)

Describing your Availability Zones

You can use the Amazon EC2 console or the command line interface to determine which Availability Zones are available for your account. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

To find your Availability Zones using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, view the options in the Region selector.
3. On the navigation pane, choose **EC2 Dashboard**.
4. The Availability Zones are listed under **Service health, Zone status**.

To find your Availability Zones using the AWS CLI

1. Use the [describe-availability-zones](#) command as follows to describe the Availability Zones within the specified Region.

```
aws ec2 describe-availability-zones --region region-name
```

2. Use the [describe-availability-zones](#) command as follows to describe the Availability Zones regardless of the opt-in status.

```
aws ec2 describe-availability-zones --all-availability-zones
```

To find your Availability Zones using the AWS Tools for Windows PowerShell

Use the [Get-EC2AvailabilityZone](#) command as follows to describe the Availability Zones within the specified Region.

```
PS C:\> Get-EC2AvailabilityZone -Region region-name
```

Launching instances in an Availability Zone

When you launch an instance, select a Region that puts your instances closer to specific customers, or meets the legal or other requirements that you have. By launching your instances in separate Availability Zones, you can protect your applications from the failure of a single location.

When you launch an instance, you can optionally specify an Availability Zone in the Region that you are using. If you do not specify an Availability Zone, we select an Availability Zone for you. When you launch your initial instances, we recommend that you accept the default Availability Zone, because this allows us

to select the best Availability Zone for you based on system health and available capacity. If you launch additional instances, specify a Zone only if your new instances must be close to, or separated from, your running instances.

Migrating an instance to another Availability Zone

If necessary, you can migrate an instance from one Availability Zone to another. For example, let's say you are trying to modify the instance type of your instance and we can't launch an instance of the new instance type in the current Availability Zone. In this case, you can migrate the instance to an Availability Zone where we are able to launch an instance of that instance type.

The migration process involves:

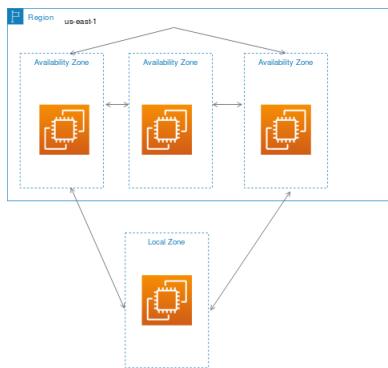
- Creating an AMI from the original instance
- Launching an instance in the new Availability Zone
- Updating the configuration of the new instance, as shown in the following procedure

To migrate an instance to another Availability Zone

1. Create an AMI from the instance. The procedure depends on your operating system and the type of root device volume for the instance. For more information, see the documentation that corresponds to your operating system and root device volume:
 - [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#)
 - [Creating an instance store-backed Linux AMI \(p. 127\)](#)
 - [Creating an Amazon EBS-Backed Windows AMI](#)
2. If you need to preserve the private IPv4 address of the instance, you must delete the subnet in the current Availability Zone and then create a subnet in the new Availability Zone with the same IPv4 address range as the original subnet. Note that you must terminate all instances in a subnet before you can delete it. Therefore, you should create AMIs from all of the instances in your subnet so that you can move all instances from the current subnet to the new subnet.
3. Launch an instance from the AMI that you just created, specifying the new Availability Zone or subnet. You can use the same instance type as the original instance, or select a new instance type. For more information, see [Launching instances in an Availability Zone \(p. 13\)](#).
4. If the original instance has an associated Elastic IP address, associate it with the new instance. For more information, see [Disassociating an Elastic IP address \(p. 786\)](#).
5. If the original instance is a Reserved Instance, change the Availability Zone for your reservation. (If you also changed the instance type, you can also change the instance type for your reservation.) For more information, see [Submitting modification requests \(p. 341\)](#).
6. (Optional) Terminate the original instance. For more information, see [Terminating an instance \(p. 608\)](#).

Local Zones

A Local Zone is an extension of an AWS Region in geographic proximity to your users. Local Zones have their own connections to the internet and support AWS Direct Connect, so resources created in a Local Zone can serve local users with low-latency communications. For more information, see [AWS Local Zones](#).



A Local Zone is represented by a Region code followed by an identifier that indicates the location, for example, `us-west-2-lax-1a`.

To use a Local Zone, you must first enable it. For more information, see [the section called “Enable Local Zones” \(p. 16\)](#). Next, create a subnet in the Local Zone. Finally, launch any of the following resources in the Local Zone subnet, so that your applications are close to your end users:

- Amazon EC2 instances
- Amazon EBS volumes
- Amazon FSx file servers
- Application Load Balancers
- Dedicated Hosts

For information about the available Local Zones, see [the section called “Available Regions” \(p. 8\)](#).

Contents

- [Describing your Local Zones \(p. 15\)](#)
- [Enable Local Zones \(p. 16\)](#)
- [Launching instances in a Local Zone \(p. 16\)](#)

Describing your Local Zones

You can use the Amazon EC2 console or the command line interface to determine which Local Zones are available for your account. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

To find your Local Zones using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, view the options in the Region selector.
3. On the navigation pane, choose **EC2 Dashboard**.
4. The Local Zones are listed under **Service health, Zone status**.

To find your Local Zones using the AWS CLI

1. Use the `describe-availability-zones` command as follows to describe the Local Zones in the specified Region.

```
aws ec2 describe-availability-zones --region region-name
```

2. Use the [describe-availability-zones](#) command as follows to describe the Local Zones regardless of whether they are enabled.

```
aws ec2 describe-availability-zones --all-availability-zones
```

To find your Local Zones using the AWS Tools for Windows PowerShell

Use the [Get-EC2AvailabilityZone](#) command as follows to describe the Local Zones in the specified Region.

```
PS C:\> Get-EC2AvailabilityZone -Region region-name
```

Enable Local Zones

Before you can specify a Local Zone for a resource or service, you must enable Local Zones.

Consideration

Some AWS resources might not be available in all Regions. Make sure that you can create the resources that you need in the desired Regions or Local Zones before launching an instance in a specific Local Zone.

To enable Local Zones using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the upper-left corner of the page, select **New EC2 Experience**. You cannot complete this task using the old console experience.
3. From the Region selector in the navigation bar, select the Region for the Local Zone.
4. On the navigation pane, choose **EC2 Dashboard**.
5. In the upper-right corner of the page, choose **Account attributes, Zones**.
6. Choose **Manage**.
7. For **Zone group**, choose **Enabled**.
8. Choose **Update zone group**.

To enable Local Zones using the AWS CLI

- Use the [modify-availability-zone-group](#) command.

Launching instances in a Local Zone

When you launch an instance, you can specify a subnet which is in a Local Zone. You also allocate an IP address from a network border group, which is a unique set of Availability Zones, Local Zones, or Wavelength Zones from which AWS advertises IP addresses, for example, `us-west-2-lax-1a`.

You can allocate the following IP addresses from a network border group:

- Elastic IPv4 addresses that Amazon provides
- IPv6 Amazon-provided VPC addresses

To launch an instance in a Local Zone:

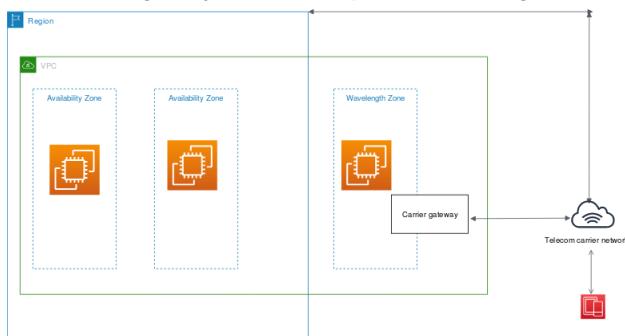
1. Enable Local Zones. For more information, see [Enable Local Zones \(p. 16\)](#).

2. Create a VPC in a Region that supports the Local Zone. For more information, see [Creating a VPC in the Amazon VPC User Guide](#).
3. Create a subnet. Select the Local Zone when you create the subnet. For more information, see [Creating a subnet in your VPC in the Amazon VPC User Guide](#).
4. Launch an instance, and select the subnet you created in the Local Zone. For more information, see [Launch your instance \(p. 497\)](#).

Wavelength Zones

AWS Wavelength enables developers to build applications that deliver ultra-low latencies to mobile devices and end users. Wavelength deploys standard AWS compute and storage services to the edge of telecommunication carriers' 5G networks. Developers can extend a virtual private cloud (VPC) to one or more Wavelength Zones, and then use AWS resources like Amazon EC2 instances to run applications that require ultra-low latency and a connection to AWS services in the Region.

A Wavelength Zone is an isolated zone in the carrier location where the Wavelength infrastructure is deployed. Wavelength Zones are tied to a Region. A Wavelength Zone is a logical extension of a Region, and is managed by the control plane in the Region.



A Wavelength Zone is represented by a Region code followed by an identifier that indicates the Wavelength Zone, for example, us-east-1-wl1-bos-wlz-1.

To use a Wavelength Zone, you must first opt in to the Zone. For more information, see [the section called "Enable Wavelength Zones" \(p. 18\)](#). Next, create a subnet in the Wavelength Zone. Finally, launch your resources in the Wavelength Zones subnet, so that your applications are closer to your end users.

Wavelength Zones are not available in every Region. For information about the Regions that support Wavelength Zones, see [Available Wavelength Zones](#) in the *AWS Wavelength Developer Guide*.

Contents

- [Describing your Wavelength Zones \(p. 17\)](#)
- [Enable Wavelength Zones \(p. 18\)](#)
- [Launching instances in a Wavelength Zone \(p. 19\)](#)

Describing your Wavelength Zones

You can use the Amazon EC2 console or the command line interface to determine which Wavelength Zones are available for your account. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

To find your Wavelength Zones using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. From the navigation bar, view the options in the Region selector.
3. On the navigation pane, choose **EC2 Dashboard**.
4. The Wavelength Zones are listed under **Service health, Zone status**.

To find your Wavelength Zones using the AWS CLI

1. Use the [describe-availability-zones](#) command as follows to describe the Wavelength Zones within the specified Region.

```
aws ec2 describe-availability-zones --region region-name
```

2. Use the [describe-availability-zones](#) command as follows to describe the Wavelength Zones regardless of the opt-in status.

```
aws ec2 describe-availability-zones --all-availability-zones
```

To find your Wavelength Zone using the AWS Tools for Windows PowerShell

Use the [Get-EC2AvailabilityZone](#) command as follows to describe the Wavelength Zone within the specified Region.

```
PS C:\> Get-EC2AvailabilityZone -Region region-name
```

Enable Wavelength Zones

Before you specify a Wavelength Zone for a resource or service, you must enable Wavelength Zones.

Considerations

- You must request access in order to use Wavelength Zones. For information about how to request Wavelength Zone access, see [AWS Wavelength](#).
- Some AWS resources are not available in all Regions. Make sure that you can create the resources that you need in the desired Region or Wavelength Zone before launching an instance in a specific Wavelength Zone.

To opt in to Wavelength Zone using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the upper-left corner of the page, select **New EC2 Experience**. You cannot complete this task using the old console experience.
3. From the Region selector in the navigation bar, select the Region for the Wavelength Zone.
4. On the navigation pane, choose **EC2 Dashboard**.
5. In the upper-right corner of the page, choose **Account attributes, Zones**.
6. Under **Wavelength Zones**, turn on each Wavelength Zone.
7. Enable the Wavelength Zone.

To enable Wavelength Zones using the AWS CLI

Use the [modify-availability-zone-group](#) command.

Launching instances in a Wavelength Zone

When you launch an instance, you can specify a subnet which is in a Wavelength Zone. You also allocate a carrier IP address from a network border group, which is a unique set of Availability Zones, Local Zones, or Wavelength Zones from which AWS advertises IP addresses, for example, us-east-1-wl1-bos-wlz-1.

For information about how to launch an instance in a Wavelength Zone, see [Get started with AWS Wavelength](#) in the *AWS Wavelength Developer Guide*.

AWS Outposts

AWS Outposts is a fully managed service that extends AWS infrastructure, services, APIs, and tools to customer premises. By providing local access to AWS managed infrastructure, AWS Outposts enables customers to build and run applications on premises using the same programming interfaces as in AWS Regions, while using local compute and storage resources for lower latency and local data processing needs.

An Outpost is a pool of AWS compute and storage capacity deployed at a customer site. AWS operates, monitors, and manages this capacity as part of an AWS Region. You can create subnets on your Outpost and specify them when you create AWS resources such as EC2 instances, EBS volumes, ECS clusters, and RDS instances. Instances in Outpost subnets communicate with other instances in the AWS Region using private IP addresses, all within the same VPC.

To begin using AWS Outposts, you must create an Outpost and order Outpost capacity. For more information about Outposts configurations, see [our catalog](#). After your Outpost equipment is installed, the compute and storage capacity is available for you when you launch Amazon EC2 instances and create Amazon EBS volumes on your Outpost.

Launching instances on an Outpost

You can launch EC2 instances in the Outpost subnet that you created. Security groups control inbound and outbound traffic for instances in an Outpost subnet, as they do for instances in an Availability Zone subnet. To connect to an EC2 instance in an Outpost subnet, you can specify a key pair when you launch the instance, as you do for instances in an Availability Zone subnet.

The root volume must be 30 GB or smaller. You can specify data volumes in the block device mapping of the AMI or the instance to provide additional storage. To trim unused blocks from the boot volume, see [How to Build Sparse EBS Volumes](#) in the AWS Partner Network Blog.

We recommend that you increase the NVMe timeout for the root volume. For more information, see [I/O operation timeout \(p. 1139\)](#).

For information about how to create an Outpost, see [Get started with AWS Outposts](#) in the *AWS Outposts User Guide*.

Creating a volume on an Outpost

You can create EBS volumes in the Outpost subnet that you created. When you create the volume, specify the Amazon Resource Name (ARN) of the Outpost.

The following `create-volume` command creates an empty 50 GB volume on the specified Outpost.

```
aws ec2 create-volume --availability-zone us-east-2a --outpost-arn arn:aws:outposts:us-east-2:123456789012:outpost/op-03e6fecad652a6138 --size 50
```

You must detach an Outpost volume before you can modify it. For more information about detaching volumes, see [Detaching an Amazon EBS volume from a Linux instance \(p. 1058\)](#).

Amazon EC2 root device volume

When you launch an instance, the *root device volume* contains the image used to boot the instance. When we introduced Amazon EC2, all AMIs were backed by Amazon EC2 instance store, which means the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3. After we introduced Amazon EBS, we introduced AMIs that are backed by Amazon EBS. This means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot.

You can choose between AMIs backed by Amazon EC2 instance store and AMIs backed by Amazon EBS. We recommend that you use AMIs backed by Amazon EBS, because they launch faster and use persistent storage.

Important

Only the following instance types support an instance store volume as the root device: C3, D2, G2, I2, M3, and R3.

For more information about the device names Amazon EC2 uses for your root volumes, see [Device naming on Linux instances \(p. 1210\)](#).

Contents

- [Root device storage concepts \(p. 20\)](#)
- [Choosing an AMI by root device type \(p. 22\)](#)
- [Determining the root device type of your instance \(p. 22\)](#)
- [Changing the root volume to persist \(p. 23\)](#)

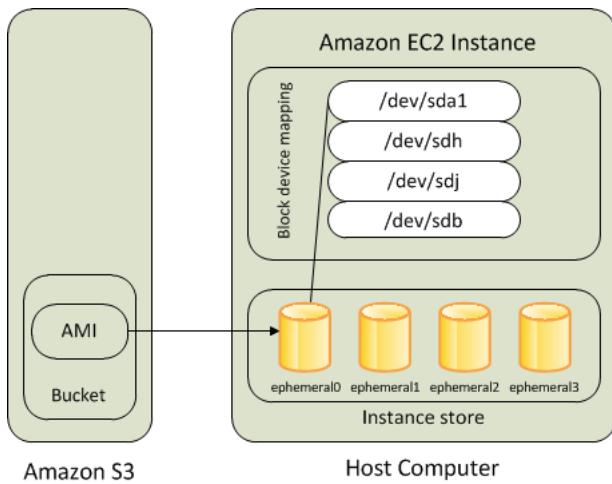
Root device storage concepts

You can launch an instance from either an instance store-backed AMI or an Amazon EBS-backed AMI. The description of an AMI includes which type of AMI it is; you'll see the root device referred to in some places as either `ebs` (for Amazon EBS-backed) or `instance store` (for instance store-backed). This is important because there are significant differences between what you can do with each type of AMI. For more information about these differences, see [Storage for the root device \(p. 100\)](#).

Instance store-backed instances

Instances that use instance stores for the root device automatically have one or more instance store volumes available, with one volume serving as the root device volume. When an instance is launched, the image that is used to boot the instance is copied to the root volume. Note that you can optionally use additional instance store volumes, depending on the instance type.

Any data on the instance store volumes persists as long as the instance is running, but this data is deleted when the instance is terminated (instance store-backed instances do not support the `Stop` action) or if it fails (such as if an underlying drive has issues).

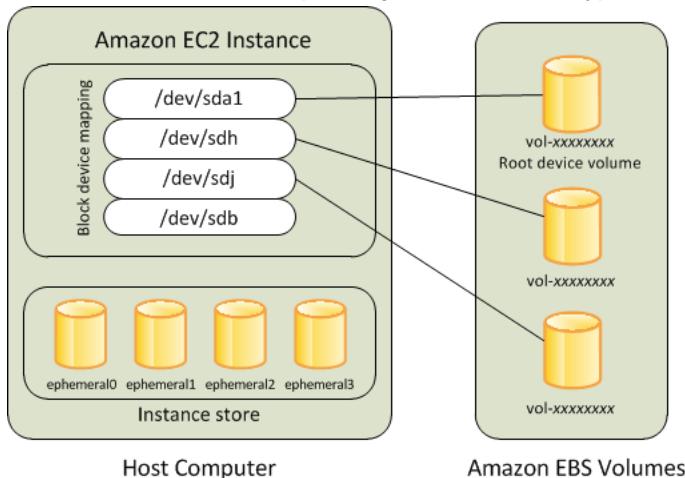


After an instance store-backed instance fails or terminates, it cannot be restored. If you plan to use Amazon EC2 instance store-backed instances, we highly recommend that you distribute the data on your instance stores across multiple Availability Zones. You should also back up critical data from your instance store volumes to persistent storage on a regular basis.

For more information, see [Amazon EC2 instance store \(p. 1188\)](#).

Amazon EBS-backed instances

Instances that use Amazon EBS for the root device automatically have an Amazon EBS volume attached. When you launch an Amazon EBS-backed instance, we create an Amazon EBS volume for each Amazon EBS snapshot referenced by the AMI you use. You can optionally use other Amazon EBS volumes or instance store volumes, depending on the instance type.



An Amazon EBS-backed instance can be stopped and later restarted without affecting data stored in the attached volumes. There are various instance- and volume-related tasks you can do when an Amazon EBS-backed instance is in a stopped state. For example, you can modify the properties of the instance, change its size, or update the kernel it is using, or you can attach your root volume to a different running instance for debugging or any other purpose.

If an Amazon EBS-backed instance fails, you can restore your session by following one of these methods:

- Stop and then start again (try this method first).
- Automatically snapshot all relevant volumes and create a new AMI. For more information, see [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#).

- Attach the volume to the new instance by following these steps:
 1. Create a snapshot of the root volume.
 2. Register a new AMI using the snapshot.
 3. Launch a new instance from the new AMI.
 4. Detach the remaining Amazon EBS volumes from the old instance.
 5. Reattach the Amazon EBS volumes to the new instance.

For more information, see [Amazon EBS volumes \(p. 1021\)](#).

Choosing an AMI by root device type

The AMI that you specify when you launch your instance determines the type of root device volume that your instance has.

To choose an Amazon EBS-backed AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **AMIs**.
3. From the filter lists, select the image type (such as **Public images**). In the search bar choose **Platform** to select the operating system (such as **Amazon Linux**), and **Root Device Type** to select **EBS images**.
4. (Optional) To get additional information to help you make your choice, choose the **Show/Hide Columns** icon, update the columns to display, and choose **Close**.
5. Choose an AMI and write down its AMI ID.

To choose an instance store-backed AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **AMIs**.
3. From the filter lists, select the image type (such as **Public images**). In the search bar, choose **Platform** to select the operating system (such as **Amazon Linux**), and **Root Device Type** to select **Instance store**.
4. (Optional) To get additional information to help you make your choice, choose the **Show/Hide Columns** icon, update the columns to display, and choose **Close**.
5. Choose an AMI and write down its AMI ID.

To verify the type of the root device volume of an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-images \(AWS CLI\)](#)
- [Get-EC2Image \(AWS Tools for Windows PowerShell\)](#)

Determining the root device type of your instance

New console

To determine the root device type of an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Instances**, and select the instance.
3. On the **Storage** tab, under **Root device details**, check the value of **Root device type** as follows:
 - If the value is `EBS`, this is an Amazon EBS-backed instance.
 - If the value is `INSTANCE-STORE`, this is an instance store-backed instance.

Old console

To determine the root device type of an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and select the instance.
3. On the **Description** tab, check the value of **Root device type** as follows:
 - If the value is `ebs`, this is an Amazon EBS-backed instance.
 - If the value is `instance store`, this is an instance store-backed instance.

To determine the root device type of an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-instances](#) (AWS CLI)
- [Get-EC2Instance](#) (AWS Tools for Windows PowerShell)

Changing the root volume to persist

By default, the root volume for an AMI backed by Amazon EBS is deleted when the instance terminates. You can change the default behavior to ensure that the volume persists after the instance terminates. To change the default behavior, set the `DeleteOnTermination` attribute to `false` using a block device mapping.

Topics

- [Configuring the root volume to persist during instance launch \(p. 23\)](#)
- [Configuring the root volume to persist for an existing instance \(p. 24\)](#)
- [Confirming that a root volume is configured to persist \(p. 25\)](#)

Configuring the root volume to persist during instance launch

You can configure the root volume to persist when you launch an instance using the Amazon EC2 console or the command line tools.

To configure the root volume to persist when you launch an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and then choose **Launch instances**.
3. On the **Choose an Amazon Machine Image (AMI)** page, select the AMI to use and choose **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, deselect **Delete On Termination** for the root volume.
6. Complete the remaining wizard pages, and then choose **Launch**.

To configure the root volume to persist when you launch an instance using the AWS CLI

Use the [run-instances](#) command and include a block device mapping that sets the `DeleteOnTermination` attribute to `false`.

```
$ aws ec2 run-instances --block-device-mappings file://mapping.json ...other parameters...
```

Specify the following in `mapping.json`.

```
[  
  {  
    "DeviceName": "/dev/sda1",  
    "Ebs": {  
      "DeleteOnTermination": false  
    }  
  }  
]
```

To configure the root volume to persist when you launch an instance using the Tools for Windows PowerShell

Use the [New-EC2Instance](#) command and include a block device mapping that sets the `DeleteOnTermination` attribute to `false`.

```
C:\> $ebs = New-Object Amazon.EC2.Model.EbsBlockDevice  
C:\> $ebs.DeleteOnTermination = $false  
C:\> $bdm = New-Object Amazon.EC2.Model.BlockDeviceMapping  
C:\> $bdm.DeviceName = "dev/xvda"  
C:\> $bdm.Ebs = $ebs  
C:\> New-EC2Instance -ImageId ami-0abcdef1234567890 -BlockDeviceMapping $bdm ...other  
parameters...
```

Configuring the root volume to persist for an existing instance

You can configure the root volume to persist for a running instance using the command line tools only.

To configure the root volume to persist for an existing instance using the AWS CLI

Use the [modify-instance-attribute](#) command with a block device mapping that sets the `DeleteOnTermination` attribute to `false`.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --block-device-mappings  
"[{\\"DeviceName\\": \"/dev/xvda\", \\"Ebs\\\": {\\"DeleteOnTermination\\": false}}]"
```

To configure the root volume to persist for an existing instance using the AWS Tools for Windows PowerShell

Use the [Edit-EC2InstanceAttribute](#) command with a block device mapping that sets the `DeleteOnTermination` attribute to `false`.

```
C:\> $ebs = New-Object Amazon.EC2.Model.EbsInstanceBlockDeviceSpecification  
C:\> $ebs.DeleteOnTermination = $false  
C:\> $bdm = New-Object Amazon.EC2.Model.InstanceBlockDeviceMappingSpecification  
C:\> $bdm.DeviceName = "/dev/xvda"  
C:\> $bdm.Ebs = $ebs  
C:\> Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -BlockDeviceMapping $bdm
```

Confirming that a root volume is configured to persist

You can confirm that a root volume is configured to persist using the Amazon EC2 console or the command line tools.

New console

To confirm that a root volume is configured to persist using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and then select the instance.
3. In the **Storage** tab, under **Block devices**, locate the entry for the root volume. If **Delete on termination** is **No**, the volume is configured to persist.

Old console

To confirm that a root volume is configured to persist using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and then select the instance.
3. In the **Description** tab, choose the entry for **Root device**. If **Delete on termination** is **False**, the volume is configured to persist.

To confirm that a root volume is configured to persist using the AWS CLI

Use the [describe-instances](#) command and verify that the **DeleteOnTermination** attribute in the **BlockDeviceMappings** response element is set to **false**.

```
$ aws ec2 describe-instances --instance-id i-1234567890abcdef0
```

```
...
    "BlockDeviceMappings": [
        {
            "DeviceName": "/dev/sda1",
            "Ebs": {
                "Status": "attached",
                "DeleteOnTermination": false,
                "VolumeId": "vol-1234567890abcdef0",
                "AttachTime": "2013-07-19T02:42:39.000Z"
            }
        }
    ...
}
```

To confirm that a root volume is configured to persist using the AWS Tools for Windows PowerShell

Use the [Get-EC2Instance](#) and verify that the **DeleteOnTermination** attribute in the **BlockDeviceMappings** response element is set to **false**.

```
C:\> (Get-EC2Instance -InstanceId i-i-1234567890abcdef0).Instances.BlockDeviceMappings.Ebs
```

Setting up with Amazon EC2

Complete the tasks in this section to get set up for launching an Amazon EC2 instance for the first time:

1. [Sign up for AWS \(p. 26\)](#)
2. [Create a key pair \(p. 26\)](#)
3. [Create a security group \(p. 27\)](#)

When you are finished, you will be ready for the [Amazon EC2 Getting started \(p. 30\)](#) tutorial.

Sign up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including Amazon EC2. You are charged only for the services that you use.

With Amazon EC2, you pay only for what you use. If you are a new AWS customer, you can get started with Amazon EC2 for free. For more information, see [AWS Free Tier](#).

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To create an AWS account

1. Open <https://portal.aws.amazon.com/billing/signup>.
2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Create a key pair

AWS uses public-key cryptography to secure the login information for your instance. A Linux instance has no password; you use a key pair to log in to your instance securely. You specify the name of the key pair when you launch your instance, then provide the private key when you log in using SSH.

If you haven't created a key pair already, you can create one using the Amazon EC2 console. Note that if you plan to launch instances in multiple Regions, you'll need to create a key pair in each Region. For more information about Regions, see [Regions and Zones \(p. 7\)](#).

You can create a key pair using one of the following methods.

New console

To create your key pair

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Key Pairs**.
3. Choose **Create key pair**.
4. For **Name**, enter a descriptive name for the key pair. Amazon EC2 associates the public key with the name that you specify as the key name. A key name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

5. For **File format**, choose the format in which to save the private key. To save the private key in a format that can be used with OpenSSH, choose **pem**. To save the private key in a format that can be used with PuTTY, choose **ppk**.
6. Choose **Create key pair**.
7. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is determined by the file format you chose. Save the private key file in a safe place.

Important

This is the only chance for you to save the private key file.

8. If you will use an SSH client on a macOS or Linux computer to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.

```
chmod 400 my-key-pair.pem
```

If you do not set these permissions, then you cannot connect to your instance using this key pair. For more information, see [Error: Unprotected private key file \(p. 1252\)](#).

Old console

To create your key pair

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.

Note

The navigation pane is on the left side of the Amazon EC2 console. If you do not see the pane, it might be minimized; choose the arrow to expand the pane.

3. Choose **Create Key Pair**.
4. For **Key pair name**, enter a name for the new key pair, and then choose **Create**. The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.
5. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is **.pem**. Save the private key file in a safe place.

Important

This is the only chance for you to save the private key file.

6. If you will use an SSH client on a macOS or Linux computer to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.

```
chmod 400 my-key-pair.pem
```

If you do not set these permissions, then you cannot connect to your instance using this key pair. For more information, see [Error: Unprotected private key file \(p. 1252\)](#).

For more information, see [Amazon EC2 key pairs and Linux instances \(p. 985\)](#).

Create a security group

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group that enable you to connect to your instance

from your IP address using SSH. You can also add rules that allow inbound and outbound HTTP and HTTPS access from anywhere.

Note that if you plan to launch instances in multiple Regions, you'll need to create a security group in each Region. For more information about Regions, see [Regions and Zones \(p. 7\)](#).

Prerequisites

You'll need the public IPv4 address of your local computer. The security group editor in the Amazon EC2 console can automatically detect the public IPv4 address for you. Alternatively, you can use the search phrase "what is my IP address" in an Internet browser, or use the following service: [Check IP](#). If you are connecting through an Internet service provider (ISP) or from behind a firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

You can create a custom security group using one of the following methods.

New console

To create a security group with least privilege

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select a Region for the security group. Security groups are specific to a Region, so you should select the same Region in which you created your key pair.
3. In the navigation pane, choose **Security Groups**.
4. Choose **Create security group**.
5. In the **Basic details** section, do the following:
 - a. Enter a name for the new security group and a description. Use a name that is easy for you to remember, such as your user name, followed by _SG_, plus the Region name. For example, *me_SG_uswest2*.
 - b. In the **VPC** list, select your default VPC for the Region.
6. In the **Inbound rules** section, create the following rules (choose **Add rule** for each new rule):
 - Choose **HTTP** from the **Type** list, and make sure that **Source** is set to **Anywhere** (0.0.0.0/0).
 - Choose **HTTPS** from the **Type** list, and make sure that **Source** is set to **Anywhere** (0.0.0.0/0).
 - Choose **SSH** from the **Type** list. In the **Source** box, choose **My IP** to automatically populate the field with the public IPv4 address of your local computer. Alternatively, choose **Custom** and specify the public IPv4 address of your computer or network in CIDR notation. To specify an individual IP address in CIDR notation, add the routing suffix /32, for example, 203.0.113.25/32. If your company allocates addresses from a range, specify the entire range, such as 203.0.113.0/24.

Warning

For security reasons, we don't recommend that you allow SSH access from all IPv4 addresses (0.0.0.0/0) to your instance, except for testing purposes and only for a short time.

7. Choose **Create security group**.

Old console

To create a security group with least privilege

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Choose **Create Security Group**.

4. Enter a name for the new security group and a description. Use a name that is easy for you to remember, such as your user name, followed by _SG_, plus the Region name. For example, *me_SG_uswest2*.
5. In the **VPC** list, select your default VPC for the Region.
6. On the **Inbound** tab, create the following rules (choose **Add rule** for each new rule):
 - Choose **HTTP** from the **Type** list, and make sure that **Source** is set to **Anywhere** (0.0.0.0/0).
 - Choose **HTTPS** from the **Type** list, and make sure that **Source** is set to **Anywhere** (0.0.0.0/0).
 - Choose **SSH** from the **Type** list. In the **Source** box, choose **My IP** to automatically populate the field with the public IPv4 address of your local computer. Alternatively, choose **Custom** and specify the public IPv4 address of your computer or network in CIDR notation. To specify an individual IP address in CIDR notation, add the routing suffix /32, for example, 203.0.113.25/32. If your company allocates addresses from a range, specify the entire range, such as 203.0.113.0/24.

Warning

For security reasons, we don't recommend that you allow SSH access from all IPv4 addresses (0.0.0.0/0) to your instance, except for testing purposes and only for a short time.

7. Choose **Create**.

Command line

To create a security group with least privilege

Use one of the following commands:

- [create-security-group](#) (AWS CLI)
- [New-EC2SecurityGroup](#) (AWS Tools for Windows PowerShell)

For more information, see [Amazon EC2 security groups for Linux instances \(p. 999\)](#).

Tutorial: Getting started with Amazon EC2 Linux instances

Use this tutorial to get started with Amazon Elastic Compute Cloud (Amazon EC2). You'll learn how to launch, connect to, and use a Linux instance. An *instance* is a virtual server in the AWS cloud. With Amazon EC2, you can set up and configure the operating system and applications that run on your instance.

To get started with a Windows instance, see [Getting started with Amazon EC2 Windows instances](#).

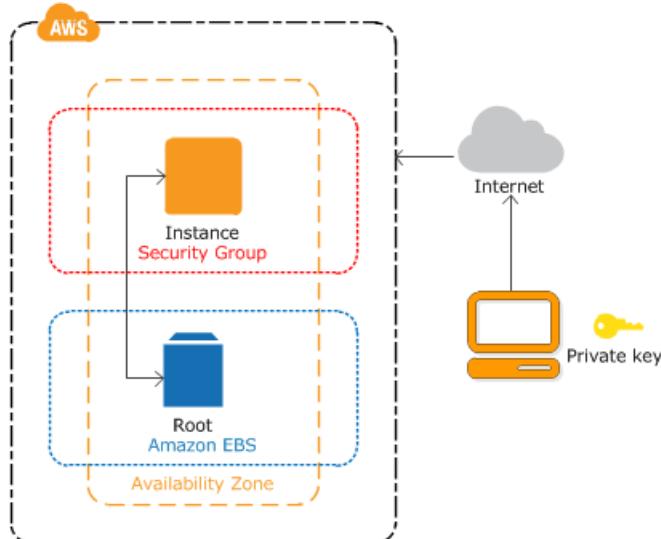
When you sign up for AWS, you can get started with Amazon EC2 using the [AWS Free Tier](#). If you created your AWS account less than 12 months ago, and have not already exceeded the free tier benefits for Amazon EC2, it will not cost you anything to complete this tutorial, because we help you select options that are within the free tier benefits. Otherwise, you'll incur the standard Amazon EC2 usage fees from the time that you launch the instance until you terminate the instance (which is the final task of this tutorial), even if it remains idle.

Contents

- [Overview \(p. 30\)](#)
- [Prerequisites \(p. 31\)](#)
- [Step 1: Launch an instance \(p. 31\)](#)
- [Step 2: Connect to your instance \(p. 32\)](#)
- [Step 3: Clean up your instance \(p. 32\)](#)
- [Next steps \(p. 33\)](#)

Overview

The instance is an Amazon EBS-backed instance (meaning that the root volume is an EBS volume). You can either specify the Availability Zone in which your instance runs, or let Amazon EC2 select an Availability Zone for you. When you launch your instance, you secure it by specifying a key pair and security group. When you connect to your instance, you must specify the private key of the key pair that you specified when launching your instance.



Tasks

To complete this tutorial, perform the following tasks:

1. [Launch an instance \(p. 31\)](#)
2. [Connect to Your Instance \(p. 32\)](#)
3. [Clean up your instance \(p. 32\)](#)

Related tutorials

- If you'd prefer to launch a Windows instance, see this tutorial in the [Amazon EC2 User Guide for Windows Instances: Getting started with Amazon EC2 Windows instances](#).
- If you'd prefer to use the command line, see this tutorial in the [AWS Command Line Interface User Guide: Using Amazon EC2 through the AWS CLI](#).

Prerequisites

Before you begin, be sure that you've completed the steps in [Setting up with Amazon EC2 \(p. 26\)](#).

Step 1: Launch an instance

You can launch a Linux instance using the AWS Management Console as described in the following procedure. This tutorial is intended to help you launch your first instance quickly, so it doesn't cover all possible options. For more information about the advanced options, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#).

To launch an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the console dashboard, choose **Launch Instance**.
3. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations, called *Amazon Machine Images (AMIs)*, that serve as templates for your instance. Select an HVM version of Amazon Linux 2. Notice that these AMIs are marked "Free tier eligible."
4. On the **Choose an Instance Type** page, you can select the hardware configuration of your instance. Select the `t2.micro` instance type, which is selected by default. The `t2.micro` instance type is eligible for the free tier. In Regions where `t2.micro` is unavailable, you can use a `t3.micro` instance under the free tier. For more information, see [AWS Free Tier](#).
5. Choose **Review and Launch** to let the wizard complete the other configuration settings for you.
6. On the **Review Instance Launch** page, under **Security Groups**, you'll see that the wizard created and selected a security group for you. You can use this security group, or alternatively you can select the security group that you created when getting set up using the following steps:
 - a. Choose **Edit security groups**.
 - b. On the **Configure Security Group** page, ensure that **Select an existing security group** is selected.
 - c. Select your security group from the list of existing security groups, and then choose **Review and Launch**.
7. On the **Review Instance Launch** page, choose **Launch**.

8. When prompted for a key pair, select **Choose an existing key pair**, then select the key pair that you created when getting set up.

Warning

Don't select **Proceed without a key pair**. If you launch your instance without a key pair, then you can't connect to it.

When you are ready, select the acknowledgement check box, and then choose **Launch Instances**.

9. A confirmation page lets you know that your instance is launching. Choose **View Instances** to close the confirmation page and return to the console.
10. On the **Instances** screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is **Pending**. After the instance starts, its state changes to **running** and it receives a public DNS name. (If the **Public DNS (IPv4)** column is hidden, choose **Show/Hide Columns** (the gear-shaped icon) in the top right corner of the page and then select **Public DNS (IPv4)**.)
11. It can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks; you can view this information in the **Status Checks** column.

Step 2: Connect to your instance

There are several ways to connect to your Linux instance. For more information, see [Connect to your Linux instance \(p. 563\)](#).

Important

You can't connect to your instance unless you launched it with a key pair for which you have the **.pem** file and you launched it with a security group that allows SSH access from your computer. If you can't connect to your instance, see [Troubleshooting connecting to your instance \(p. 1246\)](#) for assistance.

Step 3: Clean up your instance

After you've finished with the instance that you created for this tutorial, you should clean up by terminating the instance. If you want to do more with this instance before you clean up, see [Next steps \(p. 33\)](#).

Important

Terminating an instance effectively deletes it; you can't reconnect to an instance after you've terminated it.

If you launched an instance that is not within the [AWS Free Tier](#), you'll stop incurring charges for that instance as soon as the instance status changes to **shutting down** or **terminated**. If you'd like to keep your instance for later, but not incur charges, you can stop the instance now and then start it again later. For more information, see [Stopping Instances](#).

To terminate your instance

1. In the navigation pane, choose **Instances**. In the list of instances, select the instance.
2. Choose **Actions, Instance state, Terminate instance**.
3. Choose **Terminate** when prompted for confirmation.

Amazon EC2 shuts down and terminates your instance. After your instance is terminated, it remains visible on the console for a short while, and then the entry is automatically deleted. You cannot remove the terminated instance from the console display yourself.

Next steps

After you start your instance, you might want to try some of the following exercises:

- Learn how to remotely manage your EC2 instance using Run Command. For more information, see [AWS Systems Manager Run Command](#) in the *AWS Systems Manager User Guide*.
- Configure a CloudWatch alarm to notify you if your usage exceeds the Free Tier. For more information, see [Create a Billing Alarm](#) in the *AWS Billing and Cost Management User Guide*.
- Add an EBS volume. For more information, see [Creating an Amazon EBS volume \(p. 1040\)](#) and [Attaching an Amazon EBS volume to an instance \(p. 1042\)](#).
- Install the LAMP stack. For more information, see [Tutorial: Install a LAMP web server on Amazon Linux 2 \(p. 36\)](#).

Best practices for Amazon EC2

This list of practices will help you get the maximum benefit from Amazon EC2.

Security

- Manage access to AWS resources and APIs using identity federation, IAM users, and IAM roles. Establish credential management policies and procedures for creating, distributing, rotating, and revoking AWS access credentials. For more information, see [IAM Best Practices](#) in the *IAM User Guide*.
- Implement the least permissive rules for your security group. For more information, see [Security group rules \(p. 1000\)](#).
- Regularly patch, update, and secure the operating system and applications on your instance. For more information about updating Amazon Linux 2 or the Amazon Linux AMI, see [Managing Software on Your Linux Instance](#) in the *Amazon EC2 User Guide for Linux Instances*.

Storage

- Understand the implications of the root device type for data persistence, backup, and recovery. For more information, see [Storage for the root device \(p. 100\)](#).
- Use separate Amazon EBS volumes for the operating system versus your data. Ensure that the volume with your data persists after instance termination. For more information, see [Preserving Amazon EBS volumes on instance termination \(p. 610\)](#).
- Use the instance store available for your instance to store temporary data. Remember that the data stored in instance store is deleted when you stop or terminate your instance. If you use instance store for database storage, ensure that you have a cluster with a replication factor that ensures fault tolerance.
- Encrypt EBS volumes and snapshots. For more information, see [Amazon EBS encryption \(p. 1121\)](#).

Resource management

- Use instance metadata and custom resource tags to track and identify your AWS resources. For more information, see [Instance metadata and user data \(p. 658\)](#) and [Tagging your Amazon EC2 resources \(p. 1229\)](#).
- View your current limits for Amazon EC2. Plan to request any limit increases in advance of the time that you'll need them. For more information, see [Amazon EC2 service quotas \(p. 1241\)](#).

Backup and recovery

- Regularly back up your EBS volumes using [Amazon EBS snapshots \(p. 1060\)](#), and create an [Amazon Machine Image \(AMI\) \(p. 97\)](#) from your instance to save the configuration as a template for launching future instances.
- Deploy critical components of your application across multiple Availability Zones, and replicate your data appropriately.
- Design your applications to handle dynamic IP addressing when your instance restarts. For more information, see [Amazon EC2 instance IP addressing \(p. 759\)](#).
- Monitor and respond to events. For more information, see [Monitoring Amazon EC2 \(p. 694\)](#).
- Ensure that you are prepared to handle failover. For a basic solution, you can manually attach a network interface or Elastic IP address to a replacement instance. For more information, see [Elastic network interfaces \(p. 789\)](#). For an automated solution, you can use Amazon EC2 Auto Scaling. For more information, see the [Amazon EC2 Auto Scaling User Guide](#).

- Regularly test the process of recovering your instances and Amazon EBS volumes if they fail.

Networking

- Set the time-to-live (TTL) value for your applications to 255, for IPv4 and IPv6. If you use a smaller value, there is a risk that the TTL will expire while application traffic is in transit, causing reachability issues for your instances.

Tutorials for Amazon EC2 Instances Running Linux

The following tutorials show you how to perform common tasks using EC2 instances running Linux. For videos, see [AWS Instructional Videos and Labs](#).

Tutorials

- [Tutorial: Install a LAMP web server on Amazon Linux 2 \(p. 36\)](#)
- [Tutorial: Install a LAMP web server with the Amazon Linux AMI \(p. 46\)](#)
- [Tutorial: Hosting a WordPress blog with Amazon Linux \(p. 56\)](#)
- [Tutorial: Configure SSL/TLS on Amazon Linux 2 \(p. 65\)](#)
- [Tutorial: Configure SSL/TLS on Amazon Linux \(p. 80\)](#)
- [Tutorial: Increase the availability of your application on Amazon EC2 \(p. 93\)](#)

Tutorial: Install a LAMP web server on Amazon Linux 2

The following procedures help you install an Apache web server with PHP and [MariaDB](#) (a community-developed fork of MySQL) support on your Amazon Linux 2 instance (sometimes called a LAMP web server or LAMP stack). You can use this server to host a static website or deploy a dynamic PHP application that reads and writes information to a database.

Important

To set up a LAMP web server on Amazon Linux AMI, see [Tutorial: Install a LAMP web server with the Amazon Linux AMI \(p. 46\)](#).

If you are trying to set up a LAMP web server on an Ubuntu or Red Hat Enterprise Linux instance, this tutorial will not work for you. For more information about other distributions, see their specific documentation. For information about LAMP web servers on Ubuntu, see the Ubuntu community documentation [ApacheMySQLPHP](#) topic.

Option: Complete this tutorial using automation

To complete this tutorial using AWS Systems Manager Automation instead of the following tasks, run the [AWS Docs - Install a LAMP Server - AL2](#) Automation document.

Tasks

- [Step 1: Prepare the LAMP server \(p. 37\)](#)
- [Step 2: Test your LAMP server \(p. 40\)](#)
- [Step 3: Secure the database server \(p. 41\)](#)
- [Step 4: \(Optional\) Install phpMyAdmin \(p. 42\)](#)
- [Troubleshooting \(p. 45\)](#)
- [Related topics \(p. 45\)](#)

Step 1: Prepare the LAMP server

Prerequisites

- This tutorial assumes that you have already launched a new instance using Amazon Linux 2, with a public DNS name that is reachable from the internet. For more information, see [Step 1: Launch an instance \(p. 31\)](#). You must also have configured your security group to allow SSH (port 22), HTTP (port 80), and HTTPS (port 443) connections. For more information about these prerequisites, see [Authorizing inbound traffic for your Linux instances \(p. 983\)](#).
- The following procedure installs the latest PHP version available on Amazon Linux 2, currently PHP 7.2. If you plan to use PHP applications other than those described in this tutorial, you should check their compatibility with PHP 7.2.

To prepare the LAMP server

1. [Connect to your instance \(p. 32\)](#).
2. To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process may take a few minutes, but it is important to make sure that you have the latest security updates and bug fixes.

The `-y` option installs the updates without asking for confirmation. If you would like to examine the updates before installing, you can omit this option.

```
[ec2-user ~]$ sudo yum update -y
```

3. Install the `lamp-mariadb10.2-php7.2` and `php7.2` Amazon Linux Extras repositories to get the latest versions of the LAMP MariaDB and PHP packages for Amazon Linux 2.

```
[ec2-user ~]$ sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
```

If you receive an error stating `sudo: amazon-linux-extras: command not found`, then your instance was not launched with an Amazon Linux 2 AMI (perhaps you are using the Amazon Linux AMI instead). You can view your version of Amazon Linux using the following command.

```
cat /etc/system-release
```

To set up a LAMP web server on Amazon Linux AMI , see [Tutorial: Install a LAMP web server with the Amazon Linux AMI \(p. 46\)](#).

4. Now that your instance is current, you can install the Apache web server, MariaDB, and PHP software packages.

Use the `yum install` command to install multiple software packages and all related dependencies at the same time.

```
[ec2-user ~]$ sudo yum install -y httpd mariadb-server
```

You can view the current versions of these packages using the following command:

```
yum info package_name
```

5. Start the Apache web server.

```
[ec2-user ~]$ sudo systemctl start httpd
```

6. Use the **systemctl** command to configure the Apache web server to start at each system boot.

```
[ec2-user ~]$ sudo systemctl enable httpd
```

You can verify that **httpd** is on by running the following command:

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

7. Add a security rule to allow inbound HTTP (port 80) connections to your instance if you have not already done so. By default, a **launch-wizard-N** security group was set up for your instance during initialization. This group contains a single rule to allow SSH connections.
 - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 - b. Choose **Instances** and select your instance.
 - c. On the **Security** tab, view the inbound rules. You should see the following rule:

Port range	Protocol	Source
22	tcp	0.0.0.0/0
 - d. Choose the link for the security group. Using the procedures in [Adding rules to a security group \(p. 1006\)](#), add a new inbound security rule with the following values:
 - **Type:** HTTP
 - **Protocol:** TCP
 - **Port Range:** 80
 - **Source:** Custom
8. Test your web server. In a web browser, type the public DNS address (or the public IP address) of your instance. If there is no content in `/var/www/html`, you should see the Apache test page. You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS** column; if this column is hidden, choose **Show/Hide Columns** (the gear-shaped icon) and choose **Public DNS**).

If you are unable to see the Apache test page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic. For information about adding an HTTP rule to your security group, see [Adding rules to a security group \(p. 1006\)](#).

Important

If you are not using Amazon Linux, you may also need to configure the firewall on your instance to allow these connections. For more information about how to configure the firewall, see the documentation for your specific distribution.

Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being served, follow the instructions in the file `/etc/httpd/conf/wELCOME.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:



Apache **httpd** serves files that are kept in a directory called the Apache document root. The Amazon Linux Apache document root is `/var/www/html`, which by default is owned by root.

To allow the `ec2-user` account to manipulate files in this directory, you must modify the ownership and permissions of the directory. There are many ways to accomplish this task. In this tutorial, you add `ec2-user` to the `apache` group, to give the `apache` group ownership of the `/var/www` directory and assign write permissions to the group.

To set file permissions

1. Add your user (in this case, `ec2-user`) to the `apache` group.

```
[ec2-user ~]$ sudo usermod -a -G apache ec2-user
```

2. Log out and then log back in again to pick up the new group, and then verify your membership.
 - a. Log out (use the `exit` command or close the terminal window):

```
[ec2-user ~]$ exit
```

- b. To verify your membership in the `apache` group, reconnect to your instance, and then run the following command:

```
[ec2-user ~]$ groups
ec2-user adm wheel apache systemd-journal
```

3. Change the group ownership of `/var/www` and its contents to the `apache` group.

```
[ec2-user ~]$ sudo chown -R ec2-user:apache /var/www
```

4. To add group write permissions and to set the group ID on future subdirectories, change the directory permissions of /var/www and its subdirectories.

```
[ec2-user ~]$ sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;
```

5. To add group write permissions, recursively change the file permissions of /var/www and its subdirectories:

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

Now, `ec2-user` (and any future members of the `apache` group) can add, delete, and edit files in the Apache document root, enabling you to add content, such as a static website or a PHP application.

To secure your web server (Optional)

A web server running the HTTP protocol provides no transport security for the data that it sends or receives. When you connect to an HTTP server using a web browser, the URLs that you visit, the content of webpages that you receive, and the contents (including passwords) of any HTML forms that you submit are all visible to eavesdroppers anywhere along the network pathway. The best practice for securing your web server is to install support for HTTPS (HTTP Secure), which protects your data with SSL/TLS encryption.

For information about enabling HTTPS on your server, see [Tutorial: Configure SSL/TLS on Amazon Linux 2 \(p. 65\)](#).

Step 2: Test your LAMP server

If your server is installed and running, and your file permissions are set correctly, your `ec2-user` account should be able to create a PHP file in the `/var/www/html` directory that is available from the internet.

To test your LAMP server

1. Create a PHP file in the Apache document root.

```
[ec2-user ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

If you get a "Permission denied" error when trying to run this command, try logging out and logging back in again to pick up the proper group permissions that you configured in [To set file permissions \(p. 39\)](#).

2. In a web browser, type the URL of the file that you just created. This URL is the public DNS address of your instance followed by a forward slash and the file name. For example:

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

You should see the PHP information page:

PHP Version 7.2.0

System	Linux ip-172-31-22-15.us-west-2.compute.internal 4.9.62-10.57.amzn2.x86_64
Build Date	Dec 13 2017 03:34:37
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-dba.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mysqlind.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-pspell.ini, /etc/php.d/20-session.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS

If you do not see this page, verify that the `/var/www/html/phpinfo.php` file was created properly in the previous step. You can also verify that all of the required packages were installed with the following command.

```
[ec2-user ~]$ sudo yum list installed httpd mariadb-server php-mysqld
```

If any of the required packages are not listed in your output, install them with the **sudo yum install package** command. Also verify that the `php7.2` and `1amp-mariadb10.2-php7.2` extras are enabled in the output of the **amazon-linux-extras** command.

3. Delete the `phpinfo.php` file. Although this can be useful information, it should not be broadcast to the internet for security reasons.

```
[ec2-user ~]$ rm /var/www/html/phpinfo.php
```

You should now have a fully functional LAMP web server. If you add content to the Apache document root at `/var/www/html`, you should be able to view that content at the public DNS address for your instance.

Step 3: Secure the database server

The default installation of the MariaDB server has several features that are great for testing and development, but they should be disabled or removed for production servers. The **mysql_secure_installation** command walks you through the process of setting a root password and removing the insecure features from your installation. Even if you are not planning on using the MariaDB server, we recommend performing this procedure.

To secure the MariaDB server

1. Start the MariaDB server.

```
[ec2-user ~]$ sudo systemctl start mariadb
```

2. Run **mysql_secure_installation**.

```
[ec2-user ~]$ sudo mysql_secure_installation
```

- a. When prompted, type a password for the root account.

- i. Type the current root password. By default, the root account does not have a password set. Press Enter.
- ii. Type **y** to set a password, and type a secure password twice. For more information about creating a secure password, see <https://identitysafe.norton.com/password-generator/>. Make sure to store this password in a safe place.

Setting a root password for MariaDB is only the most basic measure for securing your database. When you build or install a database-driven application, you typically create a database service user for that application and avoid using the root account for anything but database administration.

- b. Type **y** to remove the anonymous user accounts.
 - c. Type **y** to disable the remote root login.
 - d. Type **y** to remove the test database.
 - e. Type **y** to reload the privilege tables and save your changes.
3. (Optional) If you do not plan to use the MariaDB server right away, stop it. You can restart it when you need it again.

```
[ec2-user ~]$ sudo systemctl stop mariadb
```

4. (Optional) If you want the MariaDB server to start at every boot, type the following command.

```
[ec2-user ~]$ sudo systemctl enable mariadb
```

Step 4: (Optional) Install phpMyAdmin

phpMyAdmin is a web-based database management tool that you can use to view and edit the MySQL databases on your EC2 instance. Follow the steps below to install and configure phpMyAdmin on your Amazon Linux instance.

Important

We do not recommend using phpMyAdmin to access a LAMP server unless you have enabled SSL/TLS in Apache; otherwise, your database administrator password and other data are transmitted insecurely across the internet. For security recommendations from the developers, see [Securing your phpMyAdmin installation](#). For general information about securing a web server on an EC2 instance, see [Tutorial: Configure SSL/TLS on Amazon Linux 2 \(p. 65\)](#).

To install phpMyAdmin

1. Install the required dependencies.

```
[ec2-user ~]$ sudo yum install php-mbstring -y
```

2. Restart Apache.

```
[ec2-user ~]$ sudo systemctl restart httpd
```

3. Restart php-fpm.

```
[ec2-user ~]$ sudo systemctl restart php-fpm
```

4. Navigate to the Apache document root at /var/www/html.

```
[ec2-user ~]$ cd /var/www/html
```

5. Select a source package for the latest phpMyAdmin release from <https://www.phpmyadmin.net/downloads>. To download the file directly to your instance, copy the link and paste it into a `wget` command, as in this example:

```
[ec2-user html]$ wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz
```

6. Create a phpMyAdmin folder and extract the package into it with the following command.

```
[ec2-user html]$ mkdir phpMyAdmin && tar -xvzf phpMyAdmin-latest-all-languages.tar.gz -C phpMyAdmin --strip-components 1
```

7. Delete the `phpMyAdmin-latest-all-languages.tar.gz` tarball.

```
[ec2-user html]$ rm phpMyAdmin-latest-all-languages.tar.gz
```

8. (Optional) If the MySQL server is not running, start it now.

```
[ec2-user ~]$ sudo systemctl start mariadb
```

9. In a web browser, type the URL of your phpMyAdmin installation. This URL is the public DNS address (or the public IP address) of your instance followed by a forward slash and the name of your installation directory. For example:

```
http://my.public.dns.amazonaws.com/phpMyAdmin
```

You should see the phpMyAdmin login page:



10. Log in to your phpMyAdmin installation with the `root` user name and the MySQL root password you created earlier.

Your installation must still be configured before you put it into service. To configure phpMyAdmin, you can [manually create a configuration file](#), [use the setup console](#), or combine both approaches.

For information about using phpMyAdmin, see the [phpMyAdmin User Guide](#).

Troubleshooting

This section offers suggestions for resolving common problems you may encounter while setting up a new LAMP server.

I can't connect to my server using a web browser.

Perform the following checks to see if your Apache web server is running and accessible.

- **Is the web server running?**

You can verify that **httpd** is on by running the following command:

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

If the **httpd** process is not running, repeat the steps described in [To prepare the LAMP server \(p. 37\)](#).

- **Is the firewall correctly configured?**

If you are unable to see the Apache test page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic. For information about adding an HTTP rule to your security group, see [Adding rules to a security group \(p. 1006\)](#).

Related topics

For more information about transferring files to your instance or installing a WordPress blog on your web server, see the following documentation:

- [Transferring files to your Linux instance using WinSCP \(p. 581\)](#)
- [Transferring files to Linux instances from Linux using SCP \(p. 567\)](#)
- [Tutorial: Hosting a WordPress blog with Amazon Linux \(p. 56\)](#)

For more information about the commands and software used in this tutorial, see the following webpages:

- Apache web server: <http://httpd.apache.org/>
- MariaDB database server: <https://mariadb.org/>
- PHP programming language: <http://php.net/>
- The `chmod` command: <https://en.wikipedia.org/wiki/Chmod>
- The `chown` command: <https://en.wikipedia.org/wiki/Chown>

For more information about registering a domain name for your web server, or transferring an existing domain name to this host, see [Creating and Migrating Domains and Subdomains to Amazon Route 53](#) in the *Amazon Route 53 Developer Guide*.

Tutorial: Install a LAMP web server with the Amazon Linux AMI

The following procedures help you install an Apache web server with PHP and MySQL support on your Amazon Linux instance (sometimes called a LAMP web server or LAMP stack). You can use this server to host a static website or deploy a dynamic PHP application that reads and writes information to a database.

Important

To set up a LAMP web server on Amazon Linux 2, see [Tutorial: Install a LAMP web server on Amazon Linux 2 \(p. 36\)](#).

If you are trying to set up a LAMP web server on an Ubuntu or Red Hat Enterprise Linux instance, this tutorial will not work for you. For more information about other distributions, see their specific documentation. For information about LAMP web servers on Ubuntu, see the Ubuntu community documentation [ApacheMySQLPHP](#) topic.

Option: Complete this tutorial using automation

To complete this tutorial using AWS Systems Manager Automation instead of the following tasks, run the [AWS Docs - Install a LAMP Server - AL](#) Automation document.

Tasks

- [Step 1: Prepare the LAMP server \(p. 46\)](#)
- [Step 2: Test your Lamp server \(p. 50\)](#)
- [Step 3: Secure the database server \(p. 51\)](#)
- [Step 4: \(Optional\) Install phpMyAdmin \(p. 52\)](#)
- [Troubleshooting \(p. 55\)](#)
- [Related topics \(p. 56\)](#)

Step 1: Prepare the LAMP server

Prerequisites

This tutorial assumes that you have already launched a new instance using the Amazon Linux AMI, with a public DNS name that is reachable from the internet. For more information, see [Step 1: Launch an instance \(p. 31\)](#). You must also have configured your security group to allow SSH (port 22), HTTP (port 80), and HTTPS (port 443) connections. For more information about these prerequisites, see [Authorizing inbound traffic for your Linux instances \(p. 983\)](#).

To install and start the LAMP web server with the Amazon Linux AMI

1. [Connect to your instance \(p. 32\)](#).
2. To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process may take a few minutes, but it is important to make sure that you have the latest security updates and bug fixes.

The `-y` option installs the updates without asking for confirmation. If you would like to examine the updates before installing, you can omit this option.

```
[ec2-user ~]$ sudo yum update -y
```

3. Now that your instance is current, you can install the Apache web server, MySQL, and PHP software packages.

Important

Some applications may not be compatible with the following recommended software environment. Before installing these packages, check whether your LAMP applications are compatible with them. If there is a problem, you may need to install an alternative environment. For more information, see [The application software I want to run on my server is incompatible with the installed PHP version or other software \(p. 55\)](#)

Use the **yum install** command to install multiple software packages and all related dependencies at the same time.

```
[ec2-user ~]$ sudo yum install -y httpd24 php72 mysql57-server php72-mysqlnd
```

If you receive the error `No package package-name available`, then your instance was not launched with the Amazon Linux AMI (perhaps you are using Amazon Linux 2 instead). You can view your version of Amazon Linux with the following command.

```
cat /etc/system-release
```

4. Start the Apache web server.

```
[ec2-user ~]$ sudo service httpd start
Starting httpd: [ OK ]
```

5. Use the **chkconfig** command to configure the Apache web server to start at each system boot.

```
[ec2-user ~]$ sudo chkconfig httpd on
```

The **chkconfig** command does not provide any confirmation message when you successfully use it to enable a service.

You can verify that **httpd** is on by running the following command:

```
[ec2-user ~]$ chkconfig --list httpd
httpd           0:off    1:off    2:on     3:on     4:on     5:on     6:off
```

Here, **httpd** is on in runlevels 2, 3, 4, and 5 (which is what you want to see).

6. Add a security rule to allow inbound HTTP (port 80) connections to your instance if you have not already done so. By default, a **launch-wizard-N** security group was set up for your instance during initialization. This group contains a single rule to allow SSH connections.
 - Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 - Choose **Instances** and select your instance.
 - On the **Security** tab, view the inbound rules. You should see the following rule:

Port range	Protocol	Source
22	tcp	0.0.0.0/0

- Choose the link for the security group. Using the procedures in [Adding rules to a security group \(p. 1006\)](#), add a new inbound security rule with the following values:
 - Type:** HTTP
 - Protocol:** TCP
 - Port Range:** 80
 - Source:** Custom

7. Test your web server. In a web browser, type the public DNS address (or the public IP address) of your instance. You can get the public DNS address for your instance using the Amazon EC2 console. If there is no content in `/var/www/html`, you should see the Apache test page. When you add content to the document root, your content appears at the public DNS address of your instance instead of this test page.

If you are unable to see the Apache test page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic. For information about adding an HTTP rule to your security group, see [Adding rules to a security group \(p. 1006\)](#).

If you are not using Amazon Linux, you may also need to configure the firewall on your instance to allow these connections. For more information about how to configure the firewall, see the documentation for your specific distribution.

Amazon Linux AMI Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly, but has not yet been configured.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting `www.example.com`, you should send e-mail to `"webmaster@example.com"`.

The [Amazon Linux AMI](#) is a supported and maintained Linux image provided by [Amazon Web Services](#) for use on [Amazon Elastic Compute Cloud \(Amazon EC2\)](#). It is designed to provide a stable, secure, and high performance execution environment for applications running on [Amazon EC2](#). It also includes packages that enable easy integration with [AWS](#), including launch configuration tools and many popular AWS libraries and tools. [Amazon Web Services](#) provides ongoing security and maintenance updates to all instances running the [Amazon Linux AMI](#). The [Amazon Linux AMI](#) is provided at no additional charge to [Amazon EC2](#) users.

If you are the website administrator:

You may now add content to the directory `/var/www/html`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the images below on Apache and Amazon Linux AMI powered HTTP servers. Thanks for using Apache and the Amazon Linux AMI!



Apache **httpd** serves files that are kept in a directory called the Apache document root. The Amazon Linux Apache document root is `/var/www/html`, which by default is owned by root.

```
[ec2-user ~]$ ls -l /var/www
total 16
drwxr-xr-x 2 root root 4096 Jul 12 01:00 cgi-bin
drwxr-xr-x 3 root root 4096 Aug  7 00:02 error
drwxr-xr-x 2 root root 4096 Jan  6 2012 html
```

```
drwxr-xr-x 3 root root 4096 Aug  7 00:02 icons
drwxr-xr-x 2 root root 4096 Aug  7 21:17 noindex
```

To allow the `ec2-user` account to manipulate files in this directory, you must modify the ownership and permissions of the directory. There are many ways to accomplish this task. In this tutorial, you add `ec2-user` to the `apache` group, to give the `apache` group ownership of the `/var/www` directory and assign write permissions to the group.

To set file permissions

1. Add your user (in this case, `ec2-user`) to the `apache` group.

```
[ec2-user ~]$ sudo usermod -a -G apache ec2-user
```

2. Log out and then log back in again to pick up the new group, and then verify your membership.
 - a. Log out (use the `exit` command or close the terminal window):

```
[ec2-user ~]$ exit
```

- b. To verify your membership in the `apache` group, reconnect to your instance, and then run the following command:

```
[ec2-user ~]$ groups
ec2-user wheel apache
```

3. Change the group ownership of `/var/www` and its contents to the `apache` group.

```
[ec2-user ~]$ sudo chown -R ec2-user:apache /var/www
```

4. To add group write permissions and to set the group ID on future subdirectories, change the directory permissions of `/var/www` and its subdirectories.

```
[ec2-user ~]$ sudo chmod 2775 /var/www
[ec2-user ~]$ find /var/www -type d -exec sudo chmod 2775 {} \;
```

5. To add group write permissions, recursively change the file permissions of `/var/www` and its subdirectories:

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

Now, `ec2-user` (and any future members of the `apache` group) can add, delete, and edit files in the Apache document root, enabling you to add content, such as a static website or a PHP application.

(Optional) Secure your web server

A web server running the HTTP protocol provides no transport security for the data that it sends or receives. When you connect to an HTTP server using a web browser, the URLs that you visit, the content of webpages that you receive, and the contents (including passwords) of any HTML forms that you submit are all visible to eavesdroppers anywhere along the network pathway. The best practice for securing your web server is to install support for HTTPS (HTTP Secure), which protects your data with SSL/TLS encryption.

For information about enabling HTTPS on your server, see [Tutorial: Configure SSL/TLS on Amazon Linux \(p. 80\)](#).

Step 2: Test your Lamp server

If your server is installed and running, and your file permissions are set correctly, your ec2-user account should be able to create a PHP file in the /var/www/html directory that is available from the internet.

To test your LAMP web server

1. Create a PHP file in the Apache document root.

```
[ec2-user ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

If you get a "Permission denied" error when trying to run this command, try logging out and logging back in again to pick up the proper group permissions that you configured in [Step 1: Prepare the LAMP server \(p. 46\)](#).

2. In a web browser, type the URL of the file that you just created. This URL is the public DNS address of your instance followed by a forward slash and the file name. For example:

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

You should see the PHP information page:

PHP Version 7.2.0

System	Linux ip-172-31-22-15.us-west-2.compute.internal 4.9.62-10.57.amzn2.x86_64
Build Date	Dec 13 2017 03:34:37
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mysqlind.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-session.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS

If you do not see this page, verify that the /var/www/html/phpinfo.php file was created properly in the previous step. You can also verify that all of the required packages were installed with the following command. The package versions in the second column do not need to match this example output.

```
[ec2-user ~]$ sudo yum list installed httpd24 php72 mysql57-server php72-mysqld  
Loaded plugins: priorities, update-motd, upgrade-helper  
Installed Packages
```

httpd24.x86_64	2.4.25-1.68.amzn1	@amzn-
updates		
mysql56-server.x86_64	5.6.35-1.23.amzn1	@amzn-
updates		
php70.x86_64	7.0.14-1.20.amzn1	@amzn-
updates		
php70-mysqlnd.x86_64	7.0.14-1.20.amzn1	@amzn-
updates		

If any of the required packages are not listed in your output, install them using the **sudo yum install package** command.

3. Delete the `phpinfo.php` file. Although this can be useful information, it should not be broadcast to the internet for security reasons.

```
[ec2-user ~]$ rm /var/www/html/phpinfo.php
```

Step 3: Secure the database server

The default installation of the MySQL server has several features that are great for testing and development, but they should be disabled or removed for production servers. The **mysql_secure_installation** command walks you through the process of setting a root password and removing the insecure features from your installation. Even if you are not planning on using the MySQL server, we recommend performing this procedure.

To secure the database server

1. Start the MySQL server.

```
[ec2-user ~]$ sudo service mysqld start
Initializing MySQL database:
...
PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
...
Starting mysqld: [ OK ]
```

2. Run **mysql_secure_installation**.

```
[ec2-user ~]$ sudo mysql_secure_installation
```

- a. When prompted, type a password for the root account.
 - i. Type the current root password. By default, the root account does not have a password set. Press Enter.
 - ii. Type **y** to set a password, and type a secure password twice. For more information about creating a secure password, see <https://identitysafe.norton.com/password-generator/>. Make sure to store this password in a safe place.

Setting a root password for MySQL is only the most basic measure for securing your database. When you build or install a database-driven application, you typically create a database service user for that application and avoid using the root account for anything but database administration.

- b. Type **y** to remove the anonymous user accounts.
- c. Type **y** to disable the remote root login.

- d. Type **y** to remove the test database.
- e. Type **y** to reload the privilege tables and save your changes.
3. (Optional) If you do not plan to use the MySQL server right away, stop it. You can restart it when you need it again.

```
[ec2-user ~]$ sudo service mysqld stop
Stopping mysqld:                                     [  OK  ]
```

4. (Optional) If you want the MySQL server to start at every boot, type the following command.

```
[ec2-user ~]$ sudo chkconfig mysqld on
```

You should now have a fully functional LAMP web server. If you add content to the Apache document root at `/var/www/html`, you should be able to view that content at the public DNS address for your instance.

Step 4: (Optional) Install phpMyAdmin

To install phpMyAdmin

[phpMyAdmin](#) is a web-based database management tool that you can use to view and edit the MySQL databases on your EC2 instance. Follow the steps below to install and configure phpMyAdmin on your Amazon Linux instance.

Important

We do not recommend using phpMyAdmin to access a LAMP server unless you have enabled SSL/TLS in Apache; otherwise, your database administrator password and other data are transmitted insecurely across the internet. For security recommendations from the developers, see [Securing your phpMyAdmin installation](#).

Note

The Amazon Linux package management system does not currently support the automatic installation of phpMyAdmin in a PHP 7 environment. This tutorial describes how to install phpMyAdmin manually.

1. Log in to your EC2 instance using SSH.
2. Install the required dependencies.

```
[ec2-user ~]$ sudo yum install php72-mbstring.x86_64 -y
```

3. Restart Apache.

```
[ec2-user ~]$ sudo service httpd restart
Stopping httpd:                                         [  OK  ]
Starting httpd:                                         [  OK  ]
```

4. Navigate to the Apache document root at `/var/www/html`.

```
[ec2-user ~]$ cd /var/www/html
[ec2-user html]$
```

5. Select a source package for the latest phpMyAdmin release from <https://www.phpmyadmin.net/downloads>. To download the file directly to your instance, copy the link and paste it into a `wget` command, as in this example:

```
[ec2-user html]$ wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz
```

6. Create a phpMyAdmin folder and extract the package into it using the following command.

```
[ec2-user html]$ mkdir phpMyAdmin && tar -xvzf phpMyAdmin-latest-all-languages.tar.gz -C phpMyAdmin --strip-components 1
```

7. Delete the *phpMyAdmin-latest-all-languages.tar.gz* tarball.

```
[ec2-user html]$ rm phpMyAdmin-latest-all-languages.tar.gz
```

8. (Optional) If the MySQL server is not running, start it now.

```
[ec2-user ~]$ sudo service mysqld start
Starting mysqld: [ OK ]
```

9. In a web browser, type the URL of your phpMyAdmin installation. This URL is the public DNS address (or the public IP address) of your instance followed by a forward slash and the name of your installation directory. For example:

```
http://my.public.dns.amazonaws.com/phpMyAdmin
```

You should see the phpMyAdmin login page:



10. Log in to your phpMyAdmin installation with the `root` user name and the MySQL root password you created earlier.

Your installation must still be configured before you put it into service. To configure phpMyAdmin, you can [manually create a configuration file](#), [use the setup console](#), or combine both approaches.

For information about using phpMyAdmin, see the [phpMyAdmin User Guide](#).

Troubleshooting

This section offers suggestions for resolving common problems you may encounter while setting up a new LAMP server.

I can't connect to my server using a web browser.

Perform the following checks to see if your Apache web server is running and accessible.

- **Is the web server running?**

You can verify that **httpd** is on by running the following command:

```
[ec2-user ~]$ chkconfig --list httpd
httpd           0:off    1:off    2:on     3:on    4:on    5:on    6:off
```

Here, **httpd** is on in runlevels 2, 3, 4, and 5 (which is what you want to see).

If the **httpd** process is not running, repeat the steps described in [Step 1: Prepare the LAMP server \(p. 46\)](#).

- **Is the firewall correctly configured?**

If you are unable to see the Apache test page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic. For information about adding an HTTP rule to your security group, see [Adding rules to a security group \(p. 1006\)](#).

The application software I want to run on my server is incompatible with the installed PHP version or other software

This tutorial recommends installing the most up-to-date versions of Apache HTTP Server, PHP, and MySQL. Before installing an additional LAMP application, check its requirements to confirm that it is compatible with your installed environment. If the latest version of PHP is not supported, it is possible (and entirely safe) to downgrade to an earlier supported configuration. You can also install more than one version of PHP in parallel, which solves certain compatibility problems with a minimum of effort. For information about configuring a preference among multiple installed PHP versions, see [Amazon Linux AMI 2016.09 Release Notes](#).

How to downgrade

The well-tested previous version of this tutorial called for the following core LAMP packages:

- **httpd24**
- **php56**
- **mysql55-server**
- **php56-mysqlnd**

If you have already installed the latest packages as recommended at the start of this tutorial, you must first uninstall these packages and other dependencies as follows:

```
[ec2-user ~]$ sudo yum remove -y httpd24 php72 mysql57-server php72-mysqlnd perl-DBD-MySQL57
```

Next, install the replacement environment:

```
[ec2-user ~]$ sudo yum install -y httpd24 php56 mysql55-server php56-mysqlnd
```

If you decide later to upgrade to the recommended environment, you must first remove the customized packages and dependencies:

```
[ec2-user ~]$ sudo yum remove -y httpd24 php56 mysql55-server php56-mysqlnd perl-DBD-MYSQL56
```

Now you can install the latest packages, as described earlier.

Related topics

For more information about transferring files to your instance or installing a WordPress blog on your web server, see the following documentation:

- [Transferring files to your Linux instance using WinSCP \(p. 581\)](#)
- [Transferring files to Linux instances from Linux using SCP \(p. 567\)](#)
- [Tutorial: Hosting a WordPress blog with Amazon Linux \(p. 56\)](#)

For more information about the commands and software used in this tutorial, see the following webpages:

- Apache web server: <http://httpd.apache.org/>
- MySQL database server: <http://www.mysql.com/>
- PHP programming language: <http://php.net/>
- The chmod command: <https://en.wikipedia.org/wiki/Chmod>
- The chown command: <https://en.wikipedia.org/wiki/Chown>

For more information about registering a domain name for your web server, or transferring an existing domain name to this host, see [Creating and Migrating Domains and Subdomains to Amazon Route 53](#) in the *Amazon Route 53 Developer Guide*.

Tutorial: Hosting a WordPress blog with Amazon Linux

The following procedures will help you install, configure, and secure a WordPress blog on your Amazon Linux instance. This tutorial is a good introduction to using Amazon EC2 in that you have full control over a web server that hosts your WordPress blog, which is not typical with a traditional hosting service.

You are responsible for updating the software packages and maintaining security patches for your server. For a more automated WordPress installation that does not require direct interaction with the web server configuration, the AWS CloudFormation service provides a WordPress template that can also get you started quickly. For more information, see [Get started](#) in the *AWS CloudFormation User Guide*. If you'd prefer to host your WordPress blog on a Windows instance, see [Deploying a WordPress blog on your Amazon EC2 Windows instance](#) in the *Amazon EC2 User Guide for Windows Instances*. If you need a high-availability solution with a decoupled database, see [Deploying a high-availability WordPress website](#) in the *AWS Elastic Beanstalk Developer Guide*.

Important

These procedures are intended for use with Amazon Linux. For more information about other distributions, see their specific documentation. Many steps in this tutorial do not work on

Ubuntu instances. For help installing WordPress on an Ubuntu instance, see [WordPress](#) in the Ubuntu documentation.

Option: Complete this tutorial using automation

To complete this tutorial using AWS Systems Manager Automation instead of the following tasks, run one of the following Automation documents: [AWS Docs - Hosting a WordPress Blog - AL](#) (Amazon Linux) or [AWS Docs - Hosting a WordPress Blog - AL2](#) (Amazon Linux 2).

Topics

- [Prerequisites \(p. 57\)](#)
- [Install WordPress \(p. 57\)](#)
- [Next steps \(p. 64\)](#)
- [Help! My public DNS name changed and now my blog is broken \(p. 64\)](#)

Prerequisites

This tutorial assumes that you have launched an Amazon Linux instance with a functional web server with PHP and database (either MySQL or MariaDB) support by following all of the steps in [Tutorial: Install a LAMP web server with the Amazon Linux AMI \(p. 46\)](#) for Amazon Linux AMI or [Tutorial: Install a LAMP web server on Amazon Linux 2 \(p. 36\)](#) for Amazon Linux 2. This tutorial also has steps for configuring a security group to allow HTTP and HTTPS traffic, as well as several steps to ensure that file permissions are set properly for your web server. For information about adding rules to your security group, see [Adding rules to a security group \(p. 1006\)](#).

We strongly recommend that you associate an Elastic IP address (EIP) to the instance you are using to host a WordPress blog. This prevents the public DNS address for your instance from changing and breaking your installation. If you own a domain name and you want to use it for your blog, you can update the DNS record for the domain name to point to your EIP address (for help with this, contact your domain name registrar). You can have one EIP address associated with a running instance at no charge. For more information, see [Elastic IP addresses \(p. 781\)](#).

If you don't already have a domain name for your blog, you can register a domain name with Route 53 and associate your instance's EIP address with your domain name. For more information, see [Registering domain names using Amazon Route 53](#) in the *Amazon Route 53 Developer Guide*.

Install WordPress

Connect to your instance, and download the WordPress installation package.

To download and unzip the WordPress installation package

1. Download the latest WordPress installation package with the `wget` command. The following command should always download the latest release.

```
[ec2-user ~]$ wget https://wordpress.org/latest.tar.gz
```

2. Unzip and unarchive the installation package. The installation folder is unzipped to a folder called `wordpress`.

```
[ec2-user ~]$ tar -xzf latest.tar.gz
```

To create a database user and database for your WordPress installation

Your WordPress installation needs to store information, such as blog posts and user comments, in a database. This procedure helps you create your blog's database and a user that is authorized to read and save information to it.

1. Start the database server.

- Amazon Linux 2

```
[ec2-user ~]$ sudo systemctl start mariadb
```

- Amazon Linux AMI

```
[ec2-user ~]$ sudo service mysqld start
```

2. Log in to the database server as the `root` user. Enter your database `root` password when prompted; this may be different than your `root` system password, or it may even be empty if you have not secured your database server.

If you have not secured your database server yet, it is important that you do so. For more information, see [To secure the MariaDB server \(p. 42\)](#) (Amazon Linux 2) or [To secure the database server \(p. 51\)](#) (Amazon Linux AMI).

```
[ec2-user ~]$ mysql -u root -p
```

3. Create a user and password for your MySQL database. Your WordPress installation uses these values to communicate with your MySQL database. Enter the following command, substituting a unique user name and password.

```
CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'your_strong_password';
```

Make sure that you create a strong password for your user. Do not use the single quote character (') in your password, because this will break the preceding command. For more information about creating a secure password, go to <http://www.pctools.com/guides/password/>. Do not reuse an existing password, and make sure to store this password in a safe place.

4. Create your database. Give your database a descriptive, meaningful name, such as `wordpress-db`.

Note

The punctuation marks surrounding the database name in the command below are called backticks. The backtick (`) key is usually located above the Tab key on a standard keyboard. Backticks are not always required, but they allow you to use otherwise illegal characters, such as hyphens, in database names.

```
CREATE DATABASE `wordpress-db`;
```

5. Grant full privileges for your database to the WordPress user that you created earlier.

```
GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpress-user"@"localhost";
```

6. Flush the database privileges to pick up all of your changes.

```
FLUSH PRIVILEGES;
```

7. Exit the `mysql` client.

```
exit
```

To create and edit the wp-config.php file

The WordPress installation folder contains a sample configuration file called `wp-config-sample.php`. In this procedure, you copy this file and edit it to fit your specific configuration.

1. Copy the `wp-config-sample.php` file to a file called `wp-config.php`. This creates a new configuration file and keeps the original sample file intact as a backup.

```
[ec2-user ~]$ cp wordpress/wp-config-sample.php wordpress/wp-config.php
```

2. Edit the `wp-config.php` file with your favorite text editor (such as `nano` or `vim`) and enter values for your installation. If you do not have a favorite text editor, `nano` is suitable for beginners.

```
[ec2-user ~]$ nano wordpress/wp-config.php
```

- a. Find the line that defines `DB_NAME` and change `database_name_here` to the database name that you created in [Step 4 \(p. 58\)](#) of [To create a database user and database for your WordPress installation \(p. 58\)](#).

```
define('DB_NAME', 'wordpress-db');
```

- b. Find the line that defines `DB_USER` and change `username_here` to the database user that you created in [Step 3 \(p. 58\)](#) of [To create a database user and database for your WordPress installation \(p. 58\)](#).

```
define('DB_USER', 'wordpress-user');
```

- c. Find the line that defines `DB_PASSWORD` and change `password_here` to the strong password that you created in [Step 3 \(p. 58\)](#) of [To create a database user and database for your WordPress installation \(p. 58\)](#).

```
define('DB_PASSWORD', 'your_strong_password');
```

- d. Find the section called `Authentication Unique Keys and Salts`. These `KEY` and `SALT` values provide a layer of encryption to the browser cookies that WordPress users store on their local machines. Basically, adding long, random values here makes your site more secure. Visit <https://api.wordpress.org/secret-key/1.1/salt/> to randomly generate a set of key values that you can copy and paste into your `wp-config.php` file. To paste text into a PuTTY terminal, place the cursor where you want to paste the text and right-click your mouse inside the PuTTY terminal.

For more information about security keys, go to <https://wordpress.org/support/article/editing-wp-config-php/#security-keys>.

Note

The values below are for example purposes only; do not use these values for your installation.

```
define('AUTH_KEY', '#U$$+[RXN8:b^-L_0(WU_+ c+WFKI-c]o]-bHw+);  
Aj[wTwSiz<Qb[mghEXcRh-');  
define('SECURE_AUTH_KEY', 'szs._P=l/|y.Lq)XjlkwS1y5NJ76E6EJ.AV0pCKZZB, *~*r ?6OP  
$eJT@;+(ndIg');  
define('LOGGED_IN_KEY', 'ju}qwre3V*+8f_zOWf?{LlGsQ]Ye@2Jh^,8x>)Y_|;(^[Iw]Pi  
+LG#A4R?7N`YB3');  
define('NONCE_KEY', 'P(g62HeZxEes/LnI^i=H,[XwK9I&[2s|:?ON}VJM%?;v2v]v+;  
+^9eXUhg@:@Cj');  
define('AUTH_SALT', 'C$DpB4Hj[JK:{ql`sRVa:{:7yShy(9A@5wg+`JJVb1fk%-  
Bx*M4(qc[Qg%JT!h');
```

```
define('SECURE_AUTH_SALT', 'd!uRu#+q#{f$Z?Z9uFPG.${+S{n-1M&%@~gL>U>NV<zpD-@2-Es7Q1O-bp28EKv'};  
define('LOGGED_IN_SALT', 'j{00P*owZf)kVD+FVLn-->.|Y%Ug4#I^*Lvd9QeZ^&XmK/e(76miC+&W+^OP/');  
define('NONCE_SALT', '-97r*V/cgxLmp?Zy4zUU4r99QO_rGs2LTd%P;|_e1tS)8_B/, .6[=UK<J_y9?JWG');
```

- e. Save the file and exit your text editor.

To install your WordPress files under the Apache document root

- Now that you've unzipped the installation folder, created a MySQL database and user, and customized the WordPress configuration file, you are ready to copy your installation files to your web server document root so you can run the installation script that completes your installation. The location of these files depends on whether you want your WordPress blog to be available at the actual root of your web server (for example, my.public.dns.amazonaws.com) or in a subdirectory or folder under the root (for example, my.public.dns.amazonaws.com/blog).
 - If you want WordPress to run at your document root, copy the contents of the wordpress installation directory (but not the directory itself) as follows:

```
[ec2-user ~]$ cp -r wordpress/* /var/www/html/
```

- If you want WordPress to run in an alternative directory under the document root, first create that directory, and then copy the files to it. In this example, WordPress will run from the directory blog:

```
[ec2-user ~]$ mkdir /var/www/html/blog  
[ec2-user ~]$ cp -r wordpress/* /var/www/html/blog/
```

Important

For security purposes, if you are not moving on to the next procedure immediately, stop the Apache web server (`httpd`) now. After you move your installation under the Apache document root, the WordPress installation script is unprotected and an attacker could gain access to your blog if the Apache web server were running. To stop the Apache web server, enter the command `sudo service httpd stop`. If you are moving on to the next procedure, you do not need to stop the Apache web server.

To allow WordPress to use permalinks

WordPress permalinks need to use Apache `.htaccess` files to work properly, but this is not enabled by default on Amazon Linux. Use this procedure to allow all overrides in the Apache document root.

1. Open the `httpd.conf` file with your favorite text editor (such as `nano` or `vim`). If you do not have a favorite text editor, `nano` is suitable for beginners.

```
[ec2-user ~]$ sudo vim /etc/httpd/conf/httpd.conf
```

2. Find the section that starts with `<Directory "/var/www/html">`.

```
<Directory "/var/www/html">  
#  
# Possible values for the Options directive are "None", "All",  
# or any combination of:  
#   Indexes Includes FollowSymLinks SymLinksIfOwnerMatch ExecCGI MultiViews  
#  
# Note that "MultiViews" must be named *explicitly* --- "Options All"  
# doesn't give it to you.
```

```
#  
# The Options directive is both complicated and important. Please see  
# http://httpd.apache.org/docs/2.4/mod/core.html#options  
# for more information.  
#  
Options Indexes FollowSymLinks  
  
#  
# AllowOverride controls what directives may be placed in .htaccess files.  
# It can be "All", "None", or any combination of the keywords:  
#   Options FileInfo AuthConfig Limit  
#  
AllowOverride None  
  
#  
# Controls who can get stuff from this server.  
#  
Require all granted  
</Directory>
```

3. Change the `AllowOverride None` line in the above section to read `AllowOverride All`.

Note

There are multiple `AllowOverride` lines in this file; be sure you change the line in the `<Directory "/var/www/html">` section.

```
AllowOverride All
```

4. Save the file and exit your text editor.

To install the PHP graphics drawing library on Amazon Linux 2

The GD library for PHP enables you to modify images. Install this library if you need to crop the header image for your blog. The version of phpMyAdmin that you install might require a specific minimum version of this library (for example, version 7.2).

Use the following command to install the PHP graphics drawing library on Amazon Linux 2. For example, if you installed php7.2 from amazon-linux-extras as part of installing the LAMP stack, this command installs version 7.2 of the PHP graphics drawing library.

```
[ec2-user ~]$ sudo yum install php-gd
```

To verify the installed version, use the following command:

```
[ec2-user ~]$ sudo yum list installed | grep php-gd
```

The following is example output:

php-gd.x86_64	7.2.30-1.amzn2	@amzn2extra-php7.2
---------------	----------------	--------------------

To install the PHP graphics drawing library on the Amazon Linux AMI

The GD library for PHP enables you to modify images. Install this library if you need to crop the header image for your blog. The version of phpMyAdmin that you install might require a specific minimum version of this library (for example, version 7.2).

To verify which versions are available, use the following command:

```
[ec2-user ~]$ yum list | grep php-gd
```

The following is an example line from the output for the PHP graphics drawing library (version 7.2):

php72-gd.x86_64	7.2.30-1.22.amzn1	amzn-updates
-----------------	-------------------	--------------

Use the following command to install a specific version of the PHP graphics drawing library (for example, version 7.2) on the Amazon Linux AMI:

[ec2-user ~]\$ sudo yum install php72-gd

To fix file permissions for the Apache web server

Some of the available features in WordPress require write access to the Apache document root (such as uploading media through the Administration screens). If you have not already done so, apply the following group memberships and permissions (as described in greater detail in the [LAMP web server tutorial \(p. 46\)](#)).

1. Grant file ownership of /var/www and its contents to the apache user.

[ec2-user ~]\$ sudo chown -R apache /var/www

2. Grant group ownership of /var/www and its contents to the apache group.

[ec2-user ~]\$ sudo chgrp -R apache /var/www

3. Change the directory permissions of /var/www and its subdirectories to add group write permissions and to set the group ID on future subdirectories.

[ec2-user ~]\$ sudo chmod 2775 /var/www [ec2-user ~]\$ find /var/www -type d -exec sudo chmod 2775 {} \;

4. Recursively change the file permissions of /var/www and its subdirectories to add group write permissions.

[ec2-user ~]\$ find /var/www -type f -exec sudo chmod 0664 {} \;

5. Restart the Apache web server to pick up the new group and permissions.

- Amazon Linux 2

[ec2-user ~]\$ sudo systemctl restart httpd

- Amazon Linux AMI

[ec2-user ~]\$ sudo service httpd restart

To run the WordPress installation script with Amazon Linux 2

You are ready to install WordPress. The commands that you use depend on the operating system. The commands in this procedure are for use with Amazon Linux 2. Use the procedure that follows this one with Amazon Linux AMI.

1. Use the **systemctl** command to ensure that the **httpd** and database services start at every system boot.

[ec2-user ~]\$ sudo systemctl enable httpd && sudo systemctl enable mariadb

2. Verify that the database server is running.

```
[ec2-user ~]$ sudo systemctl status mariadb
```

If the database service is not running, start it.

```
[ec2-user ~]$ sudo systemctl start mariadb
```

3. Verify that your Apache web server (`httpd`) is running.

```
[ec2-user ~]$ sudo systemctl status httpd
```

If the `httpd` service is not running, start it.

```
[ec2-user ~]$ sudo systemctl start httpd
```

4. In a web browser, type the URL of your WordPress blog (either the public DNS address for your instance, or that address followed by the `blog` folder). You should see the WordPress installation script. Provide the information required by the WordPress installation. Choose **Install WordPress** to complete the installation. For more information, see [Step 5: Run the Install Script](#) on the WordPress website.

To run the WordPress installation script with Amazon Linux AMI

1. Use the `chkconfig` command to ensure that the `httpd` and database services start at every system boot.

```
[ec2-user ~]$ sudo chkconfig httpd on && sudo chkconfig mysqld on
```

2. Verify that the database server is running.

```
[ec2-user ~]$ sudo service mysqld status
```

If the database service is not running, start it.

```
[ec2-user ~]$ sudo service mysqld start
```

3. Verify that your Apache web server (`httpd`) is running.

```
[ec2-user ~]$ sudo service httpd status
```

If the `httpd` service is not running, start it.

```
[ec2-user ~]$ sudo service httpd start
```

4. In a web browser, type the URL of your WordPress blog (either the public DNS address for your instance, or that address followed by the `blog` folder). You should see the WordPress installation script. Provide the information required by the WordPress installation. Choose **Install WordPress** to complete the installation. For more information, see [Step 5: Run the Install Script](#) on the WordPress website.

Next steps

After you have tested your WordPress blog, consider updating its configuration.

Use a custom domain name

If you have a domain name associated with your EC2 instance's EIP address, you can configure your blog to use that name instead of the EC2 public DNS address. For more information, see [Changing The Site URL](#) on the WordPress website.

Configure your blog

You can configure your blog to use different [themes](#) and [plugins](#) to offer a more personalized experience for your readers. However, sometimes the installation process can backfire, causing you to lose your entire blog. We strongly recommend that you create a backup Amazon Machine Image (AMI) of your instance before attempting to install any themes or plugins so you can restore your blog if anything goes wrong during installation. For more information, see [Creating your own AMI \(p. 98\)](#).

Increase capacity

If your WordPress blog becomes popular and you need more compute power or storage, consider the following steps:

- Expand the storage space on your instance. For more information, see [Amazon EBS Elastic Volumes \(p. 1109\)](#).
- Move your MySQL database to [Amazon RDS](#) to take advantage of the service's ability to scale easily.
- Migrate to a larger instance type. For more information, see [Changing the instance type \(p. 294\)](#).
- Add additional instances. For more information, see [Tutorial: Increase the availability of your application on Amazon EC2 \(p. 93\)](#).

Learn more about WordPress

For information about WordPress, see the WordPress Codex help documentation at <http://codex.wordpress.org/>. For more information about troubleshooting your installation, go to <https://wordpress.org/support/article/how-to-install-wordpress/#common-installation-problems>. For information about making your WordPress blog more secure, go to <https://wordpress.org/support/article/hardening-wordpress/>. For information about keeping your WordPress blog up-to-date, go to <https://wordpress.org/support/article/updating-wordpress/>.

Help! My public DNS name changed and now my blog is broken

Your WordPress installation is automatically configured using the public DNS address for your EC2 instance. If you stop and restart the instance, the public DNS address changes (unless it is associated with an Elastic IP address) and your blog will not work anymore because it references resources at an address that no longer exists (or is assigned to another EC2 instance). A more detailed description of the problem and several possible solutions are outlined in <https://wordpress.org/support/article/changing-the-site-url/>.

If this has happened to your WordPress installation, you may be able to recover your blog with the procedure below, which uses the `wp-cli` command line interface for WordPress.

To change your WordPress site URL with the wp-cli

1. Connect to your EC2 instance with SSH.
2. Note the old site URL and the new site URL for your instance. The old site URL is likely the public DNS name for your EC2 instance when you installed WordPress. The new site URL is the current

public DNS name for your EC2 instance. If you are not sure of your old site URL, you can use `curl` to find it with the following command.

```
[ec2-user ~]$ curl localhost | grep wp-content
```

You should see references to your old public DNS name in the output, which will look like this (old site URL in red):

```
<script type='text/javascript' src='http://ec2-52-8-139-223.us-west-1.compute.amazonaws.com/wp-content/themes/twentyfifteen/js/functions.js?ver=20150330'></script>
```

3. Download the `wp-cli` with the following command.

```
[ec2-user ~]$ curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
```

4. Search and replace the old site URL in your WordPress installation with the following command. Substitute the old and new site URLs for your EC2 instance and the path to your WordPress installation (usually `/var/www/html` or `/var/www/html/blog`).

```
[ec2-user ~]$ php wp-cli.phar search-replace 'old_site_url' 'new_site_url' --path=/path/to/wordpress/installation --skip-columns=guid
```

5. In a web browser, enter the new site URL of your WordPress blog to verify that the site is working properly again. If it is not, see <https://wordpress.org/support/article/changing-the-site-url/> and <https://wordpress.org/support/article/how-to-install-wordpress/#common-installation-problems> for more information.

Tutorial: Configure SSL/TLS on Amazon Linux 2

Secure Sockets Layer/Transport Layer Security (SSL/TLS) creates an encrypted channel between a web server and web client that protects data in transit from being eavesdropped on. This tutorial explains how to add support manually for SSL/TLS on an EC2 instance with Amazon Linux 2 and Apache web server. If you plan to offer commercial-grade services, the [AWS Certificate Manager](#), which is not discussed here, is a good option.

For historical reasons, web encryption is often referred to simply as SSL. While web browsers still support SSL, its successor protocol TLS is less vulnerable to attack. Amazon Linux 2 disables server-side support for all versions of SSL by default. [Security standards bodies](#) consider TLS 1.0 to be unsafe, and both TLS 1.0 and TLS 1.1 are on track to be formally [deprecated](#) by the IETF. This tutorial contains guidance based exclusively on enabling TLS 1.2. (A newer TLS 1.3 protocol exists, but it is not installed by default on Amazon Linux 2.) For more information about the updated encryption standards, see [RFC 7568](#) and [RFC 8446](#).

This tutorial refers to modern web encryption simply as TLS.

Important

These procedures are intended for use with Amazon Linux 2. We also assume that you are starting with a fresh Amazon EC2 instance. If you are trying to set up a LAMP web server on an instance with a different distribution, or if you are reusing an older, existing instance, some procedures in this tutorial might not work for you. For information about LAMP web servers on Ubuntu, see the Ubuntu community documentation [ApacheMySQLPHP](#). For information about Red Hat Enterprise Linux, see the Customer Portal topic [Web Servers](#).

Contents

- [Prerequisites \(p. 66\)](#)

- [Step 1: Enable TLS on the server \(p. 66\)](#)
- [Step 2: Obtain a CA-signed certificate \(p. 68\)](#)
- [Step 3: Test and harden the security configuration \(p. 73\)](#)
- [Troubleshooting \(p. 75\)](#)
- [Certificate automation: Let's Encrypt with Certbot on Amazon Linux 2 \(p. 76\)](#)

Prerequisites

Before you begin this tutorial, complete the following steps:

- Launch an EBS-backed Amazon Linux 2 instance. For more information, see [Step 1: Launch an instance \(p. 31\)](#).
- Configure your security groups to allow your instance to accept connections on the following TCP ports:
 - SSH (port 22)
 - HTTP (port 80)
 - HTTPS (port 443)

For more information, see [Authorizing inbound traffic for your Linux instances \(p. 983\)](#).

- Install the Apache web server. For step-by-step instructions, see [Tutorial: Install a LAMP Web Server on Amazon Linux 2 \(p. 36\)](#). Only the httpd package and its dependencies are needed, so you can ignore the instructions involving PHP and MariaDB.
- To identify and authenticate websites, the TLS public key infrastructure (PKI) relies on the Domain Name System (DNS). To use your EC2 instance to host a public website, you need to register a domain name for your web server or transfer an existing domain name to your Amazon EC2 host. Numerous third-party domain registration and DNS hosting services are available for this, or you can use [Amazon Route 53](#).

Step 1: Enable TLS on the server

This procedure takes you through the process of setting up TLS on Amazon Linux 2 with a self-signed digital certificate.

Note

A self-signed certificate is acceptable for testing but not production. If you expose your self-signed certificate to the internet, visitors to your site are greeted by security warnings.

To enable TLS on a server

1. [Connect to your instance \(p. 32\)](#) and confirm that Apache is running.

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

If the returned value is not "enabled," start Apache and set it to start each time the system boots.

```
[ec2-user ~]$ sudo systemctl start httpd && sudo systemctl enable httpd
```

2. To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process may take a few minutes, but it is important to make sure that you have the latest security updates and bug fixes.

Note

The `-y` option installs the updates without asking for confirmation. If you would like to examine the updates before installing, you can omit this option.

```
[ec2-user ~]$ sudo yum update -y
```

- Now that your instance is current, add TLS support by installing the Apache module `mod_ssl`.

```
[ec2-user ~]$ sudo yum install -y mod_ssl
```

Your instance now has the following files that you use to configure your secure server and create a certificate for testing:

- `/etc/httpd/conf.d/ssl.conf`

The configuration file for `mod_ssl`. It contains *directives* telling Apache where to find encryption keys and certificates, the TLS protocol versions to allow, and the encryption ciphers to accept.

- `/etc/pki/tls/certs/make-dummy-cert`

A script to generate a self-signed X.509 certificate and private key for your server host. This certificate is useful for testing that Apache is properly set up to use TLS. Because it offers no proof of identity, it should not be used in production. If used in production, it triggers warnings in Web browsers.

- Run the script to generate a self-signed dummy certificate and key for testing.

```
[ec2-user ~]$ cd /etc/pki/tls/certs  
sudo ./make-dummy-cert localhost.crt
```

This generates a new file `localhost.crt` in the `/etc/pki/tls/certs/` directory. The specified file name matches the default that is assigned in the `SSLCertificateFile` directive in `/etc/httpd/conf.d/ssl.conf`.

This file contains both a self-signed certificate and the certificate's private key. Apache requires the certificate and key to be in PEM format, which consists of Base64-encoded ASCII characters framed by "BEGIN" and "END" lines, as in the following abbreviated example.

```
-----BEGIN PRIVATE KEY-----  
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQD2KKx/8Zk94m1q  
3gQMZF9ZN6Ls19+3tHagQ5Fpo9KJDbhzLjOOC18u1PTcGmAah5kEitCEc0wzmNeo  
BC10wYR6G0rGaKtK9Dn7CuIjvubtUysVyQoMVPQ97ldeakHWeRMiEJFXg6kZZ0vr  
GvwnKoMh3DlK44D9dx7IDua2PlYx5+eroA+1Lqf32ZSaAOObBIMIYTHigwbHMz0T  
...  
56tE7THvH7vOEf4/iUOsIrEzaMaJ0mqkmY1A70qQGQKBgBF3H1qNPNHuyMcPODFs  
27hDzPDinrqusEv0ZIggkDMlh2irTipJ/GhkvtPq0l1v0fK/VXw8vSgeaBuhwJvS  
LXU9HvYq0U604FgD3nayB9hI0BE13r1HjUvbjT7moH+RhnNz6eqqdscCS09VtRAo  
4QQvAqOa8UheYeoXLdWcHaLP  
-----END PRIVATE KEY-----  
  
-----BEGIN CERTIFICATE-----  
MIIEazCCA1OgAwIBAgICWxQwDQYJKoZIhvCNQELBQAwgbExCzAJBgNVBAYTAi0t  
MRIwEAYDVQQIDA1Tb211U3RhGUxETAPBgNVBAcMCFNvbWVDaXR5MRkwFwYDVQOK  
DBBTb211T3JnYW5pemF0aW9uMR8wHQYDVQQLDBZTb211T3JnYW5pemF0aW9uYWxv  
bml0MRkwFwYDVQODDBBpcC0xNzItMzEtMjAtMjM2MSQwIgYJKoZIhvCNQkBFhVv  
...  
z5rRUE/XzxRLBZ0oWZpNWTXJkQ3uFYH6s/sBwtHpKKZMzOvDedREjNKAvk4ws6F0  
CuIjvubtUysVyQoMVPQ97ldeakHWeRMiEJFXg6kZZ0vrGvwnKoMh3DlK44D9d1U3  
WanXWehT6FiSzvB4sTEXXJN2jdW8g+sHgnZ8zCosclknYhRxCVD2vnBlZJKSzvak  
3ZazhBxtQSukFMOnWPP2a0DMMFGYUHod0BQE8sBjxg==
```

-----END CERTIFICATE-----

The file names and extensions are a convenience and have no effect on function. For example, you can call a certificate `cert.crt`, `cert.pem`, or any other file name, so long as the related directive in the `ssl.conf` file uses the same name.

Note

When you replace the default TLS files with your own customized files, be sure that they are in PEM format.

5. Open the `/etc/httpd/conf.d/ssl.conf` file and comment out the following line, because the self-signed dummy certificate also contains the key. If you do not comment out this line before you complete the next step, the Apache service fails to start.

```
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
```

6. Restart Apache.

```
[ec2-user ~]$ sudo systemctl restart httpd
```

Note

Make sure that TCP port 443 is accessible on your EC2 instance, as previously described.

7. Your Apache web server should now support HTTPS (secure HTTP) over port 443. Test it by entering the IP address or fully qualified domain name of your EC2 instance into a browser URL bar with the prefix `https://`.

Because you are connecting to a site with a self-signed, untrusted host certificate, your browser may display a series of security warnings. Override the warnings and proceed to the site.

If the default Apache test page opens, it means that you have successfully configured TLS on your server. All data passing between the browser and server is now encrypted.

Note

To prevent site visitors from encountering warning screens, you must obtain a trusted, CA-signed certificate that not only encrypts, but also publicly authenticates you as the owner of the site.

Step 2: Obtain a CA-signed certificate

You can use the following process to obtain a CA-signed certificate:

- Generate a certificate signing request (CSR) from a private key
- Submit the CSR to a certificate authority (CA)
- Obtain a signed host certificate
- Configure Apache to use the certificate

A self-signed TLS X.509 host certificate is cryptologically identical to a CA-signed certificate. The difference is social, not mathematical. A CA promises, at a minimum, to validate a domain's ownership before issuing a certificate to an applicant. Each web browser contains a list of CAs trusted by the browser vendor to do this. An X.509 certificate consists primarily of a public key that corresponds to your private server key, and a signature by the CA that is cryptographically tied to the public key. When a browser connects to a web server over HTTPS, the server presents a certificate for the browser to check against its list of trusted CAs. If the signer is on the list, or accessible through a *chain of trust* consisting of other trusted signers, the browser negotiates a fast encrypted data channel with the server and loads the page.

Important

Certificates generally cost money because of the labor involved in validating the requests, so it pays to shop around. A list of well-known CAs can be found at [dmoztols.net](#). A few CAs offer basic-level certificates free of charge. The most notable of these CAs is the [Let's Encrypt](#) project, which also supports the automation of the certificate creation and renewal process. For more information about using Let's Encrypt as your CA, see [Certificate automation: Let's Encrypt with Certbot on Amazon Linux 2 \(p. 76\)](#).

Underlying the host certificate is the key. As of 2019, [government](#) and [industry](#) groups recommend using a minimum key (modulus) size of 2048 bits for RSA keys intended to protect documents, through 2030. The default modulus size generated by OpenSSL in Amazon Linux 2 is 2048 bits, which is suitable for use in a CA-signed certificate. In the following procedure, an optional step provided for those who want a customized key, for example, one with a larger modulus or using a different encryption algorithm.

These instructions for acquiring a CA-signed host certificate do not work unless you own a registered and hosted DNS domain.

To obtain a CA-signed certificate

1. [Connect to your instance \(p. 32\)](#) and navigate to `/etc/pki/tls/private/`. This is the directory where you store the server's private key for TLS. If you prefer to use an existing host key to generate the CSR, skip to Step 3.
2. (Optional) Generate a new private key. Here are some examples of key configurations. Any of the resulting keys works with your web server, but they vary in the degree and type of security that they implement.
 - **Example 1:** Create a default RSA host key. The resulting file, `custom.key`, is a 2048-bit RSA private key.

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key
```

- **Example 2:** Create a stronger RSA key with a bigger modulus. The resulting file, `custom.key`, is a 4096-bit RSA private key.

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key 4096
```

- **Example 3:** Create a 4096-bit encrypted RSA key with password protection. The resulting file, `custom.key`, is a 4096-bit RSA private key encrypted with the AES-128 cipher.

Important

Encrypting the key provides greater security, but because an encrypted key requires a password, services depending on it cannot be auto-started. Each time you use this key, you must supply the password (in the preceding example, "abcde12345") over an SSH connection.

```
[ec2-user ~]$ sudo openssl genrsa -aes128 -passout pass:abcde12345 -out custom.key 4096
```

- **Example 4:** Create a key using a non-RSA cipher. RSA cryptography can be relatively slow because of the size of its public keys, which are based on the product of two large prime numbers. However, it is possible to create keys for TLS that use non-RSA ciphers. Keys based on the mathematics of elliptic curves are smaller and computationally faster when delivering an equivalent level of security.

```
[ec2-user ~]$ sudo openssl ecparam -name prime256v1 -out custom.key -genkey
```

The result is a 256-bit elliptic curve private key using prime256v1, a "named curve" that OpenSSL supports. Its cryptographic strength is slightly greater than a 2048-bit RSA key, [according to NIST](#).

Note

Not all CAs provide the same level of support for elliptic-curve-based keys as for RSA keys.

Make sure that the new private key has highly restrictive ownership and permissions (owner=root, group=root, read/write for owner only). The commands would be as shown in the following example.

```
[ec2-user ~]$ sudo chown root:root custom.key
[ec2-user ~]$ sudo chmod 600 custom.key
[ec2-user ~]$ ls -al custom.key
```

The preceding commands yield the following result.

```
-rw----- root root custom.key
```

After you have created and configured a satisfactory key, you can create a CSR.

3. Create a CSR using your preferred key. The following example uses **custom.key**.

```
[ec2-user ~]$ sudo openssl req -new -key custom.key -out csr.pem
```

OpenSSL opens a dialog and prompts you for the information shown in the following table. All of the fields except **Common Name** are optional for a basic, domain-validated host certificate.

Name	Description	Example
Country Name	The two-letter ISO abbreviation for your country.	US (=United States)
State or Province Name	The name of the state or province where your organization is located. This name cannot be abbreviated.	Washington
Locality Name	The location of your organization, such as a city.	Seattle
Organization Name	The full legal name of your organization. Do not abbreviate your organization name.	Example Corporation
Organizational Unit Name	Additional organizational information, if any.	Example Dept
Common Name	This value must exactly match the web address that you expect users to enter into a browser. Usually, this means a domain name with a prefixed hostname or alias in the form www.example.com . In testing with a self-signed certificate and no DNS resolution, the common name may consist of the hostname alone. CAs also offer more expensive certificates that accept wild-card names such as *.example.com .	www.example.com
Email Address	The server administrator's email address.	someone@example.com

Finally, OpenSSL prompts you for an optional challenge password. This password applies only to the CSR and to transactions between you and your CA, so follow the CA's recommendations about this and the other optional field, optional company name. The CSR challenge password has no effect on server operation.

The resulting file `csr.pem` contains your public key, your digital signature of your public key, and the metadata that you entered.

4. Submit the CSR to a CA. This usually consists of opening your CSR file in a text editor and copying the contents into a web form. At this time, you may be asked to supply one or more subject alternate names (SANs) to be placed on the certificate. If `www.example.com` is the common name, then `example.com` would be a good SAN, and vice versa. A visitor to your site entering either of these names would see an error-free connection. If your CA web form allows it, include the common name in the list of SANs. Some CAs include it automatically.

After your request has been approved, you receive a new host certificate signed by the CA. You might also be instructed to download an *intermediate certificate* file that contains additional certificates needed to complete the CA's chain of trust.

Note

Your CA might send you files in multiple formats intended for various purposes. For this tutorial, you should only use a certificate file in PEM format, which is usually (but not always) marked with a `.pem` or `.crt` file extension. If you are uncertain which file to use, open the files with a text editor and find the one containing one or more blocks beginning with the following line.

```
-- - - - -BEGIN CERTIFICATE - - - - -
```

The file should also end with the following line.

```
- - - - -END CERTIFICATE - - - - -
```

You can also test the file at the command line as shown in the following.

```
[ec2-user certs]$ openssl x509 -in certificate.crt -text
```

Verify that these lines appear in the file. Do not use files ending with `.p7b`, `.p7c`, or similar file extensions.

5. Place the new CA-signed certificate and any intermediate certificates in the `/etc/pki/tls/certs` directory.

Note

There are several ways to upload your new certificate to your EC2 instance, but the most straightforward and informative way is to open a text editor (for example, vi, nano, or notepad) on both your local computer and your instance, and then copy and paste the file contents between them. You need root [sudo] permissions when performing these operations on the EC2 instance. This way, you can see immediately if there are any permission or path problems. Be careful, however, not to add any additional lines while copying the contents, or to change them in any way.

From inside the `/etc/pki/tls/certs` directory, check that the file ownership, group, and permission settings match the highly restrictive Amazon Linux 2 defaults (owner=root, group=root, read/write for owner only). The following example shows the commands to use.

```
[ec2-user certs]$ sudo chown root:root custom.crt
[ec2-user certs]$ sudo chmod 600 custom.crt
```

```
[ec2-user certs]$ ls -al custom.crt
```

These commands should yield the following result.

```
-rw----- root root custom.crt
```

The permissions for the intermediate certificate file are less stringent (owner=root, group=root, owner can write, group can read, world can read). The following example shows the commands to use.

```
[ec2-user certs]$ sudo chown root:root intermediate.crt
[ec2-user certs]$ sudo chmod 644 intermediate.crt
[ec2-user certs]$ ls -al intermediate.crt
```

These commands should yield the following result.

```
-rw-r--r-- root root intermediate.crt
```

6. Place the private key that you used to create the CSR in the /etc/pki/tls/private/ directory.

Note

There are several ways to upload your custom key to your EC2 instance, but the most straightforward and informative way is to open a text editor (for example, vi, nano, or notepad) on both your local computer and your instance, and then copy and paste the file contents between them. You need root [sudo] permissions when performing these operations on the EC2 instance. This way, you can see immediately if there are any permission or path problems. Be careful, however, not to add any additional lines while copying the contents, or to change them in any way.

From inside the /etc/pki/tls/private directory, use the following commands to verify that the file ownership, group, and permission settings match the highly restrictive Amazon Linux 2 defaults (owner=root, group=root, read/write for owner only).

```
[ec2-user private]$ sudo chown root:root custom.key
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ ls -al custom.key
```

These commands should yield the following result.

```
-rw----- root root custom.key
```

7. Edit /etc/httpd/conf.d/ssl.conf to reflect your new certificate and key files.

- a. Provide the path and file name of the CA-signed host certificate in Apache's SSLCertificateFile directive:

```
SSLCertificateFile /etc/pki/tls/certs/custom.crt
```

- b. If you received an intermediate certificate file (intermediate.crt in this example), provide its path and file name using Apache's SSLCACertificateFile directive:

```
SSLCACertificateFile /etc/pki/tls/certs/intermediate.crt
```

Note

Some CAs combine the host certificate and the intermediate certificates in a single file, making the `SSLCACertificateFile` directive unnecessary. Consult the instructions provided by your CA.

- c. Provide the path and file name of the private key (`custom.key` in this example) in Apache's `SSLCertificateKeyFile` directive:

```
SSLCertificateKeyFile /etc/pki/tls/private/custom.key
```

8. Save `/etc/httpd/conf.d/ssl.conf` and restart Apache.

```
[ec2-user ~]$ sudo systemctl restart httpd
```

9. Test your server by entering your domain name into a browser URL bar with the prefix `https://`. Your browser should load the test page over HTTPS without generating errors.

Step 3: Test and harden the security configuration

After your TLS is operational and exposed to the public, you should test how secure it really is. This is easy to do using online services such as [Qualys SSL Labs](#), which performs a free and thorough analysis of your security setup. Based on the results, you may decide to harden the default security configuration by controlling which protocols you accept, which ciphers you prefer, and which you exclude. For more information, see [how Qualys formulates its scores](#).

Important

Real-world testing is crucial to the security of your server. Small configuration errors may lead to serious security breaches and loss of data. Because recommended security practices change constantly in response to research and emerging threats, periodic security audits are essential to good server administration.

On the [Qualys SSL Labs](#) site, enter the fully qualified domain name of your server, in the form `www.example.com`. After about two minutes, you receive a grade (from A to F) for your site and a detailed breakdown of the findings. The following table summarizes the report for a domain with settings identical to the default Apache configuration on Amazon Linux 2, and with a default Certbot certificate.

Overall rating	B
Certificate	100%
Protocol support	95%
Key exchange	70%
Cipher strength	90%

Though the overview shows that the configuration is mostly sound, the detailed report flags several potential problems, listed here in order of severity:

X The RC4 cipher is supported for use by certain older browsers. A cipher is the mathematical core of an encryption algorithm. RC4, a fast cipher used to encrypt TLS data-streams, is known to have several [serious weaknesses](#). Unless you have very good reasons to support legacy browsers, you should disable this.

X Old TLS versions are supported. The configuration supports TLS 1.0 (already deprecated) and TLS 1.1 (on a path to deprecation). Only TLS 1.2 has been recommended since 2018.

X Forward secrecy is not fully supported. [Forward secrecy](#) is a feature of algorithms that encrypt using temporary (ephemeral) session keys derived from the private key. This means in practice that attackers cannot decrypt HTTPS data even if they possess a web server's long-term private key.

To correct and future-proof the TLS configuration

1. Open the configuration file `/etc/httpd/conf.d/ssl.conf` in a text editor and comment out the following line by entering "#" at the beginning of the line.

```
#SSLProtocol all -SSLv3
```

2. Add the following directive:

```
#SSLProtocol all -SSLv3  
SSLProtocol -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 +TLSv1.2
```

This directive explicitly disables SSL versions 2 and 3, as well as TLS versions 1.0 and 1.1. The server now refuses to accept encrypted connections with clients using anything except TLS 1.2. The verbose wording in the directive conveys more clearly, to a human reader, what the server is configured to do.

Note

Disabling TLS versions 1.0 and 1.1 in this manner blocks a small percentage of outdated web browsers from accessing your site.

To modify the list of allowed ciphers

1. In the configuration file `/etc/httpd/conf.d/ssl.conf`, find the section with the `SSLCipherSuite` directive and comment out the existing line by entering "#" at the beginning of the line.

```
#SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5
```

2. Specify explicit cipher suites and a cipher order that prioritizes forward secrecy and avoids insecure ciphers. The `SSLCipherSuite` directive used here is based on output from the [Mozilla SSL Configuration Generator](#), which tailors a TLS configuration to the specific software running on your server. (For more information, see Mozilla's useful resource [Server Side TLS](#).) First determine your Apache and OpenSSL versions by using the output from the following commands.

```
[ec2-user ~]$ yum list installed | grep httpd  
[ec2-user ~]$ yum list installed | grep openssl
```

For example, if the returned information is Apache 2.4.34 and OpenSSL 1.0.2, we enter this into the generator. If you choose the "modern" compatibility model, this creates an `SSLCipherSuite` directive that aggressively enforces security but still works for most browsers. If your software doesn't support the modern configuration, you can update your software or choose the "intermediate" configuration instead.

```
SSLCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-  
CHACHA20-POLY1305:  
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:  
ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-  
AES128-SHA256
```

The selected ciphers have *ECDHE* in their names, an abbreviation for *Elliptic Curve Diffie-Hellman Ephemeral*. The term *ephemeral* indicates forward secrecy. As a by-product, these ciphers do not support RC4.

We recommend that you use an explicit list of ciphers instead of relying on defaults or terse directives whose content isn't visible.

Copy the generated directive into `/etc/httpd/conf.d/ssl.conf`.

Note

Though shown here on several lines for readability, the directive must be on a single line when copied to `/etc/httpd/conf.d/ssl.conf`, with only a colon (no spaces) between cipher names.

- Finally, uncomment the following line by removing the "#" at the beginning of the line.

```
#SSLHonorCipherOrder on
```

This directive forces the server to prefer high-ranking ciphers, including (in this case) those that support forward secrecy. With this directive turned on, the server tries to establish a strong secure connection before falling back to allowed ciphers with lesser security.

After completing both of these procedures, save the changes to `/etc/httpd/conf.d/ssl.conf` and restart Apache.

If you test the domain again on [Qualys SSL Labs](#), you should see that the RC4 vulnerability and other warnings are gone and the summary looks something like the following.

Overall rating	A
Certificate	100%
Protocol support	100%
Key exchange	90%
Cipher strength	90%

Important

Each update to OpenSSL introduces new ciphers and removes support for old ones. Keep your EC2 Amazon Linux 2 instance up-to-date, watch for security announcements from [OpenSSL](#), and be alert to reports of new security exploits in the technical press. For more information, see [Predefined SSL Security Policies for Elastic Load Balancing](#) in the *User Guide for Classic Load Balancers*.

Troubleshooting

- **My Apache webserver doesn't start unless I supply a password**

This is expected behavior if you installed an encrypted, password-protected, private server key.

You can remove the encryption and password requirement from the key. Assuming that you have a private encrypted RSA key called `custom.key` in the default directory, and that the password on it is `abcde12345`, run the following commands on your EC2 instance to generate an unencrypted version of the key.

```
[ec2-user ~]$ cd /etc/pki/tls/private/  
[ec2-user private]$ sudo cp custom.key custom.key.bak  
[ec2-user private]$ sudo openssl rsa -in custom.key -passin pass:abcde12345 -out  
    custom.key.nocrypt  
[ec2-user private]$ sudo mv custom.key.nocrypt custom.key  
[ec2-user private]$ sudo chown root:root custom.key  
[ec2-user private]$ sudo chmod 600 custom.key  
[ec2-user private]$ sudo systemctl restart httpd
```

Apache should now start without prompting you for a password.

- **I get errors when I run `sudo yum install -y mod_ssl`.**

When you are installing the required packages for SSL, you may see errors similar to the following.

```
Error: httpd24-tools conflicts with httpd-tools-2.2.34-1.16.amzn1.x86_64  
Error: httpd24 conflicts with httpd-2.2.34-1.16.amzn1.x86_64
```

This typically means that your EC2 instance is not running Amazon Linux 2. This tutorial only supports instances freshly created from an official Amazon Linux 2 AMI.

Certificate automation: Let's Encrypt with Certbot on Amazon Linux 2

The [Let's Encrypt](#) certificate authority is the centerpiece of an effort by the Electronic Frontier Foundation (EFF) to encrypt the entire internet. In line with that goal, Let's Encrypt host certificates are designed to be created, validated, installed, and maintained with minimal human intervention. The automated aspects of certificate management are carried out by a software agent running on your web server. After you install and configure the agent, it communicates securely with Let's Encrypt and performs administrative tasks on Apache and the key management system. This tutorial uses the free [Certbot](#) agent because it allows you either to supply a customized encryption key as the basis for your certificates, or to allow the agent itself to create a key based on its defaults. You can also configure Certbot to renew your certificates on a regular basis without human interaction, as described in [To automate Certbot \(p. 79\)](#). For more information, consult the Certbot [User Guide](#) and [man page](#).

Certbot is not officially supported on Amazon Linux 2, but is available for download and functions correctly when installed. We recommend that you make the following backups to protect your data and avoid inconvenience:

- Before you begin, take a snapshot of your Amazon EBS root volume. This allows you to restore the original state of your EC2 instance. For information about creating EBS snapshots, see [Creating Amazon EBS snapshots \(p. 1063\)](#).
- The procedure below requires you to edit your `httpd.conf` file, which controls Apache's operation. Certbot makes its own automated changes to this and other configuration files. Make a backup copy of your entire `/etc/httpd` directory in case you need to restore it.

Prepare to install

Complete the following procedures before you install Certbot.

1. Download the Extra Packages for Enterprise Linux (EPEL) 7 repository packages. These are required to supply dependencies needed by Certbot.

-
- a. Navigate to your home directory (/home/ec2-user). Download EPEL with the following command.

```
[ec2-user ~]$ sudo wget -r --no-parent -A 'epel-release-*' http://dl.fedoraproject.org/pub/epel/7/x86_64/Packages/e/
```

- b. Install the repository packages as shown in the following command.

```
[ec2-user ~]$ sudo rpm -Uvh dl.fedoraproject.org/pub/epel/7/x86_64/Packages/e/epel-release-*.
```

- c. Enable EPEL as shown in the following command.

```
[ec2-user ~]$ sudo yum-config-manager --enable epel*
```

You can confirm that EPEL is enabled with the following command. It should return information similar to the following.

```
[ec2-user ~]$ sudo yum repolist all

...
epel/x86_64           Extra Packages for Enterprise Linux 7 - x86_64
enabled: 12949+175
epel-debuginfo/x86_64   Extra Packages for Enterprise Linux 7 - x86_64
enabled:      2890
epel-source/x86_64     Extra Packages for Enterprise Linux 7 - x86_64
enabled:          0
epel-testing/x86_64    Extra Packages for Enterprise Linux 7 -
Testing - x86_64        enabled:    778+12
epel-testing-debuginfo/x86_64 Extra Packages for Enterprise Linux 7 -
Testing - x86_64 - Debug enabled:      107
epel-testing-source/x86_64 Extra Packages for Enterprise Linux 7 -
Testing - x86_64 - Source enabled:          0
...
```

2. Edit the main Apache configuration file, /etc/httpd/conf/httpd.conf. Locate the "Listen 80" directive and add the following lines after it, replacing the example domain names with the actual Common Name and Subject Alternative Name (SAN).

```
<VirtualHost *:80>
  DocumentRoot "/var/www/html"
  ServerName "example.com"
  ServerAlias "www.example.com"
</VirtualHost>
```

Save the file and restart Apache.

```
[ec2-user ~]$ sudo systemctl restart httpd
```

Install and run Certbot

This procedure is based on the EFF documentation for installing Certbot on [Fedora](#) and on [RHEL 7](#). It describes the default use of Certbot, resulting in a certificate based on a 2048-bit RSA key.

1. Install Certbot packages and dependencies using the following command.

Amazon Elastic Compute Cloud
User Guide for Linux Instances
Certificate automation: Let's Encrypt
with Certbot on Amazon Linux 2

```
[ec2-user ~]$ sudo yum install -y certbot python2-certbot-apache
```

2. Run Certbot.

```
[ec2-user ~]$ sudo certbot
```

3. At the prompt "Enter email address (used for urgent renewal and security notices)," enter a contact address and press Enter.
4. Agree to the Let's Encrypt Terms of Service at the prompt. Enter "A" and press Enter to proceed.

```
-----  
Please read the Terms of Service at  
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must  
agree in order to register with the ACME server at  
https://acme-v02.api.letsencrypt.org/directory  
-----  
(A)gree/(C)ancel: A
```

5. At the authorization for EFF to put you on their mailing list, enter "Y" or "N" and press Enter.
6. Certbot displays the Common Name and Subject Alternative Name (SAN) that you provided in the VirtualHost block.

```
Which names would you like to activate HTTPS for?  
-----  
1: example.com  
2: www.example.com  
-----  
Select the appropriate numbers separated by commas and/or spaces, or leave input  
blank to select all options shown (Enter 'c' to cancel):
```

Leave the input blank and press Enter.

7. Certbot displays the following output as it creates certificates and configures Apache. It then prompts you about redirecting HTTP queries to HTTPS.

```
Obtaining a new certificate  
Performing the following challenges:  
http-01 challenge for example.com  
http-01 challenge for www.example.com  
Waiting for verification...  
Cleaning up challenges  
Created an SSL vhost at /etc/httpd/conf/httpd-le-ssl.conf  
Deploying Certificate for example.com to VirtualHost /etc/httpd/conf/httpd-le-ssl.conf  
Enabling site /etc/httpd/conf/httpd-le-ssl.conf by adding Include to root configuration  
Deploying Certificate for www.example.com to VirtualHost /etc/httpd/conf/httpd-le-ssl.conf  
  
Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.  
-----  
1: No redirect - Make no further changes to the webserver configuration.  
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for  
new sites, or if you're confident your site works on HTTPS. You can undo this  
change by editing your web server's configuration.  
-----  
Select the appropriate number [1-2] then [enter] (press 'c' to cancel):
```

To allow visitors to connect to your server via unencrypted HTTP, enter "1". If you want to accept only encrypted connections via HTTPS, enter "2". Press Enter to submit your choice.

8. Certbot completes the configuration of Apache and reports success and other information.

```
Congratulations! You have successfully enabled https://example.com and  
https://www.example.com
```

```
You should test your configuration at:  
https://www.ssllabs.com/ssltest/analyze.html?d=example.com  
https://www.ssllabs.com/ssltest/analyze.html?d=www.example.com
```

IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at:
`/etc/letsencrypt/live/certbot.oneeyedman.net/fullchain.pem`
Your key file has been saved at:
`/etc/letsencrypt/live/certbot.oneeyedman.net/privkey.pem`
Your cert will expire on 2019-08-01. To obtain a new or tweaked
version of this certificate in the future, simply run certbot again
with the "certonly" option. To non-interactively renew *all* of
your certificates, run "certbot renew"
- Your account credentials have been saved in your Certbot
configuration directory at `/etc/letsencrypt`. You should make a
secure backup of this folder now. This configuration directory will
also contain certificates and private keys obtained by Certbot so
making regular backups of this folder is ideal.

9. After you complete the installation, test and optimize the security of your server as described in [Step 3: Test and harden the security configuration \(p. 73\)](#).

Configure automated certificate renewal

Certbot is designed to become an invisible, error-resistant part of your server system. By default, it generates host certificates with a short, 90-day expiration time. If you have not configured your system to call the command automatically, you must re-run the `certbot` command manually before expiration. This procedure shows how to automate Certbot by setting up a cron job.

To automate Certbot

1. Open the `/etc/crontab` file in a text editor, such as `vim` or `nano`, using `sudo`. Alternatively, use `sudo crontab -e`.
2. Add a line similar to the following and save the file.

```
39      1,13    *      *      *      root    certbot renew --no-self-upgrade
```

Here is an explanation of each component:

```
39 1,13 * * *
```

Schedules a command to be run at 01:39 and 13:39 every day. The selected values are arbitrary, but the Certbot developers suggest running the command at least twice daily. This guarantees that any certificate found to be compromised is promptly revoked and replaced.

```
root
```

The command runs with root permissions.

```
certbot renew --no-self-upgrade
```

The command to be run. The `renew` subcommand causes Certbot to check any previously obtained certificates and to renew those that are approaching expiration. The `--no-self-upgrade` flag prevents Certbot from upgrading itself without your intervention.

3. Restart the cron daemon.

```
[ec2-user ~]$ sudo systemctl restart crond
```

Tutorial: Configure SSL/TLS on Amazon Linux

Secure Sockets Layer/Transport Layer Security (SSL/TLS) creates an encrypted channel between a web server and web client that protects data in transit from being eavesdropped on. This tutorial explains how to add support manually for SSL/TLS on an EC2 instance with the Amazon Linux AMI and Apache web server. If you plan to offer commercial-grade services, the [AWS Certificate Manager](#), which is not discussed here, is a good option.

For historical reasons, web encryption is often referred to simply as SSL. While web browsers still support SSL, its successor protocol TLS is less vulnerable to attack. The Amazon Linux AMI disables server-side support all versions of SSL by default. [Security standards bodies](#) consider TLS 1.0 to be unsafe, and both TLS 1.0 and TLS 1.1 are on track to be formally [deprecated](#) by the IETF. This tutorial contains guidance based exclusively on enabling TLS 1.2. (A newer TLS 1.3 protocol exists in draft form, but is not supported on Amazon Linux.) For more information about the updated encryption standards, see [RFC 7568](#) and [RFC 8446](#).

This tutorial refers to modern web encryption simply as TLS.

Important

These procedures are intended for use with the Amazon Linux AMI. If you are trying to set up a LAMP web server on an instance with a different distribution, some procedures in this tutorial might not work for you. For information about LAMP web servers on Ubuntu, see the Ubuntu community documentation [ApacheMySQLPHP](#). For information about Red Hat Enterprise Linux, see the Customer Portal documentation [Web Servers](#).

Contents

- [Prerequisites \(p. 80\)](#)
- [Step 1: Enable TLS on the server \(p. 81\)](#)
- [Step 2: Obtain a CA-signed certificate \(p. 82\)](#)
- [Step 3: Test and harden the security configuration \(p. 87\)](#)
- [Troubleshooting \(p. 89\)](#)
- [Certificate automation: Let's Encrypt with Certbot on Amazon Linux \(p. 89\)](#)

Prerequisites

Before you begin this tutorial, complete the following steps:

- Launch an EBS-backed instance using the Amazon Linux AMI. For more information, see [Step 1: Launch an instance \(p. 31\)](#).
- Configure your security group to allow your instance to accept connections on the following TCP ports:
 - SSH (port 22)
 - HTTP (port 80)
 - HTTPS (port 443)

For more information, see [Authorizing inbound traffic for your Linux instances \(p. 983\)](#).

- Install Apache web server. For step-by-step instructions, see [Tutorial: Installing a LAMP Web Server on Amazon Linux \(p. 46\)](#). Only the http24 package and its dependencies are needed; you can ignore the instructions involving PHP and MySQL.

- To identify and authenticate web sites, the TLS public key infrastructure (PKI) relies on the Domain Name System (DNS). To use your EC2 instance to host a public web site, you need to register a domain name for your web server or transfer an existing domain name to your Amazon EC2 host. Numerous third-party domain registration and DNS hosting services are available for this, or you can use [Amazon Route 53](#).

Step 1: Enable TLS on the server

This procedure takes you through the process of setting up TLS on Amazon Linux with a self-signed digital certificate.

Note

A self-signed certificate is acceptable for testing but not production. If you expose your self-signed certificate to the internet, visitors to your site receive security warnings.

To enable TLS on a server

1. [Connect to your instance \(p. 32\)](#) and confirm that Apache is running.

```
[ec2-user ~]$ sudo service httpd status
```

If necessary, start Apache.

```
[ec2-user ~]$ sudo service httpd start
```

2. To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process may take a few minutes, but it is important to make sure you have the latest security updates and bug fixes.

Note

The `-y` option installs the updates without asking for confirmation. If you would like to examine the updates before installing, you can omit this option.

```
[ec2-user ~]$ sudo yum update -y
```

3. Now that your instance is current, add TLS support by installing the Apache module `mod_ssl`:

```
[ec2-user ~]$ sudo yum install -y mod24_ssl
```

Your instance now has the following files that you use to configure your secure server and create a certificate for testing:

`/etc/httpd/conf.d/ssl.conf`

The configuration file for `mod_ssl`. It contains "directives" telling Apache where to find encryption keys and certificates, the TLS protocol versions to allow, and the encryption ciphers to accept.

`/etc/pki/tls/private/localhost.key`

An automatically generated, 2048-bit RSA private key for your Amazon EC2 host. During installation, OpenSSL used this key to generate a self-signed host certificate, and you can also use this key to generate a certificate signing request (CSR) to submit to a certificate authority (CA).

/etc/pki/tls/certs/localhost.crt

An automatically generated, self-signed X.509 certificate for your server host. This certificate is useful for testing that Apache is properly set up to use TLS.

The .key and .crt files are both in PEM format, which consists of Base64-encoded ASCII characters framed by "BEGIN" and "END" lines, as in this abbreviated example of a certificate:

```
-----BEGIN CERTIFICATE-----  
MIIEazCCA1OgAwIBAgICWxQwDQYJKoZIhvcNAQELBQAwbExCzAJBgNVBAYTAi0t  
MRIwEAYDVQQIDAlTb21lU3RhGUxETAPBgNVBAcMCFNvbWVDaXR5MRkwFwYDVQQK  
DBBTb21lT3JnYW5pemF0aW9uMR8wHQYDVQQLDBZtb21lT3JnYW5pemF0aW9uYWxV  
bmlOMRkwFwYDVQQDDBBpcC0xNzItMzEtMjAtMjM2MSQwIgYJKoZIhvcNAQkBFhVv  
...  
z5rRUE/XzxRLBZOOwZpNWTXJkQ3uFYH6s/sBwtHpKKZMzOvDedREjNKAvk4ws6F0  
WanXWehT6FiSzvB4sTEXXJN2jdw8g+sHGnZ8zCoSclknYhHrCVD2vnBlZJKSzvak  
3ZazhBxtQSukFMOnWPP2a0DMMFGYUHod0BQE8sBJxg==  
-----END CERTIFICATE-----
```

The file names and extensions are a convenience and have no effect on function; you can call a certificate cert.crt, cert.pem, or any other file name, so long as the related directive in the ssl.conf file uses the same name.

Note

When you replace the default TLS files with your own customized files, be sure that they are in PEM format.

4. Restart Apache.

```
[ec2-user ~]$ sudo service httpd restart
```

5. Your Apache web server should now support HTTPS (secure HTTP) over port 443. Test it by typing the IP address or fully qualified domain name of your EC2 instance into a browser URL bar with the prefix **https://**. Because you are connecting to a site with a self-signed, untrusted host certificate, your browser may display a series of security warnings.

Override the warnings and proceed to the site. If the default Apache test page opens, it means that you have successfully configured TLS on your server. All data passing between the browser and server is now safely encrypted.

To prevent site visitors from encountering warning screens, you need to obtain a certificate that not only encrypts, but also publicly authenticates you as the owner of the site.

Step 2: Obtain a CA-signed certificate

You can use the following process to obtain a CA-signed certificate:

- Generate a certificate signing request (CSR) from a private key
- Submit the CSR to a certificate authority (CA)
- Obtain a signed host certificate
- Configure Apache to use the certificate

A self-signed TLS X.509 host certificate is cryptologically identical to a CA-signed certificate. The difference is social, not mathematical; a CA promises to validate, at a minimum, a domain's ownership before issuing a certificate to an applicant. Each web browser contains a list of CAs trusted by the browser vendor to do this. An X.509 certificate consists primarily of a public key that corresponds to

your private server key, and a signature by the CA that is cryptographically tied to the public key. When a browser connects to a web server over HTTPS, the server presents a certificate for the browser to check against its list of trusted CAs. If the signer is on the list, or accessible through a chain of trust consisting of other trusted signers, the browser negotiates a fast encrypted data channel with the server and loads the page.

Certificates generally cost money because of the labor involved in validating the requests, so it pays to shop around. A list of well-known CAs can be found at [dmoztools.net](#). A few CAs offer basic-level certificates free of charge. The most notable of these is the [Let's Encrypt](#) project, which also supports automation of the certificate creation and renewal process. For more information about using Let's Encrypt as your CA, see [Certificate automation: Let's Encrypt with Certbot on Amazon Linux \(p. 89\)](#).

Underlying the host certificate is the key. As of 2017, [government](#) and [industry](#) groups recommend using a minimum key (modulus) size of 2048 bits for RSA keys intended to protect documents through 2030. The default modulus size generated by OpenSSL in Amazon Linux is 2048 bits, which means that the existing auto-generated key is suitable for use in a CA-signed certificate. An alternative procedure is described below for those who desire a customized key, for instance, one with a larger modulus or using a different encryption algorithm.

These instructions for acquiring a CA-signed host certificate do not work unless you own a registered and hosted DNS domain.

To obtain a CA-signed certificate

1. [Connect to your instance \(p. 32\)](#) and navigate to `/etc/pki/tls/private/`. This is the directory where the server's private key for TLS is stored. If you prefer to use your existing host key to generate the CSR, skip to Step 3.
2. (Optional) Generate a new private key. Here are some examples of key configurations. Any of the resulting keys work with your web server, but they vary in how (and how much) security they implement.
 - **Example 1:** Create a default RSA host key. The resulting file, `custom.key`, is a 2048-bit RSA private key.

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key
```

- **Example 2:** Create a stronger RSA key with a bigger modulus. The resulting file, `custom.key`, is a 4096-bit RSA private key.

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key 4096
```

- **Example 3:** Create a 4096-bit encrypted RSA key with password protection. The resulting file, `custom.key`, is a 4096-bit RSA private key encrypted with the AES-128 cipher.

Important

Encrypting the key provides greater security, but because an encrypted key requires a password, services depending on it cannot be auto-started. Each time you use this key, you must supply the password (in the preceding example, "abcde12345") over an SSH connection.

```
[ec2-user ~]$ sudo openssl genrsa -aes128 -passout pass:abcde12345 -out custom.key 4096
```

- **Example 4:** Create a key using a non-RSA cipher. RSA cryptography can be relatively slow because of the size of its public keys, which are based on the product of two large prime numbers. However, it is possible to create keys for TLS that use non-RSA ciphers. Keys based on the mathematics of elliptic curves are smaller and computationally faster when delivering an equivalent level of security.

```
[ec2-user ~]$ sudo openssl ecparam -name prime256v1 -out custom.key -genkey
```

The result is a 256-bit elliptic curve private key using prime256v1, a "named curve" that OpenSSL supports. Its cryptographic strength is slightly greater than a 2048-bit RSA key, according to NIST.

Note

Not all CAs provide the same level of support for elliptic-curve-based keys as for RSA keys.

Make sure that the new private key has highly restrictive ownership and permissions (owner=root, group=root, read/write for owner only). The commands would be as follows:

```
[ec2-user ~]$ sudo chown root.root custom.key
[ec2-user ~]$ sudo chmod 600 custom.key
[ec2-user ~]$ ls -al custom.key
```

The commands above should yield the following result:

```
-rw----- root root custom.key
```

After you have created and configured a satisfactory key, you can create a CSR.

3. Create a CSR using your preferred key; the example below uses **custom.key**:

```
[ec2-user ~]$ sudo openssl req -new -key custom.key -out csr.pem
```

OpenSSL opens a dialog and prompts you for the information shown in the following table. All of the fields except **Common Name** are optional for a basic, domain-validated host certificate.

Name	Description	Example
Country Name	The two-letter ISO abbreviation for your country.	US (=United States)
State or Province Name	The name of the state or province where your organization is located. This name cannot be abbreviated.	Washington
Locality Name	The location of your organization, such as a city.	Seattle
Organization Name	The full legal name of your organization. Do not abbreviate your organization name.	Example Corporation
Organizational Unit Name	Additional organizational information, if any.	Example Dept
Common Name	This value must exactly match the web address that you expect users to type into a browser. Usually, this means a domain name with a prefixed host name or alias in the form www.example.com . In testing with a self-signed certificate and no DNS resolution, the common name may consist of the host name alone. CAs also offer more expensive	www.example.com

Name	Description	Example
	certificates that accept wild-card names such as *.example.com .	
Email Address	The server administrator's email address.	someone@example.com

Finally, OpenSSL prompts you for an optional challenge password. This password applies only to the CSR and to transactions between you and your CA, so follow the CA's recommendations about this and the other optional field, optional company name. The CSR challenge password has no effect on server operation.

The resulting file **csr.pem** contains your public key, your digital signature of your public key, and the metadata that you entered.

4. Submit the CSR to a CA. This usually consists of opening your CSR file in a text editor and copying the contents into a web form. At this time, you may be asked to supply one or more subject alternate names (SANs) to be placed on the certificate. If **www.example.com** is the common name, then **example.com** would be a good SAN, and vice versa. A visitor to your site typing in either of these names would see an error-free connection. If your CA web form allows it, include the common name in the list of SANs. Some CAs include it automatically.

After your request has been approved, you receive a new host certificate signed by the CA. You might also be instructed to download an *intermediate certificate* file that contains additional certificates needed to complete the CA's chain of trust.

Note

Your CA may send you files in multiple formats intended for various purposes. For this tutorial, you should only use a certificate file in PEM format, which is usually (but not always) marked with a **.pem** or **.crt** extension. If you are uncertain which file to use, open the files with a text editor and find the one containing one or more blocks beginning with the following:

```
- - - - -BEGIN CERTIFICATE - - - - -
```

The file should also end with the following:

```
- - - - -END CERTIFICATE - - - - -
```

You can also test a file at the command line as follows:

```
[ec2-user certs]$ openssl x509 -in certificate.crt -text
```

Verify that these lines appear in the file. Do not use files ending with **.p7b**, **.p7c**, or similar file extensions.

5. Place the new CA-signed certificate and any intermediate certificates in the **/etc/pki/tls/certs** directory.

Note

There are several ways to upload your custom key to your EC2 instance, but the most straightforward and informative way is to open a text editor (for example, vi, nano, or notepad) on both your local computer and your instance, and then copy and paste the file contents between them. You need root [sudo] permissions when performing these operations on the EC2 instance. This way, you can see immediately if there are any permission or path problems. Be careful, however, not to add any additional lines while copying the contents, or to change them in any way.

From inside the `/etc/pki/tls/certs` directory, use the following commands to verify that the file ownership, group, and permission settings match the highly restrictive Amazon Linux defaults (owner=root, group=root, read/write for owner only).

```
[ec2-user certs]$ sudo chown root.root custom.crt
[ec2-user certs]$ sudo chmod 600 custom.crt
[ec2-user certs]$ ls -al custom.crt
```

The commands above should yield the following result:

```
-rw----- root root custom.crt
```

The permissions for the intermediate certificate file are less stringent (owner=root, group=root, owner can write, group can read, world can read). The commands would be:

```
[ec2-user certs]$ sudo chown root.root intermediate.crt
[ec2-user certs]$ sudo chmod 644 intermediate.crt
[ec2-user certs]$ ls -al intermediate.crt
```

The commands above should yield the following result:

```
-rw-r--r-- root root intermediate.crt
```

6. If you used a custom key to create your CSR and the resulting host certificate, remove or rename the old key from the `/etc/pki/tls/private/` directory, and then install the new key there.

Note

There are several ways to upload your custom key to your EC2 instance, but the most straightforward and informative way is to open a text editor (vi, nano, notepad, etc.) on both your local computer and your instance, and then copy and paste the file contents between them. You need root [sudo] privileges when performing these operations on the EC2 instance. This way, you can see immediately if there are any permission or path problems. Be careful, however, not to add any additional lines while copying the contents, or to change them in any way.

From inside the `/etc/pki/tls/private` directory, check that the file ownership, group, and permission settings match the highly restrictive Amazon Linux defaults (owner=root, group=root, read/write for owner only). The commands would be as follows:

```
[ec2-user private]$ sudo chown root.root custom.key
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ ls -al custom.key
```

The commands above should yield the following result:

```
-rw----- root root custom.key
```

7. Edit `/etc/httpd/conf.d/ssl.conf` to reflect your new certificate and key files.
 - a. Provide the path and file name of the CA-signed host certificate in Apache's `SSLCertificateFile` directive:

```
SSLCertificateFile /etc/pki/tls/certs/custom.crt
```

- b. If you received an intermediate certificate file (`intermediate.crt` in this example), provide its path and file name using Apache's `SSLCACertificateFile` directive:

```
SSLCACertificateFile /etc/pki/tls/certs/intermediate.crt
```

Note

Some CAs combine the host certificate and the intermediate certificates in a single file, making this directive unnecessary. Consult the instructions provided by your CA.

- c. Provide the path and file name of the private key in Apache's `SSLCertificateKeyFile` directive:

```
SSLCertificateKeyFile /etc/pki/tls/private/custom.key
```

8. Save `/etc/httpd/conf.d/ssl.conf` and restart Apache.

```
[ec2-user ~]$ sudo service httpd restart
```

9. Test your server by entering your domain name into a browser URL bar with the prefix `https://`. Your browser should load the test page over HTTPS without generating errors.

Step 3: Test and harden the security configuration

After your TLS is operational and exposed to the public, you should test how secure it really is. This is easy to do using online services such as [Qualys SSL Labs](#), which performs a free and thorough analysis of your security setup. Based on the results, you may decide to harden the default security configuration by controlling which protocols you accept, which ciphers you prefer, and which you exclude. For more information, see [how Qualys formulates its scores](#).

Important

Real-world testing is crucial to the security of your server. Small configuration errors may lead to serious security breaches and loss of data. Because recommended security practices change constantly in response to research and emerging threats, periodic security audits are essential to good server administration.

On the [Qualys SSL Labs](#) site, type the fully qualified domain name of your server, in the form `www.example.com`. After about two minutes, you receive a grade (from A to F) for your site and a detailed breakdown of the findings. Though the overview shows that the configuration is mostly sound, the detailed report flags several potential problems. For example:

X The RC4 cipher is supported for use by certain older browsers. A cipher is the mathematical core of an encryption algorithm. RC4, a fast cipher used to encrypt TLS data-streams, is known to have several [serious weaknesses](#). Unless you have very good reasons to support legacy browsers, you should disable this.

X Old TLS versions are supported. The configuration supports TLS 1.0 (already deprecated) and TLS 1.1 (on a path to deprecation). Only TLS 1.2 has been recommended since 2018.

To correct the TLS configuration

1. Open the configuration file `/etc/httpd/conf.d/ssl.conf` in a text editor and comment out the following lines by typing "#" at the beginning of each:

```
#SSLProtocol all -SSLv3  
#SSLProxyProtocol all -SSLv3
```

2. Add the following directives:

```
SSLProtocol -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 +TLSv1.2
SSLProxyProtocol -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 +TLSv1.2
```

These directives explicitly disable SSL versions 2 and 3, as well as TLS versions 1.0 and 1.1. The server now refuses to accept encrypted connections with clients using anything except TLS 1.2. The verbose wording in the directive communicates more clearly, to a human reader, what the server is configured to do.

Note

Disabling TLS versions 1.0 and 1.1 in this manner blocks a small percentage of outdated web browsers from accessing your site.

To modify the list of allowed ciphers

1. Open the configuration file `/etc/httpd/conf.d/ssl.conf` and find the section with commented-out examples for configuring `SSLCipherSuite` and `SSLProxyCipherSuite`.

```
#SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5
#SSLProxyCipherSuite HIGH:MEDIUM:!aNULL:!MD5
```

Leave these as they are, and below them add the following directives:

Note

Though shown here on several lines for readability, each of these two directives must be on a single line without spaces between the cipher names.

```
SSLCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-
CHACHA20-POLY1305:
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256:ECDHE-ECDSA-AES256-SHA384:
ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:AES:!aNULL:!
eNULL:!EXPORT:!DES:
!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA

SSLProxyCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-
ECDSA-CHACHA20-POLY1305:
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256:ECDHE-ECDSA-AES256-SHA384:
ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:AES:!aNULL:!
eNULL:!EXPORT:!DES:
!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA
```

These ciphers are a subset of the much longer list of supported ciphers in OpenSSL. They were selected and ordered according to the following criteria:

- Support for forward secrecy
- Strength
- Speed
- Specific ciphers before cipher families
- Allowed ciphers before denied ciphers

Note that the high-ranking ciphers have *ECDHE* in their names, for *Elliptic Curve Diffie-Hellman Ephemeral*; the *ephemeral* indicates forward secrecy. Also, RC4 is now among the forbidden ciphers near the end.

We recommend that you use an explicit list of ciphers instead relying on defaults or terse directives whose content isn't visible.

Important

The cipher list shown here is just one of many possible lists; for instance, you might want to optimize a list for speed rather than forward secrecy.

If you anticipate a need to support older clients, you can allow the DES-CBC3-SHA cipher suite.

Finally, each update to OpenSSL introduces new ciphers and deprecates old ones. Keep your EC2 Amazon Linux instance up to date, watch for security announcements from [OpenSSL](#), and be alert to reports of new security exploits in the technical press. For more information, see [Predefined SSL Security Policies for Elastic Load Balancing](#) in the *User Guide for Classic Load Balancers*.

2. Uncomment the following line by removing the "#":

```
#SSLCipherOrder on
```

This command forces the server to prefer high-ranking ciphers, including (in this case) those that support forward secrecy. With this directive turned on, the server tries to establish a strongly secure connection before falling back to allowed ciphers with lesser security.

3. Restart Apache. If you test the domain again on [Qualys SSL Labs](#), you should see that the RC4 vulnerability is gone.

Troubleshooting

- **My Apache webserver won't start unless I supply a password**

This is expected behavior if you installed an encrypted, password-protected, private server key.

You can remove the encryption and password requirement from the key. Assuming that you have a private encrypted RSA key called `custom.key` in the default directory, and that the password on it is `abcde12345`, run the following commands on your EC2 instance to generate an unencrypted version of the key.

```
[ec2-user ~]$ cd /etc/pki/tls/private/
[ec2-user private]$ sudo cp custom.key custom.key.bak
[ec2-user private]$ sudo openssl rsa -in custom.key -passin pass:abcde12345 -out
  custom.key.nocrypt
[ec2-user private]$ sudo mv custom.key.nocrypt custom.key
[ec2-user private]$ sudo chown root.root custom.key
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ sudo service httpd restart
```

Apache should now start without prompting you for a password.

Certificate automation: Let's Encrypt with Certbot on Amazon Linux

The [Let's Encrypt](#) certificate authority is the centerpiece of the Electronic Frontier Foundation (EFF) effort to encrypt the entire internet. In line with that goal, Let's Encrypt host certificates are designed to be created, validated, installed, and maintained with minimal human intervention. The automated aspects of certificate management are carried out by an agent running on the web server. After you install and

configure the agent, it communicates securely with Let's Encrypt and performs administrative tasks on Apache and the key management system. This tutorial uses the free [Certbot](#) agent because it allows you either to supply a customized encryption key as the basis for your certificates, or to allow the agent itself to create a key based on its defaults. You can also configure Certbot to renew your certificates on a regular basis without human interaction, as described in [To automate Certbot \(p. 79\)](#). For more information, consult the Certbot [User Guide](#) or [man page](#).

Certbot is not officially supported on Amazon Linux AMI, but is available for download and functions correctly when installed. We recommend that you make the following backups to protect your data and avoid inconvenience:

- Before you begin, take a snapshot of your Amazon EBS root volume. This allows you to restore the original state of your EC2 instance. For information about creating EBS snapshots, see [Creating Amazon EBS snapshots \(p. 1063\)](#).
- The procedure below requires you to edit your `httpd.conf` file, which controls Apache's operation. Certbot makes its own automated changes to this and other configuration files. Make a backup copy of your entire `/etc/httpd` directory in case you need to restore it.

To install and run Certbot

1. Enable the Extra Packages for Enterprise Linux (EPEL) repository from the Fedora project on your instance. Packages from EPEL are required as dependencies when you run the Certbot installation script.

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

2. Download the latest release of Certbot from EFF onto your EC2 instance using the following command.

```
[ec2-user ~]$ wget https://dl.eff.org/certbot-auto
```

3. Make the downloaded file executable.

```
[ec2-user ~]$ chmod a+x certbot-auto
```

4. Run the file with root permissions and the `--debug` flag.

```
[ec2-user ~]$ sudo ./certbot-auto --debug
```

5. At the prompt "Is this ok [y/d/N]," type "y" and press Enter.
6. At the prompt "Enter email address (used for urgent renewal and security notices)," type a contact address and press Enter.
7. Agree to the Let's Encrypt Terms of Service at the prompt. Type "A" and press Enter to proceed:

```
-----  
Please read the Terms of Service at  
https://letsencrypt.org/documents/LE-SA-v1.1.1-August-1-2016.pdf. You must agree  
in order to register with the ACME server at  
https://acme-v01.api.letsencrypt.org/directory
```

```
(A)gree/(C)ancel: A
```

8. Click through the authorization for EFF put you on their mailing list by typing "Y" or "N" and press Enter.

Amazon Elastic Compute Cloud
User Guide for Linux Instances
Certificate automation: Let's Encrypt
with Certbot on Amazon Linux

-
9. At the prompt shown below, type your Common Name (the name of your domain as described above) and your Subject Alternative Name (SAN), separating the two names with a space or a comma. Then press Enter. In this example, the names have been provided:

```
No names were found in your configuration files. Please enter in your domain
name(s) (comma and/or space separated) (Enter 'c' to cancel):example.com
www.example.com
```

10. On an Amazon Linux system with a default Apache configuration, you see output similar to the example below, asking about the first name you provided. Type "1" and press Enter.

```
Obtaining a new certificate
Performing the following challenges:
tls-sni-01 challenge for example.com
tls-sni-01 challenge for www.example.com

We were unable to find a vhost with a ServerName or Address of example.com.
Which virtual host would you like to choose?
(note: conf files with multiple vhosts are not yet supported)
-----
1: ssl.conf | | HTTPS | Enabled
-----
Press 1 [enter] to confirm the selection (press 'c' to cancel): 1
```

11. Next, Certbot asks about the second name. Type "1" and press Enter.

```
We were unable to find a vhost with a ServerName or Address of www.example.com.
Which virtual host would you like to choose?
(note: conf files with multiple vhosts are not yet supported)
-----
1: ssl.conf | | HTTPS | Enabled
-----
Press 1 [enter] to confirm the selection (press 'c' to cancel): 1
```

At this point, Certbot creates your key and a CSR:

```
Waiting for verification...
Cleaning up challenges
Generating key (2048 bits): /etc/letsencrypt/keys/0000_key-certbot.pem
Creating CSR: /etc/letsencrypt/csr/0000_csr-certbot.pem
```

12. Authorize Certbot to create and all needed host certificates. When prompted for each name, type "1" and press Enter as shown in the example:

```
We were unable to find a vhost with a ServerName or Address of example.com.
Which virtual host would you like to choose?
(note: conf files with multiple vhosts are not yet supported)
-----
1: ssl.conf | | HTTPS | Enabled
-----
Press 1 [enter] to confirm the selection (press 'c' to cancel): 1
Deploying Certificate for example.com to VirtualHost /etc/httpd/conf.d/ssl.conf

We were unable to find a vhost with a ServerName or Address of www.example.com.
Which virtual host would you like to choose?
(note: conf files with multiple vhosts are not yet supported)
-----
1: ssl.conf | example.com | HTTPS | Enabled
-----
Press 1 [enter] to confirm the selection (press 'c' to cancel): 1
```

Deploying Certificate for www.example.com to VirtualHost /etc/httpd/conf.d/ssl.conf

13. Choose whether to allow insecure connections to your web server. If you choose option 2 (as shown in the example), all connections to your server will either be encrypted or rejected.

```
Please choose whether HTTPS access is required or optional.  
-----  
1: Easy - Allow both HTTP and HTTPS access to these sites  
2: Secure - Make all requests redirect to secure HTTPS access  
-----  
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2
```

Certbot completes the configuration of Apache and reports success and other information:

```
Congratulations! You have successfully enabled https://example.com and  
https://www.example.com
```

```
You should test your configuration at:  
https://www.ssllabs.com/ssltest/analyze.html?d=example.com  
https://www.ssllabs.com/ssltest/analyze.html?d=www.example.com
```

IMPORTANT NOTES:

```
- Congratulations! Your certificate and chain have been saved at  
/etc/letsencrypt/live/example.com/fullchain.pem. Your cert will  
expire on 2017-07-19. To obtain a new or tweaked version of this  
certificate in the future, simply run certbot-auto again with the  
"certonly" option. To non-interactively renew *all* of your  
certificates, run "certbot-auto renew"
```

14. After you complete the installation, test and optimize the security of your server as described in [Step 3: Test and harden the security configuration \(p. 73\)](#).

Certbot is designed to become an invisible, error-resistant part of your server system. By default, it generates host certificates with a short, 90-day expiration time. If you have not previously configured your system to call the command automatically, you must re-run the **certbot** command manually. This procedure shows how to automate Certbot by setting up a cron job.

To configure automated certificate renewal

1. Open the `/etc/crontab` file in a text editor, such as **vim** or **nano**, using **sudo**. Alternatively, use **sudo crontab -e**.
2. Add a line similar to the following and save the file.

```
39      1,13    *      *      *      root    certbot renew --no-self-upgrade
```

Here is an explanation of each component:

```
39 1,13 * * *
```

Schedules a command to be run at 01:39 and 13:39 every day. The selected values are arbitrary, but the Certbot developers suggest running the command at least twice daily. This guarantees that any certificate found to be compromised is promptly revoked and replaced.

```
root
```

The command runs with root privileges.

```
certbot renew --no-self-upgrade
```

The command to be run. The **renew** subcommand causes Certbot to check any previously obtained certificates and to renew those that are approaching expiration. The **--no-self-upgrade** flag prevents Certbot from upgrading itself without your intervention.

3. Restart the cron daemon:

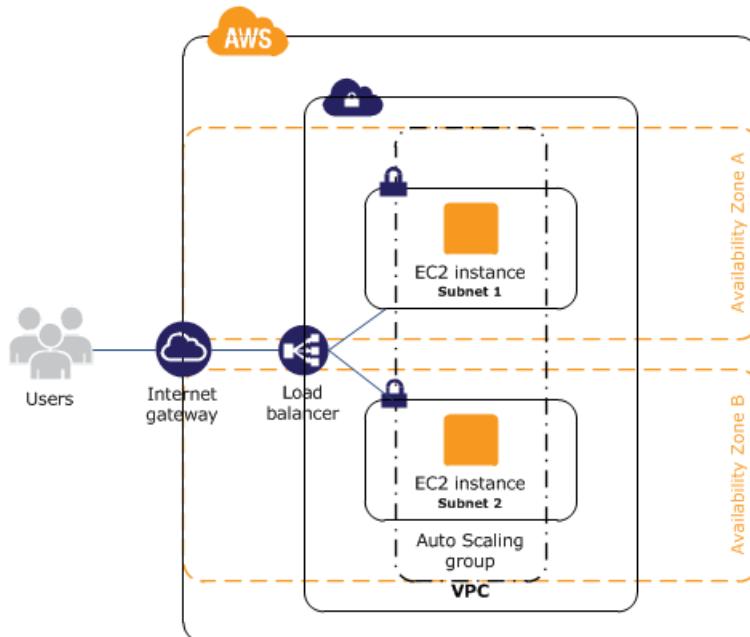
```
[ec2-user ~]$ sudo service crond restart
```

Tutorial: Increase the availability of your application on Amazon EC2

Suppose that you start out running your app or website on a single EC2 instance, and over time, traffic increases to the point that you require more than one instance to meet the demand. You can launch multiple EC2 instances from your AMI and then use Elastic Load Balancing to distribute incoming traffic for your application across these EC2 instances. This increases the availability of your application. Placing your instances in multiple Availability Zones also improves the fault tolerance in your application. If one Availability Zone experiences an outage, traffic is routed to the other Availability Zone.

You can use Amazon EC2 Auto Scaling to maintain a minimum number of running instances for your application at all times. Amazon EC2 Auto Scaling can detect when your instance or application is unhealthy and replace it automatically to maintain the availability of your application. You can also use Amazon EC2 Auto Scaling to scale your Amazon EC2 capacity up or down automatically based on demand, using criteria that you specify.

In this tutorial, we use Amazon EC2 Auto Scaling with Elastic Load Balancing to ensure that you maintain a specified number of healthy EC2 instances behind your load balancer. Note that these instances do not need public IP addresses, because traffic goes to the load balancer and is then routed to the instances. For more information, see [Amazon EC2 Auto Scaling](#) and [Elastic Load Balancing](#).



Contents

- [Prerequisites \(p. 94\)](#)
- [Scale and load balance your application \(p. 94\)](#)
- [Test your load balancer \(p. 96\)](#)

Prerequisites

This tutorial assumes that you have already done the following:

1. Create a virtual private cloud (VPC) with one public subnet in two or more Availability Zones.
2. Launch an instance in the VPC.
3. Connect to the instance and customized it. For example, installing software and applications, copying data, and attaching additional EBS volumes. For information about setting up a web server on your instance, see [Tutorial: Install a LAMP web server with the Amazon Linux AMI \(p. 46\)](#).
4. Test your application on your instance to ensure that your instance is configured correctly.
5. Create a custom Amazon Machine Image (AMI) from your instance. For more information, see [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#) or [Creating an instance store-backed Linux AMI \(p. 127\)](#).
6. (Optional) Terminate the instance if you no longer need it.
7. Create an IAM role that grants your application the access to AWS it needs. For more information, see [To create an IAM role using the IAM console \(p. 976\)](#).

Scale and load balance your application

Use the following procedure to create a load balancer, create a launch configuration for your instances, create an Auto Scaling group with two or more instances, and associate the load balancer with the Auto Scaling group.

To scale and load-balance your application

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, under **LOAD BALANCING**, choose **Load Balancers**.
3. Choose **Create Load Balancer**.
4. For **Application Load Balancer**, choose **Create**.
5. On the **Configure Load Balancer** page, do the following:
 - a. For **Name**, enter a name for your load balancer. For example, **my-1b**.
 - b. For **Scheme**, keep the default value, **internet-facing**.
 - c. For **Listeners**, keep the default, which is a listener that accepts HTTP traffic on port 80.
 - d. For **Availability Zones**, select the VPC that you used for your instances. Select an Availability Zone and then select the public subnet for that Availability Zone. Repeat for a second Availability Zone.
 - e. Choose **Next: Configure Security Settings**.
6. For this tutorial, you are not using a secure listener. Choose **Next: Configure Security Groups**.
7. On the **Configure Security Groups** page, do the following:
 - a. Choose **Create a new security group**.
 - b. Type a name and description for the security group, or keep the default name and description. This new security group contains a rule that allows traffic to the port configured for the listener.
 - c. Choose **Next: Configure Routing**.

8. On the **Configure Routing** page, do the following:
 - a. For **Target group**, keep the default, **New target group**.
 - b. For **Name**, enter a name for the target group.
 - c. Keep **Protocol** as HTTP, **Port** as 80, and **Target type** as instance.
 - d. For **Health checks**, keep the default protocol and path.
 - e. Choose **Next: Register Targets**.
9. On the **Register Targets** page, choose **Next: Review** to continue to the next page, as we'll use Amazon EC2 Auto Scaling to add EC2 instances to the target group.
10. On the **Review** page, choose **Create**. After the load balancer is created, choose **Close**.
11. On the navigation pane, under **AUTO SCALING**, choose **Launch Configurations**.
 - If you are new to Amazon EC2 Auto Scaling, you see a welcome page. Choose **Create Auto Scaling group** to start the Create Auto Scaling Group wizard, and then choose **Create launch configuration**.
 - Otherwise, choose **Create launch configuration**.
12. For **Launch configuration name**, enter a name for your launch configuration (for example, **my-launch-config**).
13. For **Amazon machine image (AMI)**, under **My AMIs**, choose the AMI that you created in [Prerequisites \(p. 94\)](#).
14. For **Instance type**, use **Choose instance type** to select an instance type.
15. (Optional) For **Additional configuration**, do the following as needed:
 - a. Choose **Request Spot Instances**. Otherwise, the instances are On-Demand instances.
 - b. For **IAM instance profile**, select the IAM role that you created in [Prerequisites \(p. 94\)](#).
 - c. Choose **Enable EC2 instance detailed monitoring within CloudWatch**. Otherwise, basic monitoring is enabled.
 - d. To configure instance metadata, expand **Advanced details**, enable or disable instance metadata, and configure the version and hop limit as needed.
 - e. To run a startup script, expand **Advanced details** and enter the script in **User data**.
 - f. To assign public IP addresses to your instances, expand **Advanced details** and set **IP address type** as needed.
16. For **Storage (volumes)**, you add volumes as needed. You can create empty EBS volumes or create EBS volumes from EBS snapshots.
17. For **Security groups**, you can select an existing security group or create a new one. This security group must allow HTTP traffic and health checks from the load balancer. If you assigned public IP addresses to your instances, you can optionally allow SSH traffic so you can connect to them.
18. For **Key pair (login)**, choose an existing key pair, create a new key pair, or proceed without a key pair. Select the acknowledgment check box.
19. Choose **Create launch configuration**.
20. After the launch configuration is created, you must create an Auto Scaling group.
 - If you are new to Amazon EC2 Auto Scaling and you are using the Create Auto Scaling group wizard, you are taken to the next step automatically.
 - Otherwise, select the Auto Scaling group and choose **Actions, Create an Auto Scaling group**.
21. On the **Choose launch template or configuration** page, enter a name for the Auto Scaling group. For example, **my-asg**. Choose **Next**.
22. On the **Configure settings** page, choose your VPC and your two public subnets. Choose **Next**.
23. On the **Configure advanced options** page, select **Enable load balancing** and choose your target group. Select the **ELB health check type** and choose **Next**.
24. For **Group size**, type the number of instances (for example, **2**). Note that we recommend that you maintain approximately the same number of instances in each Availability Zone.

25. On the **Configure group size and scaling policies** page, you can configure the size of the group or configure the group to scale dynamically based on demand. Choose **Next**.
26. (Optional) Choose **Add notifications** to configure SNS notifications for scaling activities. Choose **Next**.
27. (Optional) Choose **Add tag** to add tags. Choose **Next**.
28. On the **Review** page, edit the details as needed, and then choose **Create Auto Scaling group**.

Test your load balancer

When a client sends a request to your load balancer, the load balancer routes the request to one of its registered instances.

To test your load balancer

1. Verify that your instances are ready. From the **Auto Scaling Groups** page, select your Auto Scaling group, and then choose the **Instance management** tab. Initially, your instances are in the **Pending** state. When **Lifecycle** is **InService**, your instances are ready for use.
2. Verify that your instances are registered with the load balancer. From the **Target Groups** page, choose the name of the target group to open its details page, and then choose **Targets**. If the state of your instances is **initial**, it's possible that they are still registering. When the state of your instances is **healthy**, they are ready for use. After your instances are ready, you can go to the next step.
3. From the **Load Balancers** page, select your load balancer.
4. On the **Description** tab, locate the DNS name. This name has the following form:

`my-lb-xxxxxxxxxx.us-west-2.elb.amazonaws.com`

5. In a web browser, paste the DNS name for the load balancer into the address bar and press Enter. You'll see your website displayed.

Amazon Machine Images (AMI)

An Amazon Machine Image (AMI) provides the information required to launch an instance. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration. You can use different AMIs to launch instances when you need instances with different configurations.

An AMI includes the following:

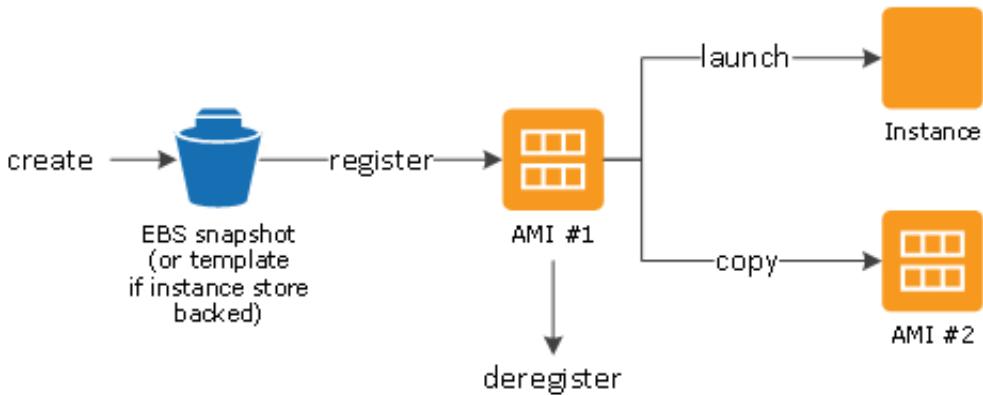
- One or more EBS snapshots, or, for instance-store-backed AMIs, a template for the root volume of the instance (for example, an operating system, an application server, and applications).
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it's launched.

Topics

- [Using an AMI \(p. 97\)](#)
- [Creating your own AMI \(p. 98\)](#)
- [Buying, sharing, and selling AMIs \(p. 98\)](#)
- [Deregistering your AMI \(p. 99\)](#)
- [Amazon Linux 2 and Amazon Linux AMI \(p. 99\)](#)
- [AMI types \(p. 99\)](#)
- [Linux AMI virtualization types \(p. 102\)](#)
- [Finding a Linux AMI \(p. 104\)](#)
- [Shared AMIs \(p. 109\)](#)
- [Paid AMIs \(p. 119\)](#)
- [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#)
- [Creating an instance store-backed Linux AMI \(p. 127\)](#)
- [Using encryption with EBS-backed AMIs \(p. 157\)](#)
- [Copying an AMI \(p. 163\)](#)
- [Obtaining billing information \(p. 169\)](#)
- [Deregistering your Linux AMI \(p. 172\)](#)
- [Amazon Linux \(p. 175\)](#)
- [User provided kernels \(p. 193\)](#)
- [Using the MATE desktop environment provided with Amazon Linux 2 \(p. 198\)](#)

Using an AMI

The following diagram summarizes the AMI lifecycle. After you create and register an AMI, you can use it to launch new instances. (You can also launch instances from an AMI if the AMI owner grants you launch permissions.) You can copy an AMI within the same Region or to different Regions. When you no longer require an AMI, you can deregister it.



You can search for an AMI that meets the criteria for your instance. You can search for AMIs provided by AWS or AMIs provided by the community. For more information, see [AMI types \(p. 99\)](#) and [Finding an AMI](#).

After you launch an instance from an AMI, you can connect to it. When you are connected to an instance, you can use it just like you use any other server. For information about launching, connecting, and using your instance, see [Amazon EC2 instances \(p. 200\)](#).

Creating your own AMI

You can launch an instance from an existing AMI, customize the instance, and then save this updated configuration as a custom AMI. Instances launched from this new custom AMI include the customizations that you made when you created the AMI.

The root storage device of the instance determines the process you follow to create an AMI. The root volume of an instance is either an Amazon EBS volume or an instance store volume. For more information about the root device volume, see [Amazon EC2 root device volume \(p. 20\)](#).

- To create an Amazon EBS-backed AMI, see [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#).
- To create an instance store-backed AMI, see [Creating an instance store-backed Linux AMI \(p. 127\)](#).

To help categorize and manage your AMIs, you can assign custom *tags* to them. For more information, see [Tagging your Amazon EC2 resources \(p. 1229\)](#).

Buying, sharing, and selling AMIs

After you create an AMI, you can keep it private so that only you can use it, or you can share it with a specified list of AWS accounts. You can also make your custom AMI public so that the community can use it. Building a safe, secure, usable AMI for public consumption is a fairly straightforward process, if you follow a few simple guidelines. For information about how to create and use shared AMIs, see [Shared AMIs \(p. 109\)](#).

You can purchase AMIs from a third party, including AMIs that come with service contracts from organizations such as Red Hat. You can also create an AMI and sell it to other Amazon EC2 users. For more information about buying or selling AMIs, see [Paid AMIs \(p. 119\)](#).

Deregistering your AMI

You can deregister an AMI when you have finished with it. After you deregister an AMI, it can't be used to launch new instances. Existing instances launched from the AMI are not affected. For more information, see [Deregistering your Linux AMI \(p. 172\)](#).

Amazon Linux 2 and Amazon Linux AMI

Amazon Linux 2 and the Amazon Linux AMI are supported and maintained Linux images provided by AWS. The following are some of the features of Amazon Linux 2 and Amazon Linux AMI:

- A stable, secure, and high-performance execution environment for applications running on Amazon EC2.
- Provided at no additional charge to Amazon EC2 users.
- Repository access to multiple versions of MySQL, PostgreSQL, Python, Ruby, Tomcat, and many more common packages.
- Updated on a regular basis to include the latest components, and these updates are also made available in the **yum** repositories for installation on running instances.
- Includes packages that enable easy integration with AWS services, such as the AWS CLI, Amazon EC2 API and AMI tools, the Boto library for Python, and the Elastic Load Balancing tools.

For more information, see [Amazon Linux \(p. 175\)](#).

AMI types

You can select an AMI to use based on the following characteristics:

- Region (see [Regions and Zones \(p. 7\)](#))
- Operating system
- Architecture (32-bit or 64-bit)
- [Launch permissions \(p. 99\)](#)
- [Storage for the root device \(p. 100\)](#)

Launch permissions

The owner of an AMI determines its availability by specifying launch permissions. Launch permissions fall into the following categories.

Launch permission	Description
public	The owner grants launch permissions to all AWS accounts.
explicit	The owner grants launch permissions to specific AWS accounts.
implicit	The owner has implicit launch permissions for an AMI.

Amazon and the Amazon EC2 community provide a large selection of public AMIs. For more information, see [Shared AMIs \(p. 109\)](#). Developers can charge for their AMIs. For more information, see [Paid AMIs \(p. 119\)](#).

Storage for the root device

All AMIs are categorized as either *backed by Amazon EBS* or *backed by instance store*. The former means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot. The latter means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3. For more information, see [Amazon EC2 root device volume \(p. 20\)](#).

The following table summarizes the important differences when using the two types of AMIs.

Characteristic	Amazon EBS-backed AMI	Amazon instance store-backed AMI
Boot time for an instance	Usually less than 1 minute	Usually less than 5 minutes
Size limit for a root device	16 TiB	10 GiB
Root device volume	Amazon EBS volume	Instance store volume
Data persistence	By default, the root volume is deleted when the instance terminates.* Data on any other Amazon EBS volumes persists after instance termination by default.	Data on any instance store volumes persists only during the life of the instance.
Modifications	The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped.	Instance attributes are fixed for the life of an instance.
Charges	You're charged for instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot.	You're charged for instance usage and storing your AMI in Amazon S3.
AMI creation/bundling	Uses a single command/call	Requires installation and use of AMI tools
Stopped state	Can be in a stopped state. Even when the instance is stopped and not running, the root volume is persisted in Amazon EBS	Cannot be in stopped state; instances are running or terminated

* By default, Amazon EBS-backed instance root volumes have the `DeleteOnTermination` flag set to `true`. For information about how to change this flag so that the volume persists after termination, see [Changing the root volume to persist \(p. 23\)](#).

Determining the root device type of your AMI

To determine the root device type of an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, click **AMIs**, and select the AMI.

3. Check the value of **Root Device Type** in the **Details** tab as follows:

- If the value is `ebs`, this is an Amazon EBS-backed AMI.
- If the value is `instance store`, this is an instance store-backed AMI.

To determine the root device type of an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `describe-images` (AWS CLI)
- `Get-EC2Image` (AWS Tools for Windows PowerShell)

Stopped state

You can stop an Amazon EBS-backed instance, but not an Amazon EC2 instance store-backed instance. Stopping causes the instance to stop running (its status goes from `running` to `stopping` to `stopped`). A stopped instance persists in Amazon EBS, which allows it to be restarted. Stopping is different from terminating; you can't restart a terminated instance. Because Amazon EC2 instance store-backed instances can't be stopped, they're either running or terminated. For more information about what happens and what you can do while an instance is stopped, see [Stop and start your instance \(p. 587\)](#).

Default data storage and persistence

Instances that use an instance store volume for the root device automatically have instance store available (the root volume contains the root partition and you can store additional data). You can add persistent storage to your instance by attaching one or more Amazon EBS volumes. Any data on an instance store volume is deleted when the instance fails or terminates. For more information, see [Instance store lifetime \(p. 1189\)](#).

Instances that use Amazon EBS for the root device automatically have an Amazon EBS volume attached. The volume appears in your list of volumes like any other. With most instance types, Amazon EBS-backed instances don't have instance store volumes by default. You can add instance store volumes or additional Amazon EBS volumes using a block device mapping. For more information, see [Block device mapping \(p. 1211\)](#).

Boot times

Instances launched from an Amazon EBS-backed AMI launch faster than instances launched from an instance store-backed AMI. When you launch an instance from an instance store-backed AMI, all the parts have to be retrieved from Amazon S3 before the instance is available. With an Amazon EBS-backed AMI, only the parts required to boot the instance need to be retrieved from the snapshot before the instance is available. However, the performance of an instance that uses an Amazon EBS volume for its root device is slower for a short time while the remaining parts are retrieved from the snapshot and loaded into the volume. When you stop and restart the instance, it launches quickly, because the state is stored in an Amazon EBS volume.

AMI creation

To create Linux AMIs backed by instance store, you must create an AMI from your instance on the instance itself using the Amazon EC2 AMI tools.

AMI creation is much easier for AMIs backed by Amazon EBS. The `CreateImage` API action creates your Amazon EBS-backed AMI and registers it. There's also a button in the AWS Management Console that lets you create an AMI from a running instance. For more information, see [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#).

How you're charged

With AMIs backed by instance store, you're charged for instance usage and storing your AMI in Amazon S3. With AMIs backed by Amazon EBS, you're charged for instance usage, Amazon EBS volume storage and usage, and storing your AMI as an Amazon EBS snapshot.

With Amazon EC2 instance store-backed AMIs, each time you customize an AMI and create a new one, all of the parts are stored in Amazon S3 for each AMI. So, the storage footprint for each customized AMI is the full size of the AMI. For Amazon EBS-backed AMIs, each time you customize an AMI and create a new one, only the changes are stored. So the storage footprint for subsequent AMIs you customize after the first is much smaller, resulting in lower AMI storage charges.

When an Amazon EBS-backed instance is stopped, you're not charged for instance usage; however, you're still charged for volume storage. As soon as you start your instance, we charge a minimum of one minute for usage. After one minute, we charge only for the seconds used. For example, if you run an instance for 20 seconds and then stop it, we charge for a full one minute. If you run an instance for 3 minutes and 40 seconds, we charge for exactly 3 minutes and 40 seconds of usage. We charge you for each second, with a one-minute minimum, that you keep the instance running, even if the instance remains idle and you don't connect to it.

Linux AMI virtualization types

Linux Amazon Machine Images use one of two types of virtualization: paravirtual (PV) or hardware virtual machine (HVM). The main differences between PV and HVM AMIs are the way in which they boot and whether they can take advantage of special hardware extensions (CPU, network, and storage) for better performance.

For the best performance, we recommend that you use current generation instance types and HVM AMIs when you launch your instances. For more information about current generation instance types, see [Amazon EC2 Instance Types](#). If you are using previous generation instance types and would like to upgrade, see [Upgrade Paths](#).

The following table compares HVM and PV AMIs.

	HVM	PV
Description	HVM AMIs are presented with a fully virtualized set of hardware and boot by executing the master boot record of the root block device of your image. This virtualization type provides the ability to run an operating system directly on top of a virtual machine without any modification, as if it were run on the bare-metal hardware. The Amazon EC2 host system emulates some or all of the underlying hardware that is presented to the guest.	PV AMIs boot with a special boot loader called PV-GRUB, which starts the boot cycle and then chain loads the kernel specified in the <code>menu.1st</code> file on your image. Paravirtual guests can run on host hardware that does not have explicit support for virtualization. Historically, PV guests had better performance than HVM guests in many cases, but because of enhancements in HVM virtualization and the availability of PV drivers for HVM AMIs, this is no longer true. For more information about PV-GRUB and its use in Amazon EC2, see Enabling Your Own Linux Kernels (p. 193) .

	HVM	PV
Support for hardware extensions	Yes. Unlike PV guests, HVM guests can take advantage of hardware extensions that provide fast access to the underlying hardware on the host system. For more information on CPU virtualization extensions available in Amazon EC2, see Intel Virtualization Technology on the Intel website. HVM AMIs are required to take advantage of enhanced networking and GPU processing. In order to pass through instructions to specialized network and GPU devices, the OS needs to be able to have access to the native hardware platform; HVM virtualization provides this access. For more information, see Enhanced networking on Linux (p. 813) and Linux accelerated computing instances (p. 278) .	No, they cannot take advantage of special hardware extensions such as enhanced networking or GPU processing.
Supported instance types	All current generation instance types support HVM AMIs.	The following previous generation instance types support PV AMIs: C1, C3, HS1, M1, M3, M2, and T1. Current generation instance types do not support PV AMIs.
Supported Regions	All Regions support HVM instances.	Asia Pacific (Tokyo), Asia Pacific (Singapore), Asia Pacific (Sydney), Europe (Frankfurt), Europe (Ireland), South America (São Paulo), US East (N. Virginia), US West (N. California), and US West (Oregon)
How to find	Verify that the virtualization type of the AMI is set to <code>hvm</code> , using the console or the describe-images command.	Verify that the virtualization type of the AMI is set to <code>paravirtual</code> , using the console or the describe-images command.

PV on HVM

Paravirtual guests traditionally performed better with storage and network operations than HVM guests because they could leverage special drivers for I/O that avoided the overhead of emulating network and disk hardware, whereas HVM guests had to translate these instructions to emulated hardware. Now PV drivers are available for HVM guests, so operating systems that cannot be ported to run in a paravirtualized environment can still see performance advantages in storage and network I/O by using them. With these PV on HVM drivers, HVM guests can get the same, or better, performance than paravirtual guests.

Finding a Linux AMI

Before you can launch an instance, you must select an AMI to use. As you select an AMI, consider the following requirements you might have for the instances that you'll launch:

- The Region
- The operating system
- The architecture: 32-bit (`i386`), 64-bit (`x86_64`), or 64-bit ARM (`arm64`)
- The root device type: Amazon EBS or instance store
- The provider (for example, Amazon Web Services)
- Additional software (for example, SQL server)

If you need to find a Windows AMI, see [Finding a Windows AMI in the Amazon EC2 User Guide for Windows Instances](#).

Contents

- [Finding a Linux AMI using the Amazon EC2 console \(p. 104\)](#)
- [Finding an AMI using the AWS CLI \(p. 105\)](#)
- [Finding the latest Amazon Linux AMI using Systems Manager \(p. 105\)](#)
- [Using a Systems Manager parameter to find an AMI \(p. 106\)](#)
- [Finding a Quick Start AMI \(p. 108\)](#)

Finding a Linux AMI using the Amazon EC2 console

You can find Linux AMIs using the Amazon EC2 console. You can select from the list of AMIs when you use the launch wizard to launch an instance, or you can search through all available AMIs using the [Images](#) page. AMI IDs are unique to each AWS Region.

To find a Linux AMI using the launch wizard

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region in which to launch your instances. You can select any Region that's available to you, regardless of your location.
3. From the console dashboard, choose **Launch instance**.
4. On the **Quick Start** tab, select from one of the commonly used AMIs in the list. If you don't see the AMI that you need, select the **My AMIs**, **AWS Marketplace**, or **Community AMIs** tab to find additional AMIs. For more information, see [Step 1: Choose an Amazon Machine Image \(AMI\) \(p. 499\)](#).

To find a Linux AMI using the Images page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region in which to launch your instances. You can select any Region that's available to you, regardless of your location.
3. In the navigation pane, choose **AMIs**.
4. (Optional) Use the **Filter** options to scope the list of displayed AMIs to see only the AMIs that interest you. For example, to list all Linux AMIs provided by AWS, select **Public images**. Choose the Search bar and select **Owner** from the menu, then select **Amazon images**. Choose the Search bar again to select **Platform** and then the operating system from the list provided.

5. (Optional) Choose the **Show/Hide Columns** icon to select which image attributes to display, such as the root device type. Alternatively, you can select an AMI from the list and view its properties in the **Details** tab.
6. Before you select an AMI, it's important that you check whether it's backed by instance store or by Amazon EBS and that you are aware of the effects of this difference. For more information, see [Storage for the root device \(p. 100\)](#).
7. To launch an instance from this AMI, select it and then choose **Launch**. For more information about launching an instance using the console, see [Launching your instance from an AMI \(p. 500\)](#). If you're not ready to launch the instance now, make note of the AMI ID for later.

Finding an AMI using the AWS CLI

You can use AWS CLI commands for Amazon EC2 to list only the Linux AMIs that meet your needs. After locating an AMI that meets your needs, make note of its ID so that you can use it to launch instances. For more information, see [Launching an Instance Using the AWS CLI](#) in the *AWS Command Line Interface User Guide*.

The [describe-images](#) command supports filtering parameters. For example, use the `--owners` parameter to display public AMIs owned by Amazon.

```
aws ec2 describe-images --owners self amazon
```

You can add the following filter to the previous command to display only AMIs backed by Amazon EBS.

```
--filters "Name=root-device-type,Values=ebs"
```

Important

Omitting the `--owners` flag from the `describe-images` command will return all images for which you have launch permissions, regardless of ownership.

Finding the latest Amazon Linux AMI using Systems Manager

Amazon EC2 provides AWS Systems Manager public parameters for AWS-maintained public AMIs that you can use when launching instances. For example, the EC2-provided parameter `/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2` is available in all Regions and always points to the latest version of the Amazon Linux 2 AMI in a given Region.

The Amazon EC2 AMI public parameters are available from the following paths:

- `/aws/service/ami-amazon-linux-latest`
- `/aws/service/ami-windows-latest`

You can view a list of all Linux AMIs in the current AWS Region by using the following command in the AWS CLI.

```
aws ssm get-parameters-by-path --path /aws/service/ami-amazon-linux-latest --query Parameters[ ].Name
```

To launch an instance using a public parameter

The following example uses the EC2-provided public parameter to launch an `m5.xlarge` instance using the latest Amazon Linux 2 AMI.

To specify the parameter in the command, use the following syntax: `resolve:ssm:public-parameter`, where `resolve:ssm` is the standard prefix and `public-parameter` is the path and name of the public parameter.

In this example, the `--count` and `--security-group` parameters are not included. For `--count`, the default is 1. If you have a default VPC and a default security group, they are used.

```
aws ec2 run-instances
  --image-id resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2
  --instance-type m5.xlarge
  --key-name MyKeyPair
```

For more information, see [Using public parameters](#) in the *AWS Systems Manager User Guide* and [Query for the latest Amazon Linux AMI IDs Using AWS Systems Manager Parameter Store](#).

Using a Systems Manager parameter to find an AMI

When you launch an instance using the EC2 launch wizard in the console, you can either select an AMI from the list, or you can select an AWS Systems Manager parameter that points to an AMI ID. If you use automation code to launch your instances, you can specify the Systems Manager parameter instead of the AMI ID.

A Systems Manager parameter is a customer-defined key-value pair that you can create in Systems Manager Parameter Store. The Parameter Store provides a central store to externalize your application configuration values. For more information, see [AWS Systems Manager Parameter Store](#) in the *AWS Systems Manager User Guide*.

When you create a parameter that points to an AMI ID, make sure that you specify the data type as `aws:ec2:image`. This data type ensures that when the parameter is created or modified, the parameter value is validated as an AMI ID. For more information, see [Native parameter support for Amazon Machine Image IDs](#) in the *AWS Systems Manager User Guide*.

Contents

- [Use cases \(p. 106\)](#)
- [Launching an instance using a Systems Manager parameter \(p. 107\)](#)
- [Permissions \(p. 108\)](#)
- [Limitations \(p. 108\)](#)

Use cases

By using Systems Manager parameters to point to AMI IDs, you can make it easier for your users to select the correct AMI when launching instances, and you can simplify the maintenance of automation code.

Easier for users

If you require instances to be launched using a specific AMI, and if that AMI is updated regularly, we recommend that you require your users to select a Systems Manager parameter to find the AMI. By requiring your users to select a Systems Manager parameter, you can ensure that the latest AMI is used to launch instances.

For example, every month in your organization you might create a new version of your AMI that has the latest operating system and application patches. You also require your users to launch instances using the latest version of your AMI. To ensure that your users use the latest version, you can create a Systems Manager parameter (for example, `golden-ami`) that points to the correct AMI ID. Each time a new version of the AMI is created, you update the AMI ID value in the parameter so that it always points to the latest AMI. Your users don't need to know about the periodic updates to the AMI, because they

continue to select the same Systems Manager parameter every time. By having users select a Systems Manager parameter, you make it easier for them to select the correct AMI for an instance launch.

Simplify automation code maintenance

If you use automation code to launch your instances, you can specify the Systems Manager parameter instead of the AMI ID. If a new version of the AMI is created, you change the AMI ID value in the parameter so that it points to the latest AMI. The automation code that references the parameter doesn't need to be modified every time a new version of the AMI is created. This greatly simplifies maintenance of automation and helps drive down deployment costs.

Note

Running instances are not affected when you change the AMI ID to which the Systems Manager parameter points.

Launching an instance using a Systems Manager parameter

You can launch an instance using the console or the AWS CLI. Instead of specifying an AMI ID, you can specify an AWS Systems Manager parameter that points to an AMI ID.

To find a Linux AMI using a Systems Manager parameter (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region in which to launch your instances. You can select any Region that's available to you, regardless of your location.
3. From the console dashboard, choose **Launch instance**.
4. Choose **Search by Systems Manager parameter** (at top right).
5. For **Systems Manager parameter**, select a parameter. The corresponding AMI ID appears next to **Currently resolves to**.
6. Choose **Search**. The AMIs that match the AMI ID appear in the list.
7. Select the AMI from the list, and choose **Select**.

For more information about launching an instance from an AMI using the launch wizard, see [Step 1: Choose an Amazon Machine Image \(AMI\) \(p. 499\)](#).

To launch an instance using an AWS Systems Manager parameter instead of an AMI ID (AWS CLI)

The following example uses the Systems Manager parameter `golden-ami` to launch an `m5.xlarge` instance. The parameter points to an AMI ID.

To specify the parameter in the command, use the following syntax: `resolve:ssm:/parameter-name`, where `resolve:ssm` is the standard prefix and `parameter-name` is the unique parameter name. Note that the parameter name is case-sensitive. Backslashes for the parameter name are only necessary when the parameter is part of a hierarchy, for example, `/amis/production/golden-ami`. You can omit the backslash if the parameter is not part of a hierarchy.

In this example, the `--count` and `--security-group` parameters are not included. For `--count`, the default is 1. If you have a default VPC and a default security group, they are used.

```
aws ec2 run-instances
  --image-id resolve:ssm:/golden-ami
  --instance-type m5.xlarge
  ...
```

To launch an instance using a specific version of an AWS Systems Manager parameter (AWS CLI)

Systems Manager parameters have version support. Each iteration of a parameter is assigned a unique version number. You can reference the version of the parameter as follows `resolve:ssm:/parameter-`

`name:version`, where `version` is the unique version number. By default, the latest version of the parameter is used when no version is specified.

The following example uses version 2 of the parameter.

In this example, the `--count` and `--security-group` parameters are not included. For `--count`, the default is 1. If you have a default VPC and a default security group, they are used.

```
aws ec2 run-instances
  --image-id resolve:ssm:/golden-ami:2
  --instance-type m5.xlarge
  ...
```

To launch an instance using a public parameter provided by AWS

Amazon EC2 provides Systems Manager public parameters for public AMIs provided by AWS. For example, the public parameter `/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2` is available in all Regions and always points to the latest version of the Amazon Linux 2 AMI in the Region.

```
aws ec2 run-instances
  --image-id resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2
  --instance-type m5.xlarge
  ...
```

Permissions

If you use Systems Manager parameters that point to AMI IDs in the launch instance wizard, you must add `ssm:DescribeParameters` and `ssm:GetParameters` to your IAM policy. `ssm:DescribeParameters` grants your IAM users the permission to view and select Systems Manager parameters. `ssm:GetParameters` grants your IAM users the permission to get the values of the Systems Manager parameters. You can also restrict access to specific Systems Manager parameters. For more information, see [Using the EC2 launch wizard \(p. 967\)](#).

Limitations

AMIs and Systems Manager parameters are Region specific. To use the same Systems Manager parameter name across Regions, create a Systems Manager parameter in each Region with the same name (for example, `golden-ami`). In each Region, point the Systems Manager parameter to an AMI in that Region.

Finding a Quick Start AMI

When you launch an instance using the Amazon EC2 console, the **Choose an Amazon Machine Image (AMI)** page includes a list of popular AMIs on the **Quick Start** tab. If you want to automate launching an instance using one of these quick start AMIs, you'll need to programmatically locate the ID of the current version of the AMI.

To locate the current version of a Quick Start AMI, you can enumerate all AMIs with its AMI name, and then find the one with the most recent creation date.

Example Example: Find the current Amazon Linux 2 AMI

```
aws ec2 describe-images \
  --owners amazon \
  --filters 'Name=name,Values=amzn2-ami-hvm-2.0.?????????.?-x86_64-gp2'
  'Name=state,Values=available' \
  --query 'reverse(sort_by(Images, &CreationDate))[:1].ImageId' \
  --output text
```

Example Example: Find the current Amazon Linux AMI

```
aws ec2 describe-images \
--owners amazon \
--filters 'Name=name,Values=amzn-ami-hvm-?????.???.????????-x86_64-gp2' \
'Name=state,Values=available' \
--query 'reverse(sort_by(Images, &CreationDate))[:1].ImageId' \
--output text
```

Example Example: Find the current Ubuntu Server 16.04 LTS AMI

```
aws ec2 describe-images \
--owners 099720109477 \
--filters 'Name=name,Values=ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64- \
server-????????' 'Name=state,Values=available' \
--query 'reverse(sort_by(Images, &CreationDate))[:1].ImageId' \
--output text
```

Example Example: Find the current Red Hat Enterprise Linux 7.5 AMI

```
aws ec2 describe-images \
--owners 309956199498 \
--filters 'Name=name,Values=RHEL-7.5_HVM_GA*' 'Name=state,Values=available' \
--query 'reverse(sort_by(Images, &CreationDate))[:1].ImageId' \
--output text
```

Example Example: Find the current SUSE Linux Enterprise Server 15 AMI

```
aws ec2 describe-images \
--owners amazon \
--filters 'Name=name,Values=suse-sles-15-v????????-hvm-ssd-x86_64' \
'Name=state,Values=available' \
--query 'reverse(sort_by(Images, &CreationDate))[:1].ImageId' \
--output text
```

Shared AMIs

A *shared AMI* is an AMI that a developer created and made available for other developers to use. One of the easiest ways to get started with Amazon EC2 is to use a shared AMI that has the components you need and then add custom content. You can also create your own AMIs and share them with others.

You use a shared AMI at your own risk. Amazon can't vouch for the integrity or security of AMIs shared by other Amazon EC2 users. Therefore, you should treat shared AMIs as you would any foreign code that you might consider deploying in your own data center and perform the appropriate due diligence. We recommend that you get an AMI from a trusted source.

Amazon's public images have an aliased owner, which appears as `amazon` in the account field. This enables you to find AMIs from Amazon easily. Other users can't alias their AMIs.

For information about creating an AMI, see [Creating an Instance Store-Backed Linux AMI](#) or [Creating an Amazon EBS-Backed Linux AMI](#). For more information about building, delivering, and maintaining your applications on the AWS Marketplace, see the [AWS Marketplace Documentation](#).

Contents

- [Finding shared AMIs \(p. 110\)](#)

- [Making an AMI public \(p. 112\)](#)
- [Sharing an AMI with specific AWS accounts \(p. 113\)](#)
- [Using bookmarks \(p. 115\)](#)
- [Guidelines for shared Linux AMIs \(p. 115\)](#)

Finding shared AMIs

You can use the Amazon EC2 console or the command line to find shared AMIs.

AMIs are a regional resource. Therefore, when searching for a shared AMI (public or private), you must search for it from within the Region from which it is being shared. To make an AMI available in a different Region, copy the AMI to the Region and then share it. For more information, see [Copying an AMI](#).

Topics

- [Finding a shared AMI \(console\) \(p. 110\)](#)
- [Finding a shared AMI \(AWS CLI\) \(p. 110\)](#)
- [Using shared AMIs \(p. 111\)](#)

Finding a shared AMI (console)

To find a shared private AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. In the first filter, choose **Private images**. All AMIs that have been shared with you are listed. To granulate your search, choose the Search bar and use the filter options provided in the menu.

To find a shared public AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. In the first filter, choose **Public images**. To granulate your search, choose the Search bar and use the filter options provided in the menu.
4. Use filters to list only the types of AMIs that interest you. For example, choose **Owner :** and then choose **Amazon images** to display only Amazon's public images.

Finding a shared AMI (AWS CLI)

Use the [describe-images](#) command (AWS CLI) to list AMIs. You can scope the list to the types of AMIs that interest you, as shown in the following examples.

Example: List all public AMIs

The following command lists all public AMIs, including any public AMIs that you own.

```
aws ec2 describe-images --executable-users all
```

Example: List AMIs with explicit launch permissions

The following command lists the AMIs for which you have explicit launch permissions. This list does not include any AMIs that you own.

```
aws ec2 describe-images --executable-users self
```

Example: List AMIs owned by Amazon

The following command lists the AMIs owned by Amazon. Amazon's public AMIs have an aliased owner, which appears as `amazon` in the account field. This enables you to find AMIs from Amazon easily. Other users can't alias their AMIs.

```
aws ec2 describe-images --owners amazon
```

Example: List AMIs owned by an account

The following command lists the AMIs owned by the specified AWS account.

```
aws ec2 describe-images --owners 123456789012
```

Example: Scope AMIs using a filter

To reduce the number of displayed AMIs, use a filter to list only the types of AMIs that interest you. For example, use the following filter to display only EBS-backed AMIs.

```
--filters "Name=root-device-type,Values=ebs"
```

Using shared AMIs

Before you use a shared AMI, take the following steps to confirm that there are no pre-installed credentials that would allow unwanted access to your instance by a third party and no pre-configured remote logging that could transmit sensitive data to a third party. Check the documentation for the Linux distribution used by the AMI for information about improving the security of the system.

To ensure that you don't accidentally lose access to your instance, we recommend that you initiate two SSH sessions and keep the second session open until you've removed credentials that you don't recognize and confirmed that you can still log into your instance using SSH.

1. Identify and disable any unauthorized public SSH keys. The only key in the file should be the key you used to launch the AMI. The following command locates `authorized_keys` files:

```
[ec2-user ~]$ sudo find / -name "authorized_keys" -print -exec cat {} \;
```

2. Disable password-based authentication for the root user. Open the `sshd_config` file and edit the `PermitRootLogin` line as follows:

```
PermitRootLogin without-password
```

Alternatively, you can disable the ability to log into the instance as the root user:

```
PermitRootLogin No
```

Restart the `sshd` service.

3. Check whether there are any other user accounts that are able to log in to your instance. Accounts with superuser privileges are particularly dangerous. Remove or lock the password of any unknown accounts.
4. Check for open ports that you aren't using and running network services listening for incoming connections.

5. To prevent preconfigured remote logging, you should delete the existing configuration file and restart the rsyslog service. For example:

```
[ec2-user ~]$ sudo rm /etc/rsyslog.conf
[ec2-user ~]$ sudo service rsyslog restart
```

6. Verify that all cron jobs are legitimate.

If you discover a public AMI that you feel presents a security risk, contact the AWS security team. For more information, see the [AWS Security Center](#).

Making an AMI public

Amazon EC2 enables you to share your AMIs with other AWS accounts. You can allow all AWS accounts to launch the AMI (make the AMI public), or only allow a few specific accounts to launch the AMI (see [Sharing an AMI with specific AWS accounts \(p. 113\)](#)). You are not billed when your AMI is launched by other AWS accounts; only the accounts launching the AMI are billed.

AMIs with encrypted volumes cannot be made public.

AMIs are a regional resource. Therefore, sharing an AMI makes it available in that region. To make an AMI available in a different Region, copy the AMI to the Region and then share it. For more information, see [Copying an AMI \(p. 163\)](#).

To avoid exposing sensitive data when you share an AMI, read the security considerations in [Guidelines for shared Linux AMIs \(p. 115\)](#) and follow the recommended actions.

If an AMI has a product code, or contains a snapshot of an encrypted volume, you can't make it public. You can share the AMI only with specific AWS accounts.

Topics

- [Sharing an AMI with all AWS accounts \(console\) \(p. 112\)](#)
- [Sharing an AMI with all AWS accounts \(AWS CLI\) \(p. 112\)](#)

Sharing an AMI with all AWS accounts (console)

After you make an AMI public, it is available in **Community AMIs** when you launch an instance in the same Region using the console. Note that it can take a short while for an AMI to appear in **Community AMIs** after you make it public. It can also take a short while for an AMI to be removed from **Community AMIs** after you make it private again.

To share a public AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. Select your AMI from the list, and then choose **Actions, Modify Image Permissions**.
4. Choose **Public** and choose **Save**.

Sharing an AMI with all AWS accounts (AWS CLI)

Each AMI has a `launchPermission` property that controls which AWS accounts, besides the owner's, are allowed to use that AMI to launch instances. By modifying the `launchPermission` property of an AMI, you can make the AMI public (which grants launch permissions to all AWS accounts) or share it with only the AWS accounts that you specify.

You can add or remove account IDs from the list of accounts that have launch permissions for an AMI. To make the AMI public, specify the `all` group. You can specify both public and explicit launch permissions.

To make an AMI public

1. Use the [modify-image-attribute](#) command as follows to add the `all` group to the `launchPermission` list for the specified AMI.

```
aws ec2 modify-image-attribute \
--image-id ami-0abcdef1234567890 \
--launch-permission "Add=[{Group=all}]"
```

2. To verify the launch permissions of the AMI, use the [describe-image-attribute](#) command.

```
aws ec2 describe-image-attribute \
--image-id ami-0abcdef1234567890 \
--attribute launchPermission
```

3. (Optional) To make the AMI private again, remove the `all` group from its launch permissions. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
aws ec2 modify-image-attribute \
--image-id ami-0abcdef1234567890 \
--launch-permission "Remove=[{Group=all}]"
```

Sharing an AMI with specific AWS accounts

You can share an AMI with specific AWS accounts without making the AMI public. All you need is the AWS account IDs. You can only share AMIs that have unencrypted volumes and volumes that are encrypted with a customer managed CMK. If you share an AMI with encrypted volumes, you must also share any CMKs used to encrypt them. For more information, see [Sharing an Amazon EBS snapshot \(p. 1073\)](#). You cannot share an AMI that has volumes that are encrypted with a AWS managed CMK.

AMIs are a regional resource. Therefore, sharing an AMI makes it available in that Region. To make an AMI available in a different Region, copy the AMI to the Region and then share it. For more information, see [Copying an AMI \(p. 163\)](#).

There is no limit to the number of AWS accounts with which an AMI can be shared. User-defined tags that you attach to a shared AMI are available only to your AWS account and not to the other accounts that the AMI is shared with.

an AMI (console)

To grant explicit launch permissions using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. Select your AMI in the list, and then choose **Actions, Modify Image Permissions**.
4. Specify the AWS account number of the user with whom you want to share the AMI in the **AWS Account Number** field, then choose **Add Permission**.

To share this AMI with multiple users, repeat this step until you have added all the required users.

5. To allow create volume permissions for snapshots, select **Add "create volume" permissions to the following associated snapshots when creating permissions**.

Note

You do not need to share the Amazon EBS snapshots that an AMI references in order to share the AMI. Only the AMI itself needs to be shared; the system automatically provides the instance access to the referenced Amazon EBS snapshots for the launch. However, you do need to share any CMKs used to encrypt snapshots that the AMI references. For more information, see [Sharing an Amazon EBS snapshot \(p. 1073\)](#).

6. Choose **Save** when you are done.
7. (Optional) To view the AWS account IDs with which you have shared the AMI, select the AMI in the list, and choose the **Permissions** tab. To find AMIs that are shared with you, see [Finding shared AMIs \(p. 110\)](#).

an AMI (AWS CLI)

Use the `modify-image-attribute` command (AWS CLI) to share an AMI as shown in the following examples.

To grant explicit launch permissions

The following command grants launch permissions for the specified AMI to the specified AWS account.

```
aws ec2 modify-image-attribute \
--image-id ami-0abcdef1234567890 \
--launch-permission "Add=[{UserId=123456789012}]"
```

The following command grants create volume permission for a snapshot.

```
aws ec2 modify-snapshot-attribute \
--snapshot-id snap-1234567890abcdef0 \
--attribute createVolumePermission \
--operation-type add \
--user-ids 123456789012
```

Note

You do not need to share the Amazon EBS snapshots that an AMI references in order to share the AMI. Only the AMI itself needs to be shared; the system automatically provides the instance access to the referenced Amazon EBS snapshots for the launch. However, you do need to share any CMKs used to encrypt snapshots that the AMI references. For more information, see [Sharing an Amazon EBS snapshot \(p. 1073\)](#).

To remove launch permissions for an account

The following command removes launch permissions for the specified AMI from the specified AWS account:

```
aws ec2 modify-image-attribute \
--image-id ami-0abcdef1234567890 \
--launch-permission "Remove=[{UserId=123456789012}]"
```

The following command removes create volume permission for a snapshot.

```
aws ec2 modify-snapshot-attribute \
--snapshot-id snap-1234567890abcdef0 \
--attribute createVolumePermission \
--operation-type remove \
--user-ids 123456789012
```

To remove all launch permissions

The following command removes all public and explicit launch permissions from the specified AMI. Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

```
aws ec2 reset-image-attribute \
--image-id ami-0abcdef1234567890 \
--attribute launchPermission
```

Using bookmarks

If you have created a public AMI, or shared an AMI with another AWS user, you can create a *bookmark* that allows a user to access your AMI and launch an instance in their own account immediately. This is an easy way to share AMI references, so users don't have to spend time finding your AMI in order to use it.

Note that your AMI must be public, or you must have shared it with the user to whom you want to send the bookmark.

To create a bookmark for your AMI

1. Type a URL with the following information, where *region* is the Region in which your AMI resides:

```
https://console.aws.amazon.com/ec2/v2/home?
region=region#LaunchInstanceWizard:ami=ami_id
```

For example, this URL launches an instance from the ami-0abcdef1234567890 AMI in the us-east-1 Region:

```
https://console.aws.amazon.com/ec2/v2/home?region=us-
east-1#LaunchInstanceWizard:ami=ami-0abcdef1234567890
```

2. Distribute the link to users who want to use your AMI.
3. To use a bookmark, choose the link or copy and paste it into your browser. The launch wizard opens, with the AMI already selected.

Guidelines for shared Linux AMIs

Use the following guidelines to reduce the attack surface and improve the reliability of the AMIs you create.

Important

No list of security guidelines can be exhaustive. Build your shared AMIs carefully and take time to consider where you might expose sensitive data.

Contents

- [Update the AMI tools before using them \(p. 116\)](#)
- [Disable password-based remote logins for root \(p. 116\)](#)
- [Disable local root access \(p. 116\)](#)
- [Remove SSH host key pairs \(p. 117\)](#)
- [Install public key credentials \(p. 117\)](#)
- [Disabling sshd DNS checks \(optional\) \(p. 118\)](#)
- [Identify yourself \(p. 119\)](#)
- [Protect yourself \(p. 119\)](#)

If you are building AMIs for AWS Marketplace, see [Building AMIs for AWS Marketplace](#) for guidelines, policies and best practices.

For additional information about sharing AMIs safely, see the following articles:

- [How To Share and Use Public AMIs in A Secure Manner](#)
- [Public AMI Publishing: Hardening and Clean-up Requirements](#)

Update the AMI tools before using them

For AMIs backed by instance store, we recommend that your AMIs download and upgrade the Amazon EC2 AMI creation tools before you use them. This ensures that new AMIs based on your shared AMIs have the latest AMI tools.

For [Amazon Linux 2](#), install the aws-amitools-ec2 package and add the AMI tools to your PATH with the following command. For the [Amazon Linux AMI](#), aws-amitools-ec2 package is already installed by default.

```
[ec2-user ~]$ sudo yum install -y aws-amitools-ec2 && export PATH=$PATH:/opt/aws/bin > /etc/profile.d/aws-amitools-ec2.sh && . /etc/profile.d/aws-amitools-ec2.sh
```

Upgrade the AMI tools with the following command:

```
[ec2-user ~]$ sudo yum upgrade -y aws-amitools-ec2
```

For other distributions, make sure you have the latest AMI tools.

Disable password-based remote logins for root

Using a fixed root password for a public AMI is a security risk that can quickly become known. Even relying on users to change the password after the first login opens a small window of opportunity for potential abuse.

To solve this problem, disable password-based remote logins for the root user.

To disable password-based remote logins for root

1. Open the /etc/ssh/sshd_config file with a text editor and locate the following line:

```
#PermitRootLogin yes
```

2. Change the line to:

```
PermitRootLogin without-password
```

The location of this configuration file might differ for your distribution, or if you are not running OpenSSH. If this is the case, consult the relevant documentation.

Disable local root access

When you work with shared AMIs, a best practice is to disable direct root logins. To do this, log into your running instance and issue the following command:

```
[ec2-user ~]$ sudo passwd -l root
```

Note

This command does not impact the use of `sudo`.

Remove SSH host key pairs

If you plan to share an AMI derived from a public AMI, remove the existing SSH host key pairs located in `/etc/ssh`. This forces SSH to generate new unique SSH key pairs when someone launches an instance using your AMI, improving security and reducing the likelihood of "man-in-the-middle" attacks.

Remove all of the following key files that are present on your system.

- `ssh_host_dsa_key`
- `ssh_host_dsa_key.pub`
- `ssh_host_key`
- `ssh_host_key.pub`
- `ssh_host_rsa_key`
- `ssh_host_rsa_key.pub`
- `ssh_host_ecdsa_key`
- `ssh_host_ecdsa_key.pub`
- `ssh_host_ed25519_key`
- `ssh_host_ed25519_key.pub`

You can securely remove all of these files with the following command.

```
[ec2-user ~]$ sudo shred -u /etc/ssh/*_key /etc/ssh/*_key.pub
```

Warning

Secure deletion utilities such as `shred` may not remove all copies of a file from your storage media. Hidden copies of files may be created by journaling file systems (including Amazon Linux default ext4), snapshots, backups, RAID, and temporary caching. For more information see the [shred documentation](#).

Important

If you forget to remove the existing SSH host key pairs from your public AMI, our routine auditing process notifies you and all customers running instances of your AMI of the potential security risk. After a short grace period, we mark the AMI private.

Install public key credentials

After configuring the AMI to prevent logging in using a password, you must make sure users can log in using another mechanism.

Amazon EC2 allows users to specify a public-private key pair name when launching an instance. When a valid key pair name is provided to the `RunInstances` API call (or through the command line API tools), the public key (the portion of the key pair that Amazon EC2 retains on the server after a call to `CreateKeyPair` or `ImportKeyPair`) is made available to the instance through an HTTP query against the instance metadata.

To log in through SSH, your AMI must retrieve the key value at boot and append it to `/root/.ssh/authorized_keys` (or the equivalent for any other user account on the AMI). Users can launch instances of your AMI with a key pair and log in without requiring a root password.

Many distributions, including Amazon Linux and Ubuntu, use the `cloud-init` package to inject public key credentials for a configured user. If your distribution does not support `cloud-init`, you can add

the following code to a system start-up script (such as `/etc/rc.local`) to pull in the public key you specified at launch for the root user.

Note

In the following example, the IP address `http://169.254.169.254/` is a link-local address and is valid only from the instance.

IMDSv2

```
if [ ! -d /root/.ssh ] ; then
    mkdir -p /root/.ssh
    chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key > /tmp/my-key
if [ $? -eq 0 ] ; then
    cat /tmp/my-key >> /root/.ssh/authorized_keys
    chmod 700 /root/.ssh/authorized_keys
    rm /tmp/my-key
fi
```

IMDSv1

```
if [ ! -d /root/.ssh ] ; then
    mkdir -p /root/.ssh
    chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key > /tmp/my-key
if [ $? -eq 0 ] ; then
    cat /tmp/my-key >> /root/.ssh/authorized_keys
    chmod 700 /root/.ssh/authorized_keys
    rm /tmp/my-key
fi
```

This can be applied to any user account; you do not need to restrict it to `root`.

Note

Rebundling an instance based on this AMI includes the key with which it was launched. To prevent the key's inclusion, you must clear out (or delete) the `authorized_keys` file or exclude this file from rebundling.

Disabling sshd DNS checks (optional)

Disabling sshd DNS checks slightly weakens your sshd security. However, if DNS resolution fails, SSH logins still work. If you do not disable sshd checks, DNS resolution failures prevent all logins.

To disable sshd DNS checks

1. Open the `/etc/ssh/sshd_config` file with a text editor and locate the following line:

```
#UseDNS yes
```

2. Change the line to:

```
UseDNS no
```

Note

The location of this configuration file can differ for your distribution or if you are not running OpenSSH. If this is the case, consult the relevant documentation.

Identify yourself

Currently, there is no easy way to know who provided a shared AMI, because each AMI is represented by an account ID.

We recommend that you post a description of your AMI, and the AMI ID, in the [Amazon EC2 forum](#). This provides a convenient central location for users who are interested in trying new shared AMIs.

Protect yourself

We recommend against storing sensitive data or software on any AMI that you share. Users who launch a shared AMI might be able to rebundle it and register it as their own. Follow these guidelines to help you to avoid some easily overlooked security risks:

- We recommend using the `--exclude directory` option on `ec2-bundle-vol` to skip any directories and subdirectories that contain secret information that you would not like to include in your bundle. In particular, exclude all user-owned SSH public/private key pairs and SSH `authorized_keys` files when bundling the image. The Amazon public AMIs store these in `/root/.ssh` for the root account, and `/home/user_name/.ssh/` for regular user accounts. For more information, see [ec2-bundle-vol \(p. 144\)](#).
- Always delete the shell history before bundling. If you attempt more than one bundle upload in the same AMI, the shell history contains your secret access key. The following example should be the last command executed before bundling from within the instance.

```
[ec2-user ~]$ shred -u ~/.history
```

Warning

The limitations of `shred` described in the warning above apply here as well.

Be aware that bash writes the history of the current session to the disk on exit. If you log out of your instance after deleting `~/.bash_history`, and then log back in, you will find that `~/.bash_history` has been re-created and contains all of the commands executed during your previous session.

Other programs besides bash also write histories to disk. Use caution and remove or exclude unnecessary dot-files and dot-directories.

- Bundling a running instance requires your private key and X.509 certificate. Put these and other credentials in a location that is not bundled (such as the instance store).

Paid AMIs

A *paid AMI* is an AMI that you can purchase from a developer.

Amazon EC2 integrates with AWS Marketplace, enabling developers to charge other Amazon EC2 users for the use of their AMIs or to provide support for instances.

The AWS Marketplace is an online store where you can buy software that runs on AWS, including AMIs that you can use to launch your EC2 instance. The AWS Marketplace AMIs are organized into categories, such as Developer Tools, to enable you to find products to suit your requirements. For more information about AWS Marketplace, see the [AWS Marketplace](#) site.

Launching an instance from a paid AMI is the same as launching an instance from any other AMI. No additional parameters are required. The instance is charged according to the rates set by the owner of

the AMI, as well as the standard usage fees for the related web services, for example, the hourly rate for running an m1.small instance type in Amazon EC2. Additional taxes might also apply. The owner of the paid AMI can confirm whether a specific instance was launched using that paid AMI.

Important

Amazon DevPay is no longer accepting new sellers or products. AWS Marketplace is now the single, unified e-commerce platform for selling software and services through AWS. For information about how to deploy and sell software from AWS Marketplace, see [Selling on AWS Marketplace](#). AWS Marketplace supports AMIs backed by Amazon EBS.

Contents

- [Selling your AMI \(p. 120\)](#)
- [Finding a paid AMI \(p. 120\)](#)
- [Purchasing a paid AMI \(p. 121\)](#)
- [Getting the product code for your instance \(p. 121\)](#)
- [Using paid support \(p. 122\)](#)
- [Bills for paid and supported AMIs \(p. 122\)](#)
- [Managing your AWS Marketplace subscriptions \(p. 122\)](#)

Selling your AMI

You can sell your AMI using AWS Marketplace. AWS Marketplace offers an organized shopping experience. Additionally, AWS Marketplace also supports AWS features such as Amazon EBS-backed AMIs, Reserved Instances, and Spot Instances.

For information about how to sell your AMI on AWS Marketplace, see [Selling on AWS Marketplace](#).

Finding a paid AMI

There are several ways that you can find AMIs that are available for you to purchase. For example, you can use [AWS Marketplace](#), the Amazon EC2 console, or the command line. Alternatively, a developer might let you know about a paid AMI themselves.

Finding a paid AMI using the console

To find a paid AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. Choose **Public images** for the first filter.
4. In the Search bar, choose **Owner**, then **AWS Marketplace**.
5. If you know the product code, choose **Product Code**, then type the product code.

Finding a paid AMI using AWS Marketplace

To find a paid AMI using AWS Marketplace

1. Open [AWS Marketplace](#).
2. Enter the name of the operating system in the search box, and click **Go**.
3. To scope the results further, use one of the categories or filters.
4. Each product is labeled with its product type: either **AMI** or **Software as a Service**.

Finding a paid AMI using the AWS CLI

You can find a paid AMI using the following [describe-images](#) command (AWS CLI).

```
aws ec2 describe-images
--owners aws-marketplace
```

This command returns numerous details that describe each AMI, including the product code for a paid AMI. The output from `describe-images` includes an entry for the product code like the following:

```
"ProductCodes": [
  {
    "ProductCodeId": "product_code",
    "ProductCodeType": "marketplace"
  }
],
```

If you know the product code, you can filter the results by product code. This example returns the most recent AMI with the specified product code.

```
aws ec2 describe-images
--owners aws-marketplace \
--filters "Name=product-code,Values=product_code" \
--query "sort_by(Images, &CreationDate)[-1].[ImageId]"
```

Purchasing a paid AMI

You must sign up for (purchase) a paid AMI before you can launch an instance using the AMI.

Typically a seller of a paid AMI presents you with information about the AMI, including its price and a link where you can buy it. When you click the link, you're first asked to log into AWS, and then you can purchase the AMI.

Purchasing a paid AMI using the console

You can purchase a paid AMI by using the Amazon EC2 launch wizard. For more information, see [Launching an AWS Marketplace instance \(p. 522\)](#).

Subscribing to a product using AWS Marketplace

To use the AWS Marketplace, you must have an AWS account. To launch instances from AWS Marketplace products, you must be signed up to use the Amazon EC2 service, and you must be subscribed to the product from which to launch the instance. There are two ways to subscribe to products in the AWS Marketplace:

- **AWS Marketplace website:** You can launch preconfigured software quickly with the 1-Click deployment feature.
- **Amazon EC2 launch wizard:** You can search for an AMI and launch an instance directly from the wizard. For more information, see [Launching an AWS Marketplace instance \(p. 522\)](#).

Getting the product code for your instance

You can retrieve the AWS Marketplace product code for your instance using its instance metadata. For more information about retrieving metadata, see [Instance metadata and user data \(p. 658\)](#).

To retrieve a product code, use the following command:

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/product-codes
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/product-codes
```

If the instance has a product code, Amazon EC2 returns it.

Using paid support

Amazon EC2 also enables developers to offer support for software (or derived AMIs). Developers can create support products that you can sign up to use. During sign-up for the support product, the developer gives you a product code, which you must then associate with your own AMI. This enables the developer to confirm that your instance is eligible for support. It also ensures that when you run instances of the product, you are charged according to the terms for the product specified by the developer.

Important

You can't use a support product with Reserved Instances. You always pay the price that's specified by the seller of the support product.

To associate a product code with your AMI, use one of the following commands, where *ami_id* is the ID of the AMI and *product_code* is the product code:

- [modify-image-attribute \(AWS CLI\)](#)

```
aws ec2 modify-image-attribute --image-id ami_id --product-codes "product_code"
```

- [Edit-EC2ImageAttribute \(AWS Tools for Windows PowerShell\)](#)

```
PS C:\> Edit-EC2ImageAttribute -ImageId ami_id -ProductCode product_code
```

After you set the product code attribute, it cannot be changed or removed.

Bills for paid and supported AMIs

At the end of each month, you receive an email with the amount your credit card has been charged for using any paid or supported AMIs during the month. This bill is separate from your regular Amazon EC2 bill. For more information, see [Paying For AWS Marketplace Products](#).

Managing your AWS Marketplace subscriptions

On the AWS Marketplace website, you can check your subscription details, view the vendor's usage instructions, manage your subscriptions, and more.

To check your subscription details

1. Log in to the [AWS Marketplace](#).

2. Choose **Your Marketplace Account**.
3. Choose **Manage your software subscriptions**.
4. All your current subscriptions are listed. Choose **Usage Instructions** to view specific instructions for using the product, for example, a user name for connecting to your running instance.

To cancel an AWS Marketplace subscription

1. Ensure that you have terminated any instances running from the subscription.
 - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 - b. In the navigation pane, choose **Instances**.
 - c. Select the instance, and choose **Actions, Instance State, Terminate**.
 - d. Choose **Yes, Terminate** when prompted for confirmation.
2. Log in to the [AWS Marketplace](#), and choose **Your Marketplace Account**, then **Manage your software subscriptions**.
3. Choose **Cancel subscription**. You are prompted to confirm your cancellation.

Note

After you've canceled your subscription, you are no longer able to launch any instances from that AMI. To use that AMI again, you need to resubscribe to it, either on the AWS Marketplace website, or through the launch wizard in the Amazon EC2 console.

Creating an Amazon EBS-backed Linux AMI

To create an Amazon EBS-backed Linux AMI, start from an instance that you've launched from an existing Amazon EBS-backed Linux AMI. This can be an AMI you have obtained from the AWS Marketplace, an AMI you have created using the [AWS Server Migration Service](#) or [VM Import/Export](#), or any other AMI you can access. After you customize the instance to suit your needs, create and register a new AMI, which you can use to launch new instances with these customizations.

The procedures described below work for Amazon EC2 instances backed by encrypted Amazon EBS volumes (including the root volume) as well as for unencrypted volumes.

The AMI creation process is different for instance store-backed AMIs. For more information about the differences between Amazon EBS-backed and instance store-backed instances, and how to determine the root device type for your instance, see [Storage for the root device \(p. 100\)](#). For more information about creating an instance store-backed Linux AMI, see [Creating an instance store-backed Linux AMI \(p. 127\)](#).

For more information about creating an Amazon EBS-backed Windows AMI, see [Creating an Amazon EBS-Backed Windows AMI](#) in the *Amazon EC2 User Guide for Windows Instances*.

Overview of creating Amazon EBS-backed AMIs

First, launch an instance from an AMI that's similar to the AMI that you'd like to create. You can connect to your instance and customize it. When the instance is configured correctly, ensure data integrity by stopping the instance before you create an AMI, then create the image. When you create an Amazon EBS-backed AMI, we automatically register it for you.

Amazon EC2 powers down the instance before creating the AMI to ensure that everything on the instance is stopped and in a consistent state during the creation process. If you're confident that your instance is in a consistent state appropriate for AMI creation, you can tell Amazon EC2 not to power down and reboot the instance. Some file systems, such as XFS, can freeze and unfreeze activity, making it safe to create the image without rebooting the instance.

During the AMI-creation process, Amazon EC2 creates snapshots of your instance's root volume and any other EBS volumes attached to your instance. You're charged for the snapshots until you deregister the AMI and delete the snapshots. For more information, see [Deregistering your Linux AMI \(p. 172\)](#). If any volumes attached to the instance are encrypted, the new AMI only launches successfully on instances that support Amazon EBS encryption. For more information, see [Amazon EBS encryption \(p. 1121\)](#).

Depending on the size of the volumes, it can take several minutes for the AMI-creation process to complete (sometimes up to 24 hours). You may find it more efficient to create snapshots of your volumes before creating your AMI. This way, only small, incremental snapshots need to be created when the AMI is created, and the process completes more quickly (the total time for snapshot creation remains the same). For more information, see [Creating Amazon EBS snapshots \(p. 1063\)](#).

After the process completes, you have a new AMI and snapshot created from the root volume of the instance. When you launch an instance using the new AMI, we create a new EBS volume for its root volume using the snapshot.

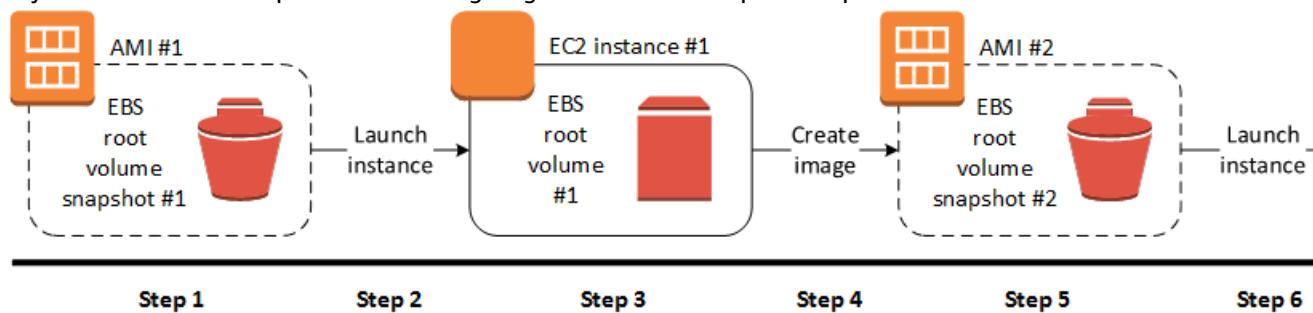
If you add instance-store volumes or EBS volumes to your instance in addition to the root device volume, the block device mapping for the new AMI contains information for these volumes, and the block device mappings for instances that you launch from the new AMI automatically contain information for these volumes. The instance-store volumes specified in the block device mapping for the new instance are new and don't contain any data from the instance store volumes of the instance you used to create the AMI. The data on EBS volumes persists. For more information, see [Block device mapping \(p. 1211\)](#).

Note

When you create a new instance from an EBS-backed AMI, you should initialize both its root volume and any additional EBS storage before putting it into production. For more information, see [Initializing Amazon EBS Volumes](#).

Creating a Linux AMI from an instance

You can create an AMI using the AWS Management Console or the command line. The following diagram summarizes the process for creating an Amazon EBS-backed AMI from a running EC2 instance. Start with an existing AMI, launch an instance, customize it, create a new AMI from it, and finally launch an instance of your new AMI. The steps in the following diagram match the steps in the procedure below.



To create an AMI from an instance using the console

1. Select an appropriate EBS-backed AMI to serve as a starting point for your new AMI, and configure it as needed before launch. For more information, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#).
2. Choose **Launch** to launch an instance of the EBS-backed AMI that you've selected. Accept the default values as you step through the wizard. For more information, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#).
3. While the instance is running, connect to it. You can perform any of the following actions on your instance to customize it for your needs:
 - Install software and applications

- Copy data
 - Reduce start time by deleting temporary files, defragmenting your hard drive, and zeroing out free space
 - Attach additional Amazon EBS volumes
4. (Optional) Create snapshots of all the volumes attached to your instance. For more information about creating snapshots, see [Creating Amazon EBS snapshots \(p. 1063\)](#).
 5. In the navigation pane, choose **Instances**, select your instance, and then choose **Actions, Image, Create Image**.

Tip

If this option is disabled, your instance isn't an Amazon EBS-backed instance.

6. In the **Create Image** dialog box, specify the following information, and then choose **Create Image**.
 - **Image name** – A unique name for the image.
 - **Image description** – An optional description of the image, up to 255 characters.
 - **No reboot** – This option is not selected by default. Amazon EC2 shuts down the instance, takes snapshots of any attached volumes, creates and registers the AMI, and then reboots the instance. Select **No reboot** to avoid having your instance shut down.

Warning

If you select **No reboot**, we can't guarantee the file system integrity of the created image.

- **Instance Volumes** – The fields in this section enable you to modify the root volume, and add additional Amazon EBS and instance store volumes. For information about each field, pause on the **i** icon next to each field to display field tooltips. Some important points are listed below.
 - To change the size of the root volume, locate **Root** in the **Volume Type** column, and for **Size (GiB)**, type the required value.
 - If you select **Delete on Termination**, when you terminate the instance created from this AMI, the EBS volume is deleted. If you clear **Delete on Termination**, when you terminate the instance, the EBS volume is not deleted. For more information, see [Preserving Amazon EBS volumes on instance termination \(p. 610\)](#).
 - To add an Amazon EBS volume, choose **Add New Volume** (which adds a new row). For **Volume Type**, choose **EBS**, and fill in the fields in the row. When you launch an instance from your new AMI, additional volumes are automatically attached to the instance. Empty volumes must be formatted and mounted. Volumes based on a snapshot must be mounted.
 - To add an instance store volume, see [Adding instance store volumes to an AMI \(p. 1197\)](#). When you launch an instance from your new AMI, additional volumes are automatically initialized and mounted. These volumes do not contain data from the instance store volumes of the running instance on which you based your AMI.
7. To view the status of your AMI while it is being created, in the navigation pane, choose **AMIs**. Initially, the status is **Pending** but should change to **Available** after a few minutes.

(Optional) To view the snapshot that was created for the new AMI, choose **Snapshots**. When you launch an instance from this AMI, we use this snapshot to create its root device volume.
8. Launch an instance from your new AMI. For more information, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#).
9. The new running instance contains all of the customizations that you applied in previous steps.

To create an AMI from an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [create-image \(AWS CLI\)](#)

- [New-EC2Image](#) (AWS Tools for Windows PowerShell)

Creating a Linux AMI from a snapshot

If you have a snapshot of the root device volume of an instance, you can create an AMI from this snapshot using the AWS Management Console or the command line.

To create an AMI from a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Elastic Block Store**, choose **Snapshots**.
3. Choose the snapshot and choose **Actions, Create Image**.
4. In the **Create Image from EBS Snapshot** dialog box, complete the fields to create your AMI, then choose **Create**. If you're re-creating a parent instance, then choose the same options as the parent instance.
 - **Architecture:** Choose **i386** for 32-bit or **x86_64** for 64-bit.
 - **Root device name:** Enter the appropriate name for the root volume. For more information, see [Device naming on Linux instances \(p. 1210\)](#).
 - **Virtualization type:** Choose whether instances launched from this AMI use paravirtual (PV) or hardware virtual machine (HVM) virtualization. For more information, see [Linux AMI virtualization types \(p. 102\)](#).
 - (PV virtualization type only) **Kernel ID** and **RAM disk ID:** Choose the AKI and ARI from the lists. If you choose the default AKI or don't choose an AKI, you must specify an AKI every time you launch an instance using this AMI. In addition, your instance may fail the health checks if the default AKI is incompatible with the instance.
 - (Optional) **Block Device Mappings:** Add volumes or expand the default size of the root volume for the AMI. For more information about resizing the file system on your instance for a larger volume, see [Extending a Linux file system after resizing a volume \(p. 1117\)](#).

To create an AMI from a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [register-image](#) (AWS CLI)
- [Register-EC2Image](#) (AWS Tools for Windows PowerShell)

Launching an instance from an AMI you created

You can launch an instance from an AMI that you created from an instance or snapshot.

To launch an instance from your AMI

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Images**, choose **AMIs**.
3. Set the filter to **Owned by me** and select your AMI.
4. Choose **Actions, Launch**.
5. Follow the wizard to launch your instance. For more information about each step in the wizard, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#).

Creating an instance store-backed Linux AMI

The AMI that you specify when you launch your instance determines the type of root device volume

To create an instance store-backed Linux AMI, start from an instance that you've launched from an existing instance store-backed Linux AMI. After you've customized the instance to suit your needs, bundle the volume and register a new AMI, which you can use to launch new instances with these customizations.

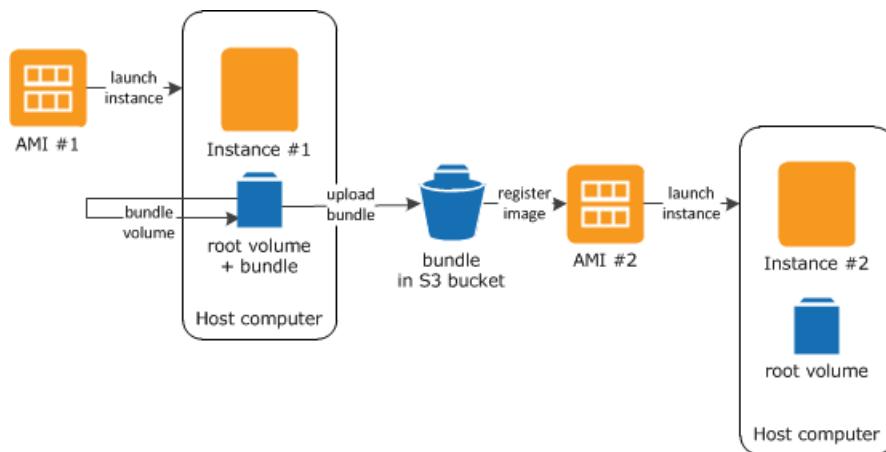
Important

Only the following instance types support an instance store volume as the root device: C3, D2, G2, I2, M3, and R3.

The AMI creation process is different for Amazon EBS-backed AMIs. For more information about the differences between Amazon EBS-backed and instance store-backed instances, and how to determine the root device type for your instance, see [Storage for the root device \(p. 100\)](#). If you need to create an Amazon EBS-backed Linux AMI, see [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#).

Overview of the creation process for instance store-backed AMIs

The following diagram summarizes the process of creating an AMI from an instance store-backed instance.



First, launch an instance from an AMI that's similar to the AMI that you'd like to create. You can connect to your instance and customize it. When the instance is set up the way you want it, you can bundle it. It takes several minutes for the bundling process to complete. After the process completes, you have a bundle, which consists of an image manifest (`image.manifest.xml`) and files (`image.part.xx`) that contain a template for the root volume. Next you upload the bundle to your Amazon S3 bucket and then register your AMI.

When you launch an instance using the new AMI, we create the root volume for the instance using the bundle that you uploaded to Amazon S3. The storage space used by the bundle in Amazon S3 incurs charges to your account until you delete it. For more information, see [Deregistering your Linux AMI \(p. 172\)](#).

If you add instance store volumes to your instance in addition to the root device volume, the block device mapping for the new AMI contains information for these volumes, and the block device mappings for instances that you launch from the new AMI automatically contain information for these volumes. For more information, see [Block device mapping \(p. 1211\)](#).

Prerequisites

Before you can create an AMI, you must complete the following tasks:

- Install the AMI tools. For more information, see [Setting up the AMI tools \(p. 128\)](#).
- Install the AWS CLI. For more information, see [Getting Set Up with the AWS Command Line Interface](#).
- Ensure that you have an Amazon S3 bucket for the bundle. To create an Amazon S3 bucket, open the Amazon S3 console and click **Create Bucket**. Alternatively, you can use the AWS CLI `mb` command.
- Ensure that you have your AWS account ID. For more information, see [AWS Account Identifiers](#) in the [AWS General Reference](#).
- Ensure that you have your access key ID and secret access key. For more information, see [Access Keys](#) in the [AWS General Reference](#).
- Ensure that you have an X.509 certificate and corresponding private key.
 - If you need to create an X.509 certificate, see [Managing signing certificates \(p. 130\)](#). The X.509 certificate and private key are used to encrypt and decrypt your AMI.
 - [China (Beijing)] Use the `$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2-cn-north-1.pem` certificate.
 - [AWS GovCloud (US-West)] Use the `$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2-gov.pem` certificate.
- Connect to your instance and customize it. For example, you can install software and applications, copy data, delete temporary files, and modify the Linux configuration.

Tasks

- [Setting up the AMI tools \(p. 128\)](#)
- [Creating an AMI from an instance store-backed Amazon Linux instance \(p. 131\)](#)
- [Creating an AMI from an instance store-backed Ubuntu instance \(p. 134\)](#)
- [Converting your instance store-backed AMI to an Amazon EBS-backed AMI \(p. 138\)](#)

Setting up the AMI tools

You can use the AMI tools to create and manage instance store-backed Linux AMIs. To use the tools, you must install them on your Linux instance. The AMI tools are available as both an RPM and as a .zip file for Linux distributions that don't support RPM.

To set up the AMI tools using the RPM

1. Install Ruby using the package manager for your Linux distribution, such as yum. For example:

```
[ec2-user ~]$ sudo yum install -y ruby
```

2. Download the RPM file using a tool such as wget or curl. For example:

```
[ec2-user ~]$ wget https://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.noarch.rpm
```

3. Verify the RPM file's signature using the following command:

```
[ec2-user ~]$ rpm -K ec2-ami-tools.noarch.rpm
```

The command above should indicate that the file's SHA1 and MD5 hashes are OK. If the command indicates that the hashes are NOT OK, use the following command to view the file's Header SHA1 and MD5 hashes:

```
[ec2-user ~]$ rpm -Kv ec2-ami-tools.noarch.rpm
```

Then, compare your file's Header SHA1 and MD5 hashes with the following verified AMI tools hashes to confirm the file's authenticity:

- Header SHA1: a1f662d6f25f69871104e6a62187fa4df508f880
- MD5: 9faff05258064e2f7909b66142de6782

If your file's Header SHA1 and MD5 hashes match the verified AMI tools hashes, continue to the next step.

4. Install the RPM using the following command:

```
[ec2-user ~]$ sudo yum install ec2-ami-tools.noarch.rpm
```

5. Verify your AMI tools installation using the [ec2-ami-tools-version \(p. 141\)](#) command.

```
[ec2-user ~]$ ec2-ami-tools-version
```

Note

If you receive a load error such as "cannot load such file -- ec2/amitools/version (LoadError)", complete the next step to add the location of your AMI tools installation to your RUBYLIB path.

6. (Optional) If you received an error in the previous step, add the location of your AMI tools installation to your RUBYLIB path.

- a. Run the following command to determine the paths to add.

```
[ec2-user ~]$ rpm -qil ec2-ami-tools | grep ec2/amitools/version
/usr/lib/ruby/site_ruby/ec2/amitools/version.rb
/usr/lib64/ruby/site_ruby/ec2/amitools/version.rb
```

In the above example, the missing file from the previous load error is located at /usr/lib/ruby/site_ruby and /usr/lib64/ruby/site_ruby.

- b. Add the locations from the previous step to your RUBYLIB path.

```
[ec2-user ~]$ export RUBYLIB=$RUBYLIB:/usr/lib/ruby/site_ruby:/usr/lib64/ruby/site_ruby
```

- c. Verify your AMI tools installation using the [ec2-ami-tools-version \(p. 141\)](#) command.

```
[ec2-user ~]$ ec2-ami-tools-version
```

To set up the AMI tools using the .zip file

1. Install Ruby and unzip using the package manager for your Linux distribution, such as **apt-get**. For example:

```
[ec2-user ~]$ sudo apt-get update -y && sudo apt-get install -y ruby unzip
```

2. Download the .zip file using a tool such as wget or curl. For example:

```
[ec2-user ~]$ wget https://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.zip
```

3. Unzip the files into a suitable installation directory, such as /usr/local/ec2.

```
[ec2-user ~]$ sudo mkdir -p /usr/local/ec2  
$ sudo unzip ec2-ami-tools.zip -d /usr/local/ec2
```

Notice that the .zip file contains a folder ec2-ami-tools-**x.x.x**, where **x.x.x** is the version number of the tools (for example, ec2-ami-tools-1.5.7).

4. Set the EC2_AMITOOL_HOME environment variable to the installation directory for the tools. For example:

```
[ec2-user ~]$ export EC2_AMITOOL_HOME=/usr/local/ec2/ec2-ami-tools-x.x.x
```

5. Add the tools to your PATH environment variable. For example:

```
[ec2-user ~]$ export PATH=$EC2_AMITOOL_HOME/bin:$PATH
```

6. You can verify your AMI tools installation using the [ec2-ami-tools-version \(p. 141\)](#) command.

```
[ec2-user ~]$ ec2-ami-tools-version
```

Managing signing certificates

Certain commands in the AMI tools require a signing certificate (also known as X.509 certificate). You must create the certificate and then upload it to AWS. For example, you can use a third-party tool such as OpenSSL to create the certificate.

To create a signing certificate

1. Install and configure OpenSSL.
2. Create a private key using the openssl genrsa command and save the output to a .pem file. We recommend that you create a 2048- or 4096-bit RSA key.

```
openssl genrsa 2048 > private-key.pem
```

3. Generate a certificate using the openssl req command.

```
openssl req -new -x509 -nodes -sha256 -days 365 -key private-key.pem -outform PEM -  
out certificate.pem
```

To upload the certificate to AWS, use the [upload-signing-certificate](#) command.

```
aws iam upload-signing-certificate --user-name user-name --certificate-body file://path/to/certificate.pem
```

To list the certificates for a user, use the [list-signing-certificates](#) command:

```
aws iam list-signing-certificates --user-name user-name
```

To disable or re-enable a signing certificate for a user, use the [update-signing-certificate](#) command. The following command disables the certificate:

```
aws iam update-signing-certificate --certificate-id OFHPLP4ZULTHYPMSYEX7O4BEXAMPLE --  
status Inactive --user-name user-name
```

To delete a certificate, use the [delete-signing-certificate](#) command:

```
aws iam delete-signing-certificate --user-name user-name --certificate-  
id OFHPLP4ZULTHYPMSYEX7O4BEXAMPLE
```

Creating an AMI from an instance store-backed instance

The following procedures are for creating an instance store-backed AMI from an instance store-backed instance. Before you begin, ensure that you've read the [Prerequisites \(p. 128\)](#).

Topics

- [Creating an AMI from an instance store-backed Amazon Linux instance \(p. 131\)](#)
- [Creating an AMI from an instance store-backed Ubuntu instance \(p. 134\)](#)

Creating an AMI from an instance store-backed Amazon Linux instance

This section describes the creation of an AMI from an Amazon Linux instance. The following procedures may not work for instances running other Linux distributions. For Ubuntu-specific procedures, see [Creating an AMI from an instance store-backed Ubuntu instance \(p. 134\)](#).

To prepare to use the AMI tools (HVM instances only)

1. The AMI tools require GRUB Legacy to boot properly. Use the following command to install GRUB:

```
[ec2-user ~]$ sudo yum install -y grub
```

2. Install the partition management packages with the following command:

```
[ec2-user ~]$ sudo yum install -y gdisk kpartx parted
```

To create an AMI from an instance store-backed Amazon Linux instance

This procedure assumes that you have satisfied the prerequisites in [Prerequisites \(p. 128\)](#).

1. Upload your credentials to your instance. We use these credentials to ensure that only you and Amazon EC2 can access your AMI.
 - a. Create a temporary directory on your instance for your credentials as follows:

```
[ec2-user ~]$ mkdir /tmp/cert
```

This enables you to exclude your credentials from the created image.

- b. Copy your X.509 certificate and corresponding private key from your computer to the `/tmp/cert` directory on your instance using a secure copy tool such as [scp \(p. 567\)](#). The `-i my-private-key.pem` option in the following `scp` command is the private key you use to connect to your instance with SSH, not the X.509 private key. For example:

```
you@your_computer:~ $ scp -i my-private-key.pem /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem /path/to/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-  
user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/  
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 717 0.7KB/s 00:00  
cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 685 0.7KB/s 00:00
```

Alternatively, because these are plain text files, you can open the certificate and key in a text editor and copy their contents into new files in `/tmp/cert`.

2. Prepare the bundle to upload to Amazon S3 by running the [ec2-bundle-vol \(p. 144\)](#) command from inside your instance. Be sure to specify the `-e` option to exclude the directory where your credentials are stored. By default, the bundle process excludes files that might contain sensitive information. These files include `*.sw`, `*.swo`, `*.swp`, `*.pem`, `*.priv`, `*id_rsa`, `*id_dsa`, `*.gpg`, `*.jks`, `*/.ssh/authorized_keys`, and `*/.bash_history`. To include all of these files, use the `--no-filter` option. To include some of these files, use the `--include` option.

Important

By default, the AMI bundling process creates a compressed, encrypted collection of files in the `/tmp` directory that represents your root volume. If you do not have enough free disk space in `/tmp` to store the bundle, you need to specify a different location for the bundle to be stored with the `-d /path/to/bundle/storage` option. Some instances have ephemeral storage mounted at `/mnt` or `/media/ephemeral0` that you can use, or you can also [create \(p. 1040\)](#), [attach \(p. 1042\)](#), and [mount \(p. 1046\)](#) a new Amazon EBS volume to store the bundle.

- a. You must run the `ec2-bundle-vol` command as root. For most commands, you can use `sudo` to gain elevated permissions, but in this case, you should run `sudo -E su` to keep your environment variables.

```
[ec2-user ~]$ sudo -E su
```

Note that bash prompt now identifies you as the root user, and that the dollar sign has been replaced by a hash tag, signalling that you are in a root shell:

```
[root ec2-user]#
```

- b. To create the AMI bundle, run the [ec2-bundle-vol \(p. 144\)](#) command as follows:

```
[root ec2-user]# ec2-bundle-vol -k /tmp/cert/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c /tmp/cert/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 123456789012 -r x86_64 -e /tmp/cert --partition gpt
```

Note

For the China (Beijing) and AWS GovCloud (US-West) Regions, use the `--ec2cert` parameter and specify the certificates as per the [prerequisites \(p. 128\)](#).

It can take a few minutes to create the image. When this command completes, your /tmp (or non-default) directory contains the bundle (`image.manifest.xml`, plus multiple `image.part.xx` files).

- c. Exit from the root shell.

```
[root ec2-user]# exit
```

3. (Optional) To add more instance store volumes, edit the block device mappings in the `image.manifest.xml` file for your AMI. For more information, see [Block device mapping \(p. 1211\)](#).
 - a. Create a backup of your `image.manifest.xml` file.

```
[ec2-user ~]$ sudo cp /tmp/image.manifest.xml /tmp/image.manifest.xml.bak
```

- b. Reformat the `image.manifest.xml` file so that it is easier to read and edit.

```
[ec2-user ~]$ sudo xmllint --format /tmp/image.manifest.xml.bak > sudo /tmp/image.manifest.xml
```

- c. Edit the block device mappings in `image.manifest.xml` with a text editor. The example below shows a new entry for the `ephemeral1` instance store volume.

Note

For a list of excluded files, see [ec2-bundle-vol \(p. 144\)](#).

```
<block_device_mapping>
  <mapping>
    <virtual>ami</virtual>
    <device>sda</device>
  </mapping>
  <mapping>
    <virtual>ephemeral0</virtual>
    <device>sdb</device>
  </mapping>
  <mapping>
    <virtual>ephemeral1</virtual>
    <device>sdc</device>
  </mapping>
  <mapping>
    <virtual>root</virtual>
    <device>/dev/sda1</device>
  </mapping>
</block_device_mapping>
```

- d. Save the `image.manifest.xml` file and exit your text editor.
4. To upload your bundle to Amazon S3, run the [ec2-upload-bundle \(p. 154\)](#) command as follows.

```
[ec2-user ~]$ ec2-upload-bundle -b my-s3-bucket/bundle_folder/bundle_name -m /tmp/image.manifest.xml -a your_access_key_id -s your_secret_access_key
```

Important

To register your AMI in a Region other than US East (N. Virginia), you must specify both the target Region with the `--region` option and a bucket path that already exists in the target Region or a unique bucket path that can be created in the target Region.

5. (Optional) After the bundle is uploaded to Amazon S3, you can remove the bundle from the /tmp directory on the instance using the following `rm` command:

```
[ec2-user ~]$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
```

Important

If you specified a path with the `-d /path/to/bundle/storage` option in [Step 2 \(p. 132\)](#), use that path instead of `/tmp`.

6. To register your AMI, run the [register-image](#) command as follows.

```
[ec2-user ~]$ aws ec2 register-image --image-location my-s3-bucket/bundle_folder/bundle_name/image.manifest.xml --name AMI_name --virtualization-type hvm
```

Important

If you previously specified a Region for the [ec2-upload-bundle \(p. 154\)](#) command, specify that Region again for this command.

Creating an AMI from an instance store-backed Ubuntu instance

This section describes the creation of an AMI from an Ubuntu Linux instance with an instance store volume as the root volume. The following procedures may not work for instances running other Linux distributions. For procedures specific to Amazon Linux, see [Creating an AMI from an instance store-backed Amazon Linux instance \(p. 131\)](#).

To prepare to use the AMI tools (HVM instances only)

The AMI tools require GRUB Legacy to boot properly. However, Ubuntu is configured to use GRUB 2. You must check to see that your instance uses GRUB Legacy, and if not, you need to install and configure it.

HVM instances also require partitioning tools to be installed for the AMI tools to work properly.

1. GRUB Legacy (version `0.9x` or less) must be installed on your instance. Check to see if GRUB Legacy is present and install it if necessary.
 - a. Check the version of your GRUB installation.

```
ubuntu:~$ grub-install --version
grub-install (GRUB) 1.99-21ubuntu3.10
```

In this example, the GRUB version is greater than `0.9x`, so you must install GRUB Legacy. Proceed to [Step 1.b \(p. 134\)](#). If GRUB Legacy is already present, you can skip to [Step 2 \(p. 134\)](#).

- b. Install the `grub` package using the following command.

```
ubuntu:~$ sudo apt-get install -y grub
```

2. Install the following partition management packages using the package manager for your distribution.
 - `gdisk` (some distributions may call this package `gptfdisk` instead)
 - `kpartx`
 - `parted`

Use the following command.

```
ubuntu:~$ sudo apt-get install -y gdisk kpartx parted
```

3. Check the kernel parameters for your instance.

```
ubuntu:~$ cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-3.2.0-54-virtual root=UUID=4f392932-ed93-4f8f-
aee7-72bc5bb6ca9d ro console=ttyS0 xen_emul_unplug=unnecessary
```

Note the options following the kernel and root device parameters: `ro`, `console=ttyS0`, and `xen_emul_unplug=unnecessary`. Your options may differ.

4. Check the kernel entries in `/boot/grub/menu.lst`.

```
ubuntu:~$ grep ^kernel /boot/grub/menu.lst
kernel   /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro console=hvc0
kernel   /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro single
kernel   /boot/memtest86+.bin
```

Note that the `console` parameter is pointing to `hvc0` instead of `ttyS0` and that the `xen_emul_unplug=unnecessary` parameter is missing. Again, your options may differ.

5. Edit the `/boot/grub/menu.lst` file with your favorite text editor (such as **vim** or **nano**) to change the `console` and add the parameters you identified earlier to the boot entries.

```
title      Ubuntu 12.04.3 LTS, kernel 3.2.0-54-virtual
root      (hd0)
kernel    /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs
          ro console=ttyS0 xen_emul_unplug=unnecessary
initrd    /boot/initrd.img-3.2.0-54-virtual

title      Ubuntu 12.04.3 LTS, kernel 3.2.0-54-virtual (recovery mode)
root      (hd0)
kernel    /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro
          single console=ttyS0 xen_emul_unplug=unnecessary
initrd    /boot/initrd.img-3.2.0-54-virtual

title      Ubuntu 12.04.3 LTS, memtest86+
root      (hd0)
kernel    /boot/memtest86+.bin
```

6. Verify that your kernel entries now contain the correct parameters.

```
ubuntu:~$ grep ^kernel /boot/grub/menu.lst
kernel   /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro console=ttyS0
          xen_emul_unplug=unnecessary
kernel   /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro single
          console=ttyS0 xen_emul_unplug=unnecessary
kernel   /boot/memtest86+.bin
```

7. [For Ubuntu 14.04 and later only] Starting with Ubuntu 14.04, instance store backed Ubuntu AMIs use a GPT partition table and a separate EFI partition mounted at `/boot/efi`. The **ec2-bundle-vol** command will not bundle this boot partition, so you need to comment out the `/etc/fstab` entry for the EFI partition as shown in the following example.

```
LABEL=cloudimg-rootfs   /           ext4  defaults          0 0
#LABEL=UEFI            /boot/efi     vfat   defaults          0 0
/dev/xvdb             /mnt        auto   defaults,nobootwait,comment=cloudconfig 0          2
```

To create an AMI from an instance store-backed Ubuntu instance

This procedure assumes that you have satisfied the prerequisites in [Prerequisites \(p. 128\)](#).

1. Upload your credentials to your instance. We use these credentials to ensure that only you and Amazon EC2 can access your AMI.

- a. Create a temporary directory on your instance for your credentials as follows:

```
ubuntu:~$ mkdir /tmp/cert
```

This enables you to exclude your credentials from the created image.

- b. Copy your X.509 certificate and private key from your computer to the /tmp/cert directory on your instance, using a secure copy tool such as [scp \(p. 567\)](#). The -i *my-private-key.pem* option in the following **scp** command is the private key you use to connect to your instance with SSH, not the X.509 private key. For example:

```
you@your_computer:~ $ scp -i my-private-key.pem /  
path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem /  
path/to/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-  
user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/  
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 717 0.7KB/s 00:00  
cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 685 0.7KB/s 00:00
```

Alternatively, because these are plain text files, you can open the certificate and key in a text editor and copy their contents into new files in /tmp/cert.

2. Prepare the bundle to upload to Amazon S3 by running the [ec2-bundle-vol \(p. 144\)](#) command from your instance. Be sure to specify the -e option to exclude the directory where your credentials are stored. By default, the bundle process excludes files that might contain sensitive information. These files include *.sw, *.swo, *.swp, *.pem, *.priv, *id_rsa*, *id_dsa*, *.gpg, *.jks, */.ssh/authorized_keys, and */.bash_history. To include all of these files, use the --no-filter option. To include some of these files, use the --include option.

Important

By default, the AMI bundling process creates a compressed, encrypted collection of files in the /tmp directory that represents your root volume. If you do not have enough free disk space in /tmp to store the bundle, you need to specify a different location for the bundle to be stored with the -d */path/to/bundle/storage* option. Some instances have ephemeral storage mounted at /mnt or /media/ephemeral0 that you can use, or you can also [create \(p. 1040\)](#), [attach \(p. 1042\)](#), and [mount \(p. 1046\)](#) a new Amazon EBS volume to store the bundle.

- a. You must run the **ec2-bundle-vol** command needs as root. For most commands, you can use **sudo** to gain elevated permissions, but in this case, you should run **sudo -E su** to keep your environment variables.

```
ubuntu:~$ sudo -E su
```

Note that bash prompt now identifies you as the root user, and that the dollar sign has been replaced by a hash tag, signalling that you are in a root shell:

```
root@ubuntu:~#
```

- b. To create the AMI bundle, run the [ec2-bundle-vol \(p. 144\)](#) command as follows.

```
root@ubuntu:# ec2-bundle-vol -k /tmp/cert/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem  
-c /tmp/cert/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u your_aws_account_id -r  
x86_64 -e /tmp/cert --partition gpt
```

Important

For Ubuntu 14.04 and later HVM instances, add the `--partition mbr` flag to bundle the boot instructions properly; otherwise, your newly-created AMI will not boot.

It can take a few minutes to create the image. When this command completes, your `tmp` directory contains the bundle (`image.manifest.xml`, plus multiple `image.part.xx` files).

- c. Exit from the root shell.

```
root@ubuntu:# exit
```

3. (Optional) To add more instance store volumes, edit the block device mappings in the `image.manifest.xml` file for your AMI. For more information, see [Block device mapping \(p. 1211\)](#).

- a. Create a backup of your `image.manifest.xml` file.

```
ubuntu:~$ sudo cp /tmp/image.manifest.xml /tmp/image.manifest.xml.bak
```

- b. Reformat the `image.manifest.xml` file so that it is easier to read and edit.

```
ubuntu:~$ sudo xmllint --format /tmp/image.manifest.xml.bak > /tmp/  
image.manifest.xml
```

- c. Edit the block device mappings in `image.manifest.xml` with a text editor. The example below shows a new entry for the `ephemeral1` instance store volume.

```
<block_device_mapping>  
  <mapping>  
    <virtual>ami</virtual>  
    <device>sda</device>  
  </mapping>  
  <mapping>  
    <virtual>ephemeral0</virtual>  
    <device>sdb</device>  
  </mapping>  
  <mapping>  
    <virtual>ephemeral1</virtual>  
    <device>sdc</device>  
  </mapping>  
  <mapping>  
    <virtual>root</virtual>  
    <device>/dev/sda1</device>  
  </mapping>  
</block_device_mapping>
```

- d. Save the `image.manifest.xml` file and exit your text editor.
4. To upload your bundle to Amazon S3, run the [ec2-upload-bundle \(p. 154\)](#) command as follows.

```
ubuntu:~$ ec2-upload-bundle -b my-s3-bucket/bundle_folder/bundle_name -m /tmp/  
image.manifest.xml -a your_access_key_id -s your_secret_access_key
```

Important

If you intend to register your AMI in a Region other than US East (N. Virginia), you must specify both the target Region with the `--region` option and a bucket path that already exists in the target Region or a unique bucket path that can be created in the target Region.

5. (Optional) After the bundle is uploaded to Amazon S3, you can remove the bundle from the `/tmp` directory on the instance using the following `rm` command:

```
ubuntu:~$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
```

Important

If you specified a path with the `-d` `/path/to/bundle/storage` option in Step 2 (p. 136), use that same path below, instead of `/tmp`.

6. To register your AMI, run the `register-image` AWS CLI command as follows.

```
ubuntu:~$ aws ec2 register-image --image-location my-s3-bucket/bundle_folder/bundle_name/image.manifest.xml --name AMI_name --virtualization-type hvm
```

Important

If you previously specified a Region for the `ec2-upload-bundle` (p. 154) command, specify that Region again for this command.

7. [Ubuntu 14.04 and later] Uncomment the EFI entry in `/etc/fstab`; otherwise, your running instance will not be able to reboot.

Converting your instance store-backed AMI to an Amazon EBS-backed AMI

You can convert an instance store-backed Linux AMI that you own to an Amazon EBS-backed Linux AMI.

Important

You can't convert an instance store-backed Windows AMI to an Amazon EBS-backed Windows AMI and you cannot convert an AMI that you do not own.

To convert an instance store-backed AMI to an Amazon EBS-backed AMI

1. Launch an Amazon Linux instance from an Amazon EBS-backed AMI. For more information, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#). Amazon Linux instances have the AWS CLI and AMI tools pre-installed.
2. Upload the X.509 private key that you used to bundle your instance store-backed AMI to your instance. We use this key to ensure that only you and Amazon EC2 can access your AMI.
 - a. Create a temporary directory on your instance for your X.509 private key as follows:

```
[ec2-user ~]$ mkdir /tmp/cert
```

- b. Copy your X.509 private key from your computer to the `/tmp/cert` directory on your instance, using a secure copy tool such as `scp` (p. 567). The `my-private-key` parameter in the following command is the private key you use to connect to your instance with SSH. For example:

```
you@your_computer:~ $ scp -i my-private-key.pem /  
path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-  
user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
```

```
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 717      0.7KB/s  00:00
```

- Set environment variables for your AWS access key and secret key.

```
[ec2-user ~]$ export AWS_ACCESS_KEY_ID=your_access_key_id
[ec2-user ~]$ export AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

- Prepare an Amazon EBS volume for your new AMI.

- Create an empty Amazon EBS volume in the same Availability Zone as your instance using the [create-volume](#) command. Note the volume ID in the command output.

Important

This Amazon EBS volume must be the same size or larger than the original instance store root volume.

```
[ec2-user ~]$ aws ec2 create-volume --size 10 --region us-west-2 --availability-zone us-west-2b
```

- Attach the volume to your Amazon EBS-backed instance using the [attach-volume](#) command.

```
[ec2-user ~]$ aws ec2 attach-volume --volume-id volume_id --instance-id instance_id
--device /dev/sdb --region us-west-2
```

- Create a folder for your bundle.

```
[ec2-user ~]$ mkdir /tmp/bundle
```

- Download the bundle for your instance store-based AMI to /tmp/bundle using the [ec2-download-bundle](#) (p. 150) command.

```
[ec2-user ~]$ ec2-download-bundle -b my-s3-bucket/bundle_folder/bundle_name -m
image.manifest.xml -a $AWS_ACCESS_KEY_ID -s $AWS_SECRET_ACCESS_KEY --privatekey /path/
to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d /tmp/bundle
```

- Reconstitute the image file from the bundle using the [ec2-unbundle](#) (p. 153) command.

- Change directories to the bundle folder.

```
[ec2-user ~]$ cd /tmp/bundle/
```

- Run the [ec2-unbundle](#) (p. 153) command.

```
[ec2-user bundle]$ ec2-unbundle -m image.manifest.xml --privatekey /path/to/pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
```

- Copy the files from the unbundled image to the new Amazon EBS volume.

```
[ec2-user bundle]$ sudo dd if=/tmp/bundle/image of=/dev/sdb bs=1M
```

- Probe the volume for any new partitions that were unbundled.

```
[ec2-user bundle]$ sudo partprobe /dev/sdb1
```

- List the block devices to find the device name to mount.

```
[ec2-user bundle]$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
/dev/sda    202:0    0   8G  0 disk
```

```
##/dev/sda1 202:1    0   8G  0 part /
/dev/sdb    202:80   0  10G  0 disk
##/dev/sdb1 202:81   0  10G  0 part
```

In this example, the partition to mount is `/dev/sdb1`, but your device name will likely be different. If your volume is not partitioned, then the device to mount will be similar to `/dev/sdb` (without a device partition trailing digit).

11. Create a mount point for the new Amazon EBS volume and mount the volume.

```
[ec2-user bundle]$ sudo mkdir /mnt/ebs
[ec2-user bundle]$ sudo mount /dev/sdb1 /mnt/ebs
```

12. Open the `/etc/fstab` file on the EBS volume with your favorite text editor (such as `vim` or `nano`) and remove any entries for instance store (ephemeral) volumes. Because the Amazon EBS volume is mounted on `/mnt/ebs`, the `fstab` file is located at `/mnt/ebs/etc/fstab`.

```
[ec2-user bundle]$ sudo nano /mnt/ebs/etc/fstab
#
LABEL=/      /          ext4      defaults,noatime 1  1
tmpfs       /dev/shm    tmpfs     defaults        0  0
devpts      /dev/pts    devpts    gid=5,mode=620 0  0
sysfs       /sys        sysfs    defaults        0  0
proc         /proc       proc     defaults        0  0
/dev/sdb     /media/ephemeral0 auto      defaults,comment=cloudconfig 0
2
```

In this example, the last line should be removed.

13. Unmount the volume and detach it from the instance.

```
[ec2-user bundle]$ sudo umount /mnt/ebs
[ec2-user bundle]$ aws ec2 detach-volume --volume-id volume_id --region us-west-2
```

14. Create an AMI from the new Amazon EBS volume as follows.

- Create a snapshot of the new Amazon EBS volume.

```
[ec2-user bundle]$ aws ec2 create-snapshot --region us-west-2 --description
"your_snapshot_description" --volume-id volume_id
```

- Check to see that your snapshot is complete.

```
[ec2-user bundle]$ aws ec2 describe-snapshots --region us-west-2 --snapshot-
id snapshot_id
```

- Identify the processor architecture, virtualization type, and the kernel image (aki) used on the original AMI with the `describe-images` command. You need the AMI ID of the original instance store-backed AMI for this step.

```
[ec2-user bundle]$ aws ec2 describe-images --region us-west-2 --image-id ami_id --
output text
IMAGES x86_64 amazon/amzn-ami-pv-2013.09.2.x86_64-s3 ami-8ef297be amazon available
public machine aki-fc8f11cc instance-store paravirtual xen
```

In this example, the architecture is `x86_64` and the kernel image ID is `aki-fc8f11cc`. Use these values in the following step. If the output of the above command also lists an `ari` ID, take note of that as well.

- d. Register your new AMI with the snapshot ID of your new Amazon EBS volume and the values from the previous step. If the previous command output listed an `ari` ID, include that in the following command with `--ramdisk-id ari_id`.

```
[ec2-user bundle]$ aws ec2 register-image --region us-west-2 --  
name your_new_ami_name --block-device-mappings DeviceName=device-name,Ebs={SnapshotId=snapshot_id} --virtualization-type paravirtual --architecture x86_64 --kernel-id aki-fc8f11cc --root-device-name device-name
```

15. (Optional) After you have tested that you can launch an instance from your new AMI, you can delete the Amazon EBS volume that you created for this procedure.

```
aws ec2 delete-volume --volume-id volume_id
```

AMI tools reference

You can use the AMI tools commands to create and manage instance store-backed Linux AMIs. To set up the tools, see [Setting up the AMI tools \(p. 128\)](#).

For information about your access keys, see [Best Practices for Managing AWS Access Keys](#).

Commands

- [ec2-ami-tools-version \(p. 141\)](#)
- [ec2-bundle-image \(p. 142\)](#)
- [ec2-bundle-vol \(p. 144\)](#)
- [ec2-delete-bundle \(p. 148\)](#)
- [ec2-download-bundle \(p. 150\)](#)
- [ec2-migrate-manifest \(p. 152\)](#)
- [ec2-unbundle \(p. 153\)](#)
- [ec2-upload-bundle \(p. 154\)](#)
- [Common options for AMI tools \(p. 157\)](#)

ec2-ami-tools-version

Description

Describes the version of the AMI tools.

Syntax

```
ec2-ami-tools-version
```

Output

The version information.

Example

This example command displays the version information for the AMI tools that you're using.

```
[ec2-user ~]$ ec2-ami-tools-version  
1.5.2 20071010
```

ec2-bundle-image

Description

Creates an instance store-backed Linux AMI from an operating system image created in a loopback file.

Syntax

```
ec2-bundle-image -c path -k path -u account -i path [-d path] [--ec2cert path]  
[-r architecture] [--productcodes code1,code2,...] [-B mapping] [-p prefix]
```

Options

-c, --cert *path*

The user's PEM encoded RSA public key certificate file.

Required: Yes

-k, --privatekey *path*

The path to a PEM-encoded RSA key file. You'll need to specify this key to unbundle this bundle, so keep it in a safe place. Note that the key doesn't have to be registered to your AWS account.

Required: Yes

-u, --user *account*

The user's AWS account ID, without dashes.

Required: Yes

-i, --image *path*

The path to the image to bundle.

Required: Yes

-d, --destination *path*

The directory in which to create the bundle.

Default: /tmp

Required: No

--ec2cert *path*

The path to the Amazon EC2 X.509 public key certificate used to encrypt the image manifest.

The us-gov-west-1 and cn-north-1 Regions use a non-default public key certificate and the path to that certificate must be specified with this option. The path to the certificate varies based on the installation method of the AMI tools. For Amazon Linux, the certificates are located at /opt/aws/amitools/ec2/etc/ec2/amitools/. If you installed the AMI tools from the RPM or ZIP file in [Setting up the AMI tools \(p. 128\)](#), the certificates are located at \$EC2_AMITOOL_HOME/etc/ec2/amitools/.

Required: Only for the us-gov-west-1 and cn-north-1 Regions.

-r, --arch *architecture*

Image architecture. If you don't provide the architecture on the command line, you'll be prompted for it when bundling starts.

Valid values: i386 | x86_64

Required: No

--productcodes *code1,code2,...*

Product codes to attach to the image at registration time, separated by commas.

Required: No

-B, --block-device-mapping *mapping*

Defines how block devices are exposed to an instance of this AMI if its instance type supports the specified device.

Specify a comma-separated list of key-value pairs, where each key is a virtual name and each value is the corresponding device name. Virtual names include the following:

- **ami**—The root file system device, as seen by the instance
- **root**—The root file system device, as seen by the kernel
- **swap**—The swap device, as seen by the instance
- **ephemeralN**—The Nth instance store volume

Required: No

-p, --prefix *prefix*

The filename prefix for bundled AMI files.

Default: The name of the image file. For example, if the image path is /var/spool/my-image/version-2/debian.img, then the default prefix is debian.img.

Required: No

--kernel *kernel_id*

Deprecated. Use [register-image](#) to set the kernel.

Required: No

--ramdisk *ramdisk_id*

Deprecated. Use [register-image](#) to set the RAM disk if required.

Required: No

Output

Status messages describing the stages and status of the bundling process.

Example

This example creates a bundled AMI from an operating system image that was created in a loopback file.

```
[ec2-user ~]$ ec2-bundle-image -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -i image.img -d bundled/ -r x86_64
Please specify a value for arch [i386]:
Bundling image file...
Splitting bundled/image.gz.crypt...
Created image.part.00
Created image.part.01
Created image.part.02
Created image.part.03
Created image.part.04
Created image.part.05
```

```
Created image.part.06
Created image.part.07
Created image.part.08
Created image.part.09
Created image.part.10
Created image.part.11
Created image.part.12
Created image.part.13
Created image.part.14
Generating digests for each part...
Digests generated.
Creating bundle manifest...
ec2-bundle-image complete.
```

ec2-bundle-vol

Description

Creates an instance store-backed Linux AMI by compressing, encrypting, and signing a copy of the root device volume for the instance.

Amazon EC2 attempts to inherit product codes, kernel settings, RAM disk settings, and block device mappings from the instance.

By default, the bundle process excludes files that might contain sensitive information. These files include *.sw, *.swo, *.swp, *.pem, *.priv, *id_rsa*, *id_dsa*, *.gpg, *.jks, */.ssh/authorized_keys, and */.bash_history. To include all of these files, use the --no-filter option. To include some of these files, use the --include option.

For more information, see [Creating an instance store-backed Linux AMI \(p. 127\)](#).

Syntax

```
ec2-bundle-vol -c path -k path -u account [-d path] [--ec2cert path] [-r architecture] [--productcodes code1,code2,...] [-B mapping] [--all] [-e directory1,directory2,...] [-i file1,file2,...] [--no-filter] [-p prefix] [-s size] [--[no-]inherit] [-v volume] [-P type] [-S script] [--fstab path] [--generate-fstab] [--grub-config path]
```

Options

-c, --cert *path*

The user's PEM encoded RSA public key certificate file.

Required: Yes

-k, --privatekey *path*

The path to the user's PEM-encoded RSA key file.

Required: Yes

-u, --user *account*

The user's AWS account ID, without dashes.

Required: Yes

-d, --destination *destination*

The directory in which to create the bundle.

Default: /tmp

Required: No

--ec2cert *path*

The path to the Amazon EC2 X.509 public key certificate used to encrypt the image manifest.

The us-gov-west-1 and cn-north-1 Regions use a non-default public key certificate and the path to that certificate must be specified with this option. The path to the certificate varies based on the installation method of the AMI tools. For Amazon Linux, the certificates are located at /opt/aws/amitools/ec2/etc/ec2/amitools/. If you installed the AMI tools from the RPM or ZIP file in [Setting up the AMI tools \(p. 128\)](#), the certificates are located at \$EC2_AMITOOL_HOME/etc/ec2/amitools/.

Required: Only for the us-gov-west-1 and cn-north-1 Regions.

-r, --arch *architecture*

The image architecture. If you don't provide this on the command line, you'll be prompted to provide it when the bundling starts.

Valid values: i386 | x86_64

Required: No

--productcodes *code1,code2,...*

Product codes to attach to the image at registration time, separated by commas.

Required: No

-B, --block-device-mapping *mapping*

Defines how block devices are exposed to an instance of this AMI if its instance type supports the specified device.

Specify a comma-separated list of key-value pairs, where each key is a virtual name and each value is the corresponding device name. Virtual names include the following:

- ami—The root file system device, as seen by the instance
- root—The root file system device, as seen by the kernel
- swap—The swap device, as seen by the instance
- ephemeralN—The Nth instance store volume

Required: No

-a, --all

Bundle all directories, including those on remotely mounted file systems.

Required: No

-e, --exclude *directory1,directory2,...*

A list of absolute directory paths and files to exclude from the bundle operation. This parameter overrides the --all option. When exclude is specified, the directories and subdirectories listed with the parameter will not be bundled with the volume.

Required: No

-i, --include *file1,file2,...*

A list of files to include in the bundle operation. The specified files would otherwise be excluded from the AMI because they might contain sensitive information.

Required: No

--no-filter

If specified, we won't exclude files from the AMI because they might contain sensitive information.

Required: No

-p, --prefix *prefix*

The file name prefix for bundled AMI files.

Default: `image`

Required: No

-s, --size *size*

The size, in MB (1024 * 1024 bytes), of the image file to create. The maximum size is 10240 MB.

Default: 10240

Required: No

--[no-]inherit

Indicates whether the image should inherit the instance's metadata (the default is to inherit). Bundling fails if you enable `--inherit` but the instance metadata is not accessible.

Required: No

-v, --volume *volume*

The absolute path to the mounted volume from which to create the bundle.

Default: The root directory (/)

Required: No

-P, --partition *type*

Indicates whether the disk image should use a partition table. If you don't specify a partition table type, the default is the type used on the parent block device of the volume, if applicable, otherwise the default is gpt.

Valid values: `mbr` | `gpt` | `none`

Required: No

-S, --script *script*

A customization script to be run right before bundling. The script must expect a single argument, the mount point of the volume.

Required: No

--fstab *path*

The path to the fstab to bundle into the image. If this is not specified, Amazon EC2 bundles /etc/fstab.

Required: No

--generate-fstab

Bundles the volume using an Amazon EC2-provided fstab.

Required: No

--grub-config

The path to an alternate grub configuration file to bundle into the image. By default, `ec2-bundle-vol` expects either `/boot/grub/menu.lst` or `/boot/grub/grub.conf` to exist on the cloned image. This option allows you to specify a path to an alternative grub configuration file, which will then be copied over the defaults (if present).

Required: No

--kernel *kernel_id*

Deprecated. Use [register-image](#) to set the kernel.

Required: No

--ramdisk *ramdisk_id*

Deprecated. Use [register-image](#) to set the RAM disk if required.

Required: No

Output

Status messages describing the stages and status of the bundling.

Example

This example creates a bundled AMI by compressing, encrypting and signing a snapshot of the local machine's root file system.

```
[ec2-user ~]$ ec2-bundle-vol -d /mnt -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -r x86_64
Copying / into the image file /mnt/image...
Excluding:
  sys
  dev/shm
  proc
  dev/pts
  proc/sys/fs/binfmt_misc
  dev
  media
  mnt
  proc
  sys
  tmp/image
  mnt/img-mnt
1+0 records in
1+0 records out
mke2fs 1.38 (30-Jun-2005)
warning: 256 blocks unused.

Splitting /mnt/image.gz.crypt...
Created image.part.00
Created image.part.01
Created image.part.02
Created image.part.03
...
Created image.part.22
Created image.part.23
Generating digests for each part...
Digests generated.
Creating bundle manifest...
```

Bundle Volume complete.

ec2-delete-bundle

Description

Deletes the specified bundle from Amazon S3 storage. After you delete a bundle, you can't launch instances from the corresponding AMI.

Syntax

```
ec2-delete-bundle -b bucket -a access_key_id -s secret_access_key [-t token]  
[--url url] [--region region] [--sigv version] [-m path] [-p prefix] [--clear]  
[--retry] [-y]
```

Options

-b, --bucket *bucket*

The name of the Amazon S3 bucket containing the bundled AMI, followed by an optional '/'-delimited path prefix

Required: Yes

-a, --access-key *access_key_id*

The AWS access key ID.

Required: Yes

-s, --secret-key *secret_access_key*

The AWS secret access key.

Required: Yes

-t, --delegation-token *token*

The delegation token to pass along to the AWS request. For more information, see the [Using Temporary Security Credentials](#).

Required: Only when you are using temporary security credentials.

Default: The value of the `AWS_DELEGATION_TOKEN` environment variable (if set).

--region *region*

The Region to use in the request signature.

Default: `us-east-1`

Required: Required if using signature version 4

--sigv *version*

The signature version to use when signing the request.

Valid values: 2 | 4

Default: 4

Required: No

-m, --manifestpath

The path to the manifest file.

Required: You must specify **--prefix** or **--manifest**.

-p, --prefix prefix

The bundled AMI filename prefix. Provide the entire prefix. For example, if the prefix is image.img, use **-p image.img** and not **-p image**.

Required: You must specify **--prefix** or **--manifest**.

--clear

Deletes the Amazon S3 bucket if it's empty after deleting the specified bundle.

Required: No

--retry

Automatically retries on all Amazon S3 errors, up to five times per operation.

Required: No

-y, --yes

Automatically assumes the answer to all prompts is yes.

Required: No

Output

Amazon EC2 displays status messages indicating the stages and status of the delete process.

Example

This example deletes a bundle from Amazon S3.

```
[ec2-user ~]$ ec2-delete-bundle -b DOC-EXAMPLE-BUCKET1 -a your_access_key_id -  
s your_secret_access_key  
Deleting files:  
DOC-EXAMPLE-BUCKET1/  
image.manifest.xml  
DOC-EXAMPLE-BUCKET1/  
image.part.00  
DOC-EXAMPLE-BUCKET1/  
image.part.01  
DOC-EXAMPLE-BUCKET1/  
image.part.02  
DOC-EXAMPLE-BUCKET1/  
image.part.03  
DOC-EXAMPLE-BUCKET1/  
image.part.04  
DOC-EXAMPLE-BUCKET1/  
image.part.05  
DOC-EXAMPLE-BUCKET1/image.part.06  
Continue? [y/n]  
y  
Deleted DOC-EXAMPLE-BUCKET1/image.manifest.xml  
Deleted DOC-EXAMPLE-BUCKET1/image.part.00  
Deleted DOC-EXAMPLE-BUCKET1/image.part.01  
Deleted DOC-EXAMPLE-BUCKET1/image.part.02  
Deleted DOC-EXAMPLE-BUCKET1/image.part.03
```

```
Deleted DOC-EXAMPLE-BUCKET1/image.part.04
Deleted DOC-EXAMPLE-BUCKET1/image.part.05
Deleted DOC-EXAMPLE-BUCKET1/image.part.06
ec2-delete-bundle complete.
```

ec2-download-bundle

Description

Downloads the specified instance store-backed Linux AMIs from Amazon S3 storage.

Syntax

```
ec2-download-bundle -b bucket -a access_key_id -s secret_access_key -k path
[--url url] [--region region] [--sigv version] [-m file] [-p prefix] [-d directory] [--retry]
```

Options

-b, --bucket *bucket*

The name of the Amazon S3 bucket where the bundle is located, followed by an optional '/'-delimited path prefix.

Required: Yes

-a, --access-key *access_key_id*

The AWS access key ID.

Required: Yes

-s, --secret-key *secret_access_key*

The AWS secret access key.

Required: Yes

-k, --privatekey *path*

The private key used to decrypt the manifest.

Required: Yes

--url *url*

The Amazon S3 service URL.

Default: <https://s3.amazonaws.com/>

Required: No

--region *region*

The Region to use in the request signature.

Default: us-east-1

Required: Required if using signature version 4

--sigv *version*

The signature version to use when signing the request.

Valid values: 2 | 4

Default: 4

Required: No

`-m, --manifest file`

The name of the manifest file (without the path). We recommend that you specify either the manifest (`-m`) or a prefix (`-p`).

Required: No

`-p, --prefix prefix`

The filename prefix for the bundled AMI files.

Default: image

Required: No

`-d, --directory directory`

The directory where the downloaded bundle is saved. The directory must exist.

Default: The current working directory.

Required: No

`--retry`

Automatically retries on all Amazon S3 errors, up to five times per operation.

Required: No

Output

Status messages indicating the various stages of the download process are displayed.

Example

This example creates the bundled directory (using the Linux `mkdir` command) and downloads the bundle from the `DOC-EXAMPLE-BUCKET1` Amazon S3 bucket.

```
[ec2-user ~]$ mkdir bundled
[ec2-user ~]$ ec2-download-bundle -b DOC-EXAMPLE-BUCKET1/bundles/bundle_name
-m image.manifest.xml -a your_access_key_id -s your_secret_access_key -k pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d mybundle
Downloading manifest image.manifest.xml from DOC-EXAMPLE-BUCKET1 to mybundle/
image.manifest.xml ...
Downloading part image.part.00 from DOC-EXAMPLE-BUCKET1/bundles/bundle_name to mybundle/
image.part.00 ...
Downloaded image.part.00 from DOC-EXAMPLE-BUCKET1
Downloading part image.part.01 from DOC-EXAMPLE-BUCKET1/bundles/bundle_name to mybundle/
image.part.01 ...
Downloaded image.part.01 from DOC-EXAMPLE-BUCKET1
Downloading part image.part.02 from DOC-EXAMPLE-BUCKET1/bundles/bundle_name to mybundle/
image.part.02 ...
Downloaded image.part.02 from DOC-EXAMPLE-BUCKET1
Downloading part image.part.03 from DOC-EXAMPLE-BUCKET1/bundles/bundle_name to mybundle/
image.part.03 ...
Downloaded image.part.03 from DOC-EXAMPLE-BUCKET1
Downloading part image.part.04 from DOC-EXAMPLE-BUCKET1/bundles/bundle_name to mybundle/
image.part.04 ...
Downloaded image.part.04 from DOC-EXAMPLE-BUCKET1
Downloading part image.part.05 from DOC-EXAMPLE-BUCKET1/bundles/bundle_name to mybundle/
image.part.05 ...
```

```
Downloaded image.part.05 from DOC-EXAMPLE-BUCKET1
Downloading part image.part.06 from DOC-EXAMPLE-BUCKET1/bundles/bundle_name to mybundle/
image.part.06 ...
Downloaded image.part.06 from DOC-EXAMPLE-BUCKET1
```

ec2-migrate-manifest

Description

Modifies an instance store-backed Linux AMI (for example, its certificate, kernel, and RAM disk) so that it supports a different Region.

Syntax

```
ec2-migrate-manifest -c path -k path -m path {(-a access_key_id -s secret_access_key --region region) | (--no-mapping)} [--ec2cert ec2_cert_path] [--kernel kernel-id] [--ramdisk ramdisk_id]
```

Options

-c, --cert path

The user's PEM encoded RSA public key certificate file.

Required: Yes

-k, --privatekey path

The path to the user's PEM-encoded RSA key file.

Required: Yes

--manifest path

The path to the manifest file.

Required: Yes

-a, --access-key access_key_id

The AWS access key ID.

Required: Required if using automatic mapping.

-s, --secret-key secret_access_key

The AWS secret access key.

Required: Required if using automatic mapping.

--region region

The Region to look up in the mapping file.

Required: Required if using automatic mapping.

--no-mapping

Disables automatic mapping of kernels and RAM disks.

During migration, Amazon EC2 replaces the kernel and RAM disk in the manifest file with a kernel and RAM disk designed for the destination region. Unless the **--no-mapping** parameter is given, **ec2-migrate-bundle** might use the **DescribeRegions** and **DescribeImages** operations to perform automated mappings.

Required: Required if you're not providing the `-a`, `-s`, and `--region` options used for automatic mapping.

`--ec2cert path`

The path to the Amazon EC2 X.509 public key certificate used to encrypt the image manifest.

The `us-gov-west-1` and `cn-north-1` Regions use a non-default public key certificate and the path to that certificate must be specified with this option. The path to the certificate varies based on the installation method of the AMI tools. For Amazon Linux, the certificates are located at `/opt/aws/amitools/ec2/etc/ec2/amitools/`. If you installed the AMI tools from the ZIP file in [Setting up the AMI tools \(p. 128\)](#), the certificates are located at `$EC2_AMITOOL_HOME/etc/ec2/amitools/`.

Required: Only for the `us-gov-west-1` and `cn-north-1` Regions.

`--kernel kernel_id`

The ID of the kernel to select.

Important

We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see [Enabling Your Own Linux Kernels \(p. 193\)](#).

Required: No

`--ramdisk ramdisk_id`

The ID of the RAM disk to select.

Important

We recommend that you use PV-GRUB instead of kernels and RAM disks. For more information, see [Enabling Your Own Linux Kernels \(p. 193\)](#).

Required: No

Output

Status messages describing the stages and status of the bundling process.

Example

This example copies the AMI specified in the `my-ami.manifest.xml` manifest from the US to the EU.

```
[ec2-user ~]$ ec2-migrate-manifest --manifest my-ami.manifest.xml --cert cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CL0.pem --privatekey pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CL0.pem --region eu-west-1

Backing up manifest...
Successfully migrated my-ami.manifest.xml It is now suitable for use in eu-west-1.
```

ec2-unbundle

Description

Re-creates the bundle from an instance store-backed Linux AMI.

Syntax

```
ec2-unbundle -k path -m path [-s source_directory] [-d destination_directory]
```

Options

-k, --privatekey *path*

The path to your PEM-encoded RSA key file.

Required: Yes

-m, --manifest *path*

The path to the manifest file.

Required: Yes

-s, --source *source_directory*

The directory containing the bundle.

Default: The current directory.

Required: No

-d, --destination *destination_directory*

The directory in which to unbundle the AMI. The destination directory must exist.

Default: The current directory.

Required: No

Example

This Linux and UNIX example unbundles the AMI specified in the `image.manifest.xml` file.

```
[ec2-user ~]$ mkdir unbundled
$ ec2-unbundle -m mybundle/image.manifest.xml -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -s
mybundle -d unbundled
$ ls -l unbundled
total 1025008
-rw-r--r-- 1 root root 1048578048 Aug 25 23:46 image.img
```

Output

Status messages indicating the various stages of the unbundling process are displayed.

ec2-upload-bundle

Description

Uploads the bundle for an instance store-backed Linux AMI to Amazon S3 and sets the appropriate ACLs on the uploaded objects. For more information, see [Creating an instance store-backed Linux AMI \(p. 127\)](#).

Syntax

```
ec2-upload-bundle -b bucket -a access_key_id -s secret_access_key [-t token] -m
path [--url url] [--region region] [--sigv version] [--acl acl] [-d directory]
[--part part] [--retry] [--skipmanifest]
```

Options

-b, --bucket *bucket*

The name of the Amazon S3 bucket in which to store the bundle, followed by an optional '/'-delimited path prefix. If the bucket doesn't exist, it's created if the bucket name is available.

Required: Yes

-a, --access-key *access_key_id*

Your AWS access key ID.

Required: Yes

-s, --secret-key *secret_access_key*

Your AWS secret access key.

Required: Yes

-t, --delegation-token *token*

The delegation token to pass along to the AWS request. For more information, see the [Using Temporary Security Credentials](#).

Required: Only when you are using temporary security credentials.

Default: The value of the `AWS_DELEGATION_TOKEN` environment variable (if set).

-m, --manifest *path*

The path to the manifest file. The manifest file is created during the bundling process and can be found in the directory containing the bundle.

Required: Yes

--url *url*

Deprecated. Use the `--region` option instead unless your bucket is constrained to the `EU` location (and not `eu-west-1`). The `--location` flag is the only way to target that specific location restraint.

The Amazon S3 endpoint service URL.

Default: `https://s3.amazonaws.com/`

Required: No

--region *region*

The Region to use in the request signature for the destination S3 bucket.

- If the bucket doesn't exist and you don't specify a Region, the tool creates the bucket without a location constraint (in `us-east-1`).
- If the bucket doesn't exist and you specify a Region, the tool creates the bucket in the specified Region.
- If the bucket exists and you don't specify a Region, the tool uses the bucket's location.
- If the bucket exists and you specify `us-east-1` as the Region, the tool uses the bucket's actual location without any error message, any existing matching files are over-written.
- If the bucket exists and you specify a Region (other than `us-east-1`) that doesn't match the bucket's actual location, the tool exits with an error.

If your bucket is constrained to the `EU` location (and not `eu-west-1`), use the `--location` flag instead. The `--location` flag is the only way to target that specific location restraint.

Default: us-east-1

Required: Required if using signature version 4

--sigv *version*

The signature version to use when signing the request.

Valid values: 2 | 4

Default: 4

Required: No

--acl *acl*

The access control list policy of the bundled image.

Valid values: public-read | aws-exec-read

Default: aws-exec-read

Required: No

-d, --directory *directory*

The directory containing the bundled AMI parts.

Default: The directory containing the manifest file (see the -m option).

Required: No

--part *part*

Starts uploading the specified part and all subsequent parts. For example, --part 04.

Required: No

--retry

Automatically retries on all Amazon S3 errors, up to five times per operation.

Required: No

--skipmanifest

Does not upload the manifest.

Required: No

--location *location*

Deprecated. Use the --region option instead, unless your bucket is constrained to the EU location (and not eu-west-1). The --location flag is the only way to target that specific location restraint.

The location constraint of the destination Amazon S3 bucket. If the bucket exists and you specify a location that doesn't match the bucket's actual location, the tool exits with an error. If the bucket exists and you don't specify a location, the tool uses the bucket's location. If the bucket doesn't exist and you specify a location, the tool creates the bucket in the specified location. If the bucket doesn't exist and you don't specify a location, the tool creates the bucket without a location constraint (in us-east-1).

Default: If --region is specified, the location is set to that specified Region. If --region is not specified, the location defaults to us-east-1.

Required: No

Output

Amazon EC2 displays status messages that indicate the stages and status of the upload process.

Example

This example uploads the bundle specified by the `image.manifest.xml` manifest.

```
[ec2-user ~]$ ec2-upload-bundle -b DOC-EXAMPLE-BUCKET1/bundles/bundle_name -m image.manifest.xml -a your_access_key_id -s your_secret_access_key
Creating bucket...
Uploading bundled image parts to the S3 bucket DOC-EXAMPLE-BUCKET1 ...
Uploaded image.part.00
Uploaded image.part.01
Uploaded image.part.02
Uploaded image.part.03
Uploaded image.part.04
Uploaded image.part.05
Uploaded image.part.06
Uploaded image.part.07
Uploaded image.part.08
Uploaded image.part.09
Uploaded image.part.10
Uploaded image.part.11
Uploaded image.part.12
Uploaded image.part.13
Uploaded image.part.14
Uploading manifest ...
Uploaded manifest.
Bundle upload completed.
```

Common options for AMI tools

Most of the AMI tools accept the following optional parameters.

`--help, -h`

Displays the help message.

`--version`

Displays the version and copyright notice.

`--manual`

Displays the manual entry.

`--batch`

Runs in batch mode, suppressing interactive prompts.

`--debug`

Displays information that can be useful when troubleshooting problems.

Using encryption with EBS-backed AMIs

AMIs that are backed by Amazon EBS snapshots can take advantage of Amazon EBS encryption. Snapshots of both data and root volumes can be encrypted and attached to an AMI. You can launch instances and copy images with full EBS encryption support included. Encryption parameters for these operations are supported in all Regions where AWS KMS is available.

EC2 instances with encrypted EBS volumes are launched from AMIs in the same way as other instances. In addition, when you launch an instance from an AMI backed by unencrypted EBS snapshots, you can encrypt some or all of the volumes during launch.

Like EBS volumes, snapshots in AMIs can be encrypted by either your default AWS Key Management Service customer master key (CMK), or to a customer managed key that you specify. You must in all cases have permission to use the selected key.

AMIs with encrypted snapshots can be shared across AWS accounts. For more information, see [Shared AMIs](#).

Encryption with EBS-backed AMIs topics

- [Instance-launching scenarios \(p. 158\)](#)
- [Image-copying scenarios \(p. 161\)](#)

Instance-launching scenarios

Amazon EC2 instances are launched from AMIs using the `RunInstances` action with parameters supplied through block device mapping, either by means of the AWS Management Console or directly using the Amazon EC2 API or CLI. For more information about block device mapping, see [Block device mapping](#). For examples of controlling block device mapping from the AWS CLI, see [Launch, List, and Terminate EC2 Instances](#).

By default, without explicit encryption parameters, a `RunInstances` action maintains the existing encryption state of an AMI's source snapshots while restoring EBS volumes from them. If [Encryption by default \(p. 1124\)](#) is enabled, all volumes created from the AMI (whether from encrypted or unencrypted snapshots) will be encrypted. If encryption by default is not enabled, then the instance maintains the encryption state of the AMI.

You can also launch an instance and simultaneously apply a new encryption state to the resulting volumes by supplying encryption parameters. Consequently, the following behaviors are observed:

Launch with no encryption parameters

- An unencrypted snapshot is restored to an unencrypted volume, unless encryption by default is enabled, in which case all the newly created volumes will be encrypted.
- An encrypted snapshot that you own is restored to a volume that is encrypted to the same CMK.
- An encrypted snapshot that you do not own (for example, the AMI is shared with you) is restored to a volume that is encrypted by your AWS account's default CMK.

The default behaviors can be overridden by supplying encryption parameters. The available parameters are `Encrypted` and `KmsKeyId`. Setting only the `Encrypted` parameter results in the following:

Instance launch behaviors with `Encrypted` set, but no `KmsKeyId` specified

- An unencrypted snapshot is restored to an EBS volume that is encrypted by your AWS account's default CMK.
- An encrypted snapshot that you own is restored to an EBS volume encrypted by the same CMK. (In other words, the `Encrypted` parameter has no effect.)
- An encrypted snapshot that you do not own (i.e., the AMI is shared with you) is restored to a volume that is encrypted by your AWS account's default CMK. (In other words, the `Encrypted` parameter has no effect.)

Setting both the `Encrypted` and `KmsKeyId` parameters allows you to specify a non-default CMK for an encryption operation. The following behaviors result:

Instance with both `Encrypted` and `KmsKeyId` set

- An unencrypted snapshot is restored to an EBS volume encrypted by the specified CMK.
- An encrypted snapshot is restored to an EBS volume encrypted not to the original CMK, but instead to the specified CMK.

Submitting a `KmsKeyId` without also setting the `Encrypted` parameter results in an error.

The following sections provide examples of launching instances from AMIs using non-default encryption parameters. In each of these scenarios, parameters supplied to the `RunInstances` action result in a change of encryption state during restoration of a volume from a snapshot.

Note

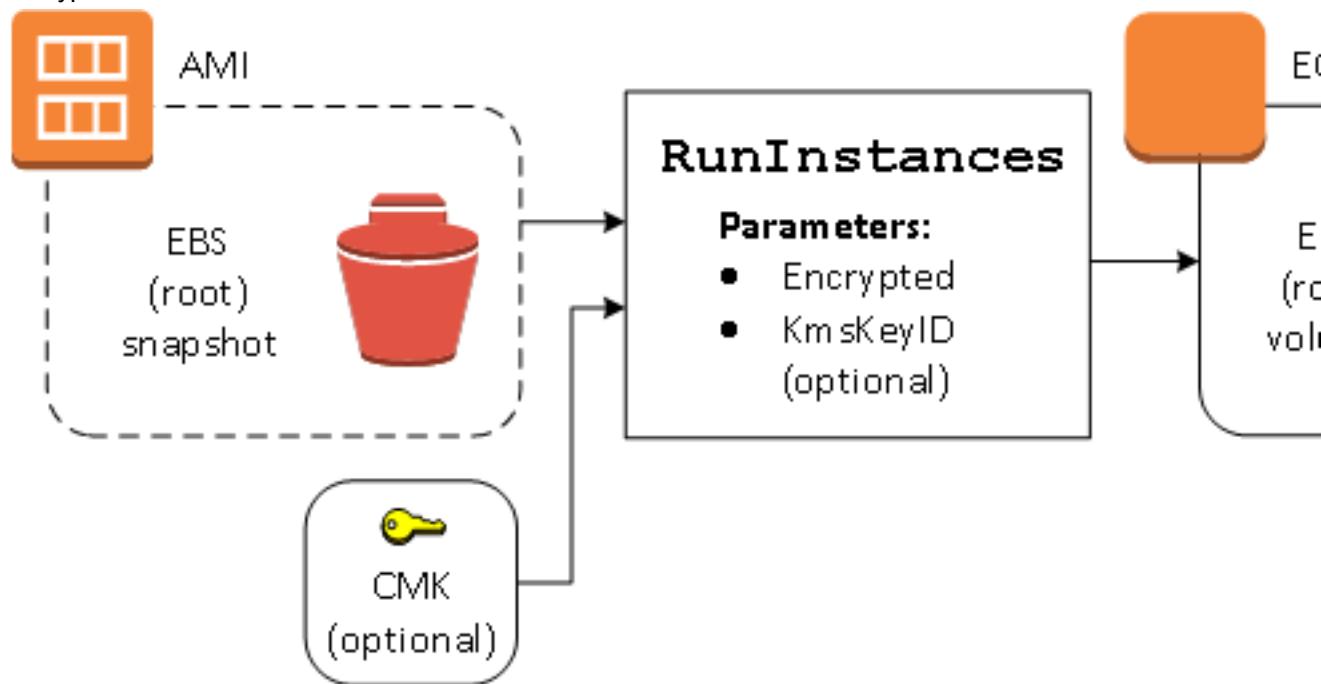
For detailed console procedures to launch an instance from an AMI, see [Launch Your Instance](#).

For documentation of the `RunInstances` API, see [RunInstances](#).

For documentation of the `run-instances` command in the AWS Command Line Interface, see [run-instances](#).

Encrypt a volume during launch

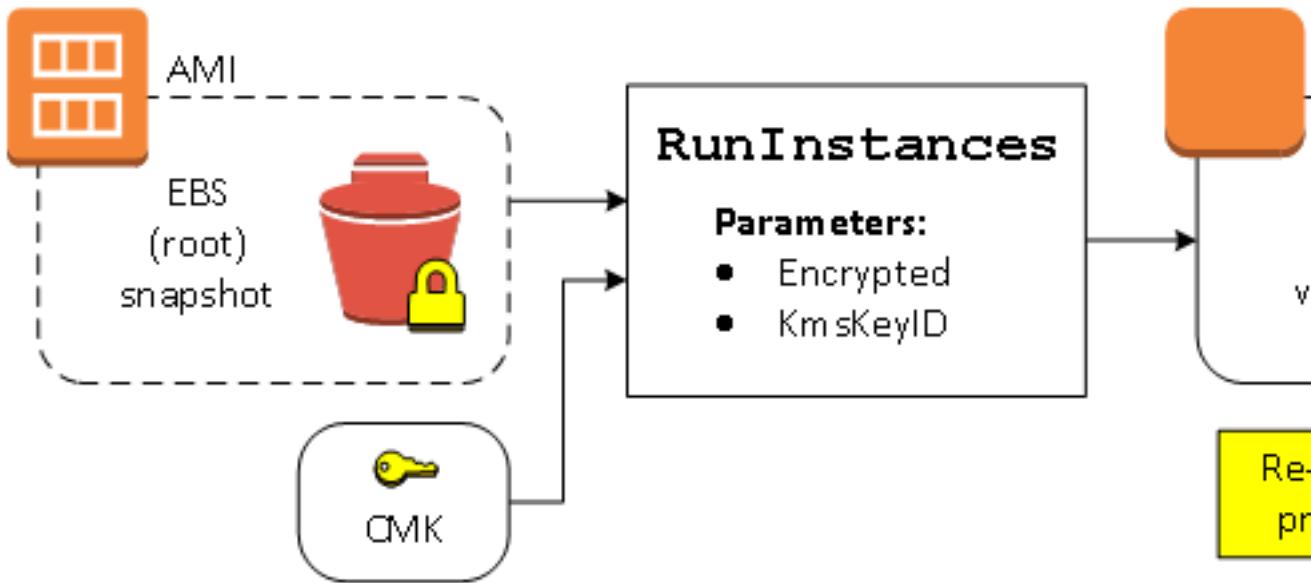
In this example, an AMI backed by an unencrypted snapshot is used to launch an EC2 instance with an encrypted EBS volume.



The `Encrypted` parameter alone results in the volume for this instance being encrypted. Providing a `KmsKeyId` parameter is optional. If no key ID is specified, the AWS account's default CMK is used to encrypt the volume. To encrypt the volume to a different CMK that you own, supply the `KmsKeyId` parameter.

Re-encrypt a volume during launch

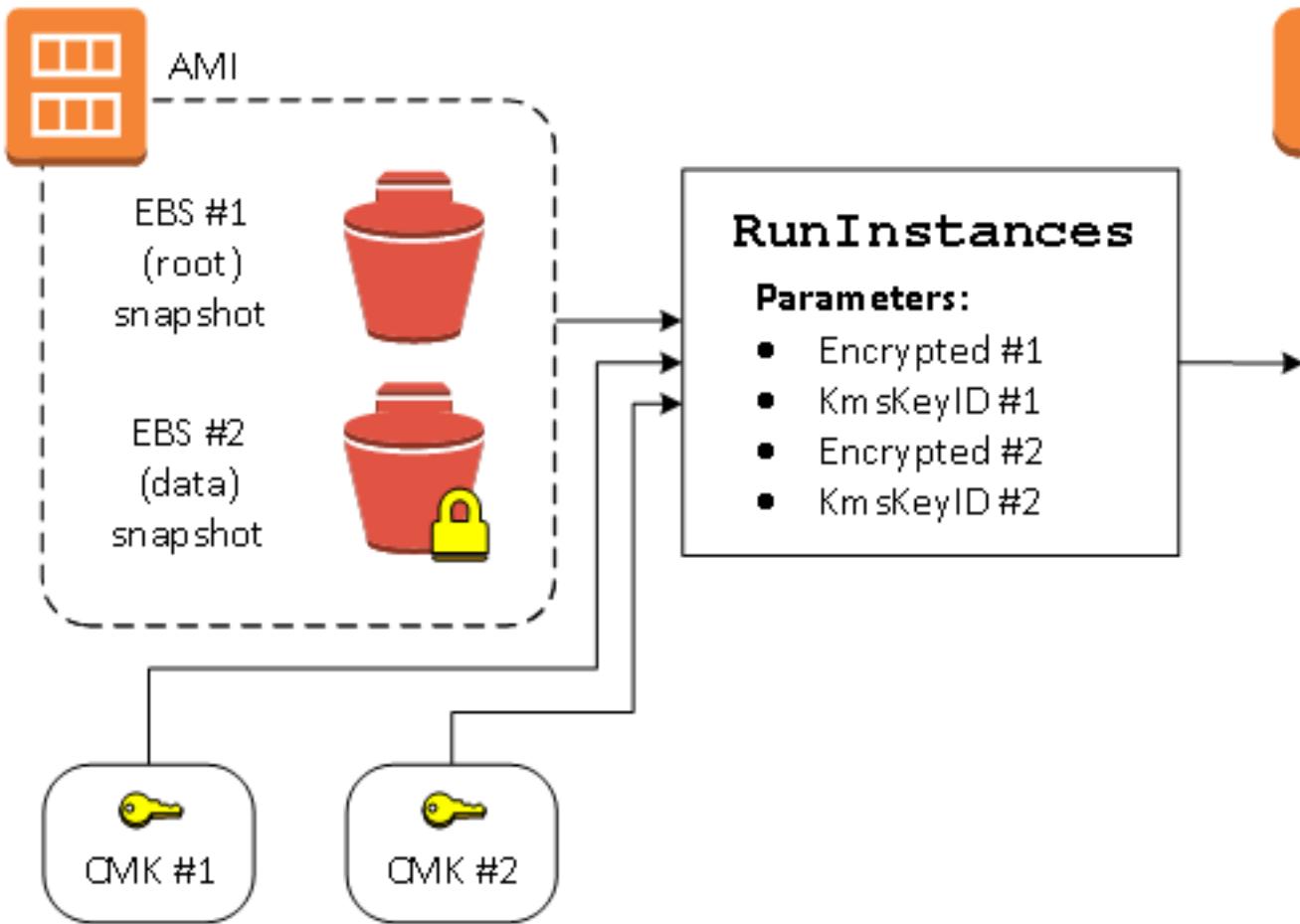
In this example, an AMI backed by an encrypted snapshot is used to launch an EC2 instance with an EBS volume encrypted by a new CMK.



If you own the AMI and supply no encryption parameters, the resulting instance has a volume encrypted by the same key as the snapshot. If the AMI is shared rather than owned by you, and you supply no encryption parameters, the volume is encrypted by your default CMK. With encryption parameters supplied as shown, the volume is encrypted by the specified CMK.

Change encryption state of multiple volumes during launch

In this more complex example, an AMI backed by multiple snapshots (each with its own encryption state) is used to launch an EC2 instance with a newly encrypted volume and a re-encrypted volume.



In this scenario, the `RunInstances` action is supplied with encryption parameters for each of the source snapshots. When all possible encryption parameters are specified, the resulting instance is the same regardless of whether you own the AMI.

Image-copying scenarios

Amazon EC2 AMIs are copied using the `CopyImage` action, either through the AWS Management Console or directly using the Amazon EC2 API or CLI.

By default, without explicit encryption parameters, a `CopyImage` action maintains the existing encryption state of an AMI's source snapshots during copy. You can also copy an AMI and simultaneously apply a new encryption state to its associated EBS snapshots by supplying encryption parameters. Consequently, the following behaviors are observed:

Copy with no encryption parameters

- An unencrypted snapshot is copied to another unencrypted snapshot, unless encryption by default is enabled, in which case all the newly created snapshots will be encrypted.
- An encrypted snapshot that you own is copied to a snapshot encrypted with the same key.
- An encrypted snapshot that you do not own (that is, the AMI is shared with you) is copied to a snapshot that is encrypted by your AWS account's default CMK.

All of these default behaviors can be overridden by supplying encryption parameters. The available parameters are `Encrypted` and `KmsKeyId`. Setting only the `Encrypted` parameter results in the following:

Copy-image behaviors with `Encrypted` set, but no `KmsKeyId` specified

- An unencrypted snapshot is copied to a snapshot encrypted by the AWS account's default CMK.
- An encrypted snapshot is copied to a snapshot encrypted by the same CMK. (In other words, the `Encrypted` parameter has no effect.)
- An encrypted snapshot that you do not own (i.e., the AMI is shared with you) is copied to a volume that is encrypted by your AWS account's default CMK. (In other words, the `Encrypted` parameter has no effect.)

Setting both the `Encrypted` and `KmsKeyId` parameters allows you to specify a customer managed CMK for an encryption operation. The following behaviors result:

Copy-image behaviors with both `Encrypted` and `KmsKeyId` set

- An unencrypted snapshot is copied to a snapshot encrypted by the specified CMK.
- An encrypted snapshot is copied to a snapshot encrypted not to the original CMK, but instead to the specified CMK.

Submitting a `KmsKeyId` without also setting the `Encrypted` parameter results in an error.

The following section provides an example of copying an AMI using non-default encryption parameters, resulting in a change of encryption state.

Note

For detailed console procedures to copy an AMI, see [Copying an AMI](#).

For documentation of the `CopyImage` API, see [CopyImage](#).

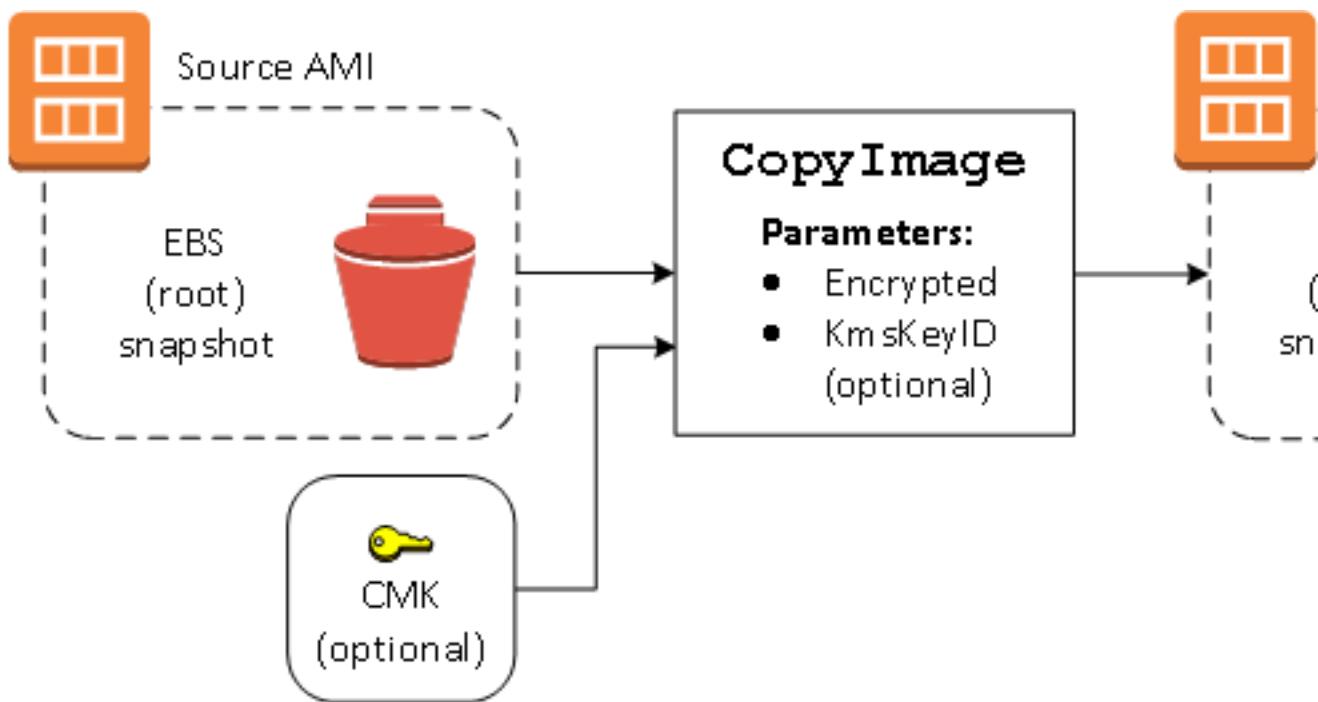
For documentation of the command `copy-image` in the AWS Command Line Interface, see [copy-image](#).

Encrypt an unencrypted image during copy

In this scenario, an AMI backed by an unencrypted root snapshot is copied to an AMI with an encrypted root snapshot. The `CopyImage` action is invoked with two encryption parameters, including a CMK. As a result, the encryption status of the root snapshot changes, so that the target AMI is backed by a root snapshot containing the same data as the source snapshot, but encrypted using the specified key. You incur storage costs for the snapshots in both AMIs, as well as charges for any instances you launch from either AMI.

Note

Enabling [encryption by default \(p. 1124\)](#) has the same effect as setting the `Encrypted` parameter to `true` for all snapshots in the AMI.



Setting the `Encrypted` parameter encrypts the single snapshot for this instance. If you do not specify the `KmsKeyId` parameter, the default CMK is used to encrypt the snapshot copy.

Note

You can also copy an image with multiple snapshots and configure the encryption state of each individually.

Copying an AMI

You can copy an Amazon Machine Image (AMI) within or across AWS Regions using the AWS Management Console, the AWS Command Line Interface or SDKs, or the Amazon EC2 API, all of which support the `CopyImage` action. You can copy both Amazon EBS-backed AMIs and instance-store-backed AMIs. You can copy AMIs with encrypted snapshots and also change encryption status during the copy process.

Copying a source AMI results in an identical but distinct target AMI with its own unique identifier. In the case of an Amazon EBS-backed AMI, each of its backing snapshots is, by default, copied to an identical but distinct target snapshot. (The sole exceptions are when you choose to encrypt or re-encrypt the snapshot.) You can change or deregister the source AMI with no effect on the target AMI. The reverse is also true.

There are no charges for copying an AMI. However, standard storage and data transfer rates apply. If you copy an EBS-backed AMI, you will incur charges for the storage of any additional EBS snapshots.

AWS does not copy launch permissions, user-defined tags, or Amazon S3 bucket permissions from the source AMI to the new AMI. After the copy operation is complete, you can apply launch permissions, user-defined tags, and Amazon S3 bucket permissions to the new AMI.

You can't copy an AMI that was obtained from the AWS Marketplace, regardless of whether you obtained it directly or it was shared with you. Instead, launch an EC2 instance using the AWS Marketplace AMI and then create an AMI from the instance. For more information, see [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#).

Topics

- [Permissions for copying an instance store-backed AMI \(p. 164\)](#)
- [Cross-Region copying \(p. 165\)](#)
- [Cross-account copying \(p. 166\)](#)
- [Encryption and copying \(p. 166\)](#)
- [Copying an AMI \(p. 167\)](#)
- [Stopping a pending AMI copy operation \(p. 168\)](#)

Permissions for copying an instance store-backed AMI

If you use an IAM user to copy an instance store-backed AMI, the user must have the following Amazon S3 permissions: `s3:CreateBucket`, `s3:GetBucketAcl`, `s3>ListAllMyBuckets`, `s3:GetObject`, `s3:PutObject`, and `s3:PutObjectAcl`.

The following example policy allows the user to copy the AMI source in the specified bucket to the specified Region.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "s3>ListAllMyBuckets",  
            "Resource": [  
                "arn:aws:s3:::*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": "s3:GetObject",  
            "Resource": [  
                "arn:aws:s3:::ami-source-bucket/*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3>CreateBucket",  
                "s3:GetBucketAcl",  
                "s3:PutObjectAcl",  
                "s3:PutObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::amis-for-123456789012-in-us-east-1*"  
            ]  
        }  
    ]  
}
```

To find the Amazon Resource Name (ARN) of the AMI source bucket, open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>, in the navigation pane choose **AMIs**, and locate the bucket name in the **Source** column.

Note

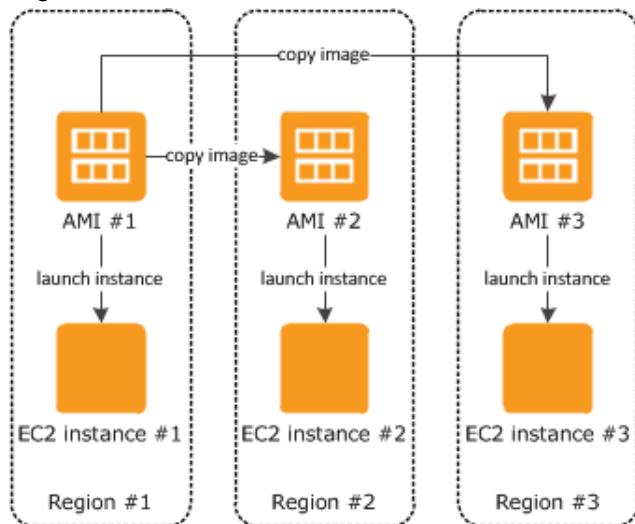
The `s3>CreateBucket` permission is only needed the first time that the IAM user copies an instance store-backed AMI to an individual Region. After that, the Amazon S3 bucket that is already created in the Region is used to store all future AMIs that you copy to that Region.

Cross-Region copying

Copying an AMI across geographically diverse Regions provides the following benefits:

- **Consistent global deployment:** Copying an AMI from one Region to another enables you to launch consistent instances in different Regions based on the same AMI.
- **Scalability:** You can more easily design and build global applications that meet the needs of your users, regardless of their location.
- **Performance:** You can increase performance by distributing your application, as well as locating critical components of your application in closer proximity to your users. You can also take advantage of Region-specific features, such as instance types or other AWS services.
- **High availability:** You can design and deploy applications across AWS Regions, to increase availability.

The following diagram shows the relations among a source AMI and two copied AMIs in different Regions, as well as the EC2 instances launched from each. When you launch an instance from an AMI, it resides in the same Region where the AMI resides. If you make changes to the source AMI and want those changes to be reflected in the AMIs in the target Regions, you must recopy the source AMI to the target Regions.



When you first copy an instance store-backed AMI to a Region, we create an Amazon S3 bucket for the AMIs copied to that Region. All instance store-backed AMIs that you copy to that Region are stored in this bucket. The bucket names have the following format: `amis-for-account-in-region-hash`. For example: `amis-for-123456789012-in-us-east-2-yhjmxvp6`.

Prerequisite

Prior to copying an AMI, you must ensure that the contents of the source AMI are updated to support running in a different Region. For example, you should update any database connection strings or similar application configuration data to point to the appropriate resources. Otherwise, instances launched from the new AMI in the destination Region may still use the resources from the source Region, which can impact performance and cost.

Limits

- Destination Regions are limited to 50 concurrent AMI copies.
- You cannot copy a paravirtual (PV) AMI to a Region that does not support PV AMIs. For more information, see [Linux AMI virtualization types \(p. 102\)](#).

Cross-account copying

You can share an AMI with another AWS account. Sharing an AMI does not affect the ownership of the AMI. The owning account is charged for the storage in the Region. For more information, see [Sharing an AMI with specific AWS accounts \(p. 113\)](#).

If you copy an AMI that has been shared with your account, you are the owner of the target AMI in your account. The owner of the source AMI is charged standard Amazon EBS or Amazon S3 transfer fees, and you are charged for the storage of the target AMI in the destination Region.

Resource Permissions

To copy an AMI that was shared with you from another account, the owner of the source AMI must grant you read permissions for the storage that backs the AMI, either the associated EBS snapshot (for an Amazon EBS-backed AMI) or an associated S3 bucket (for an instance store-backed AMI). If the shared AMI has encrypted snapshots, the owner must share the key or keys with you as well.

Encryption and copying

The following table shows encryption support for various AMI-copying scenarios. While it is possible to copy an unencrypted snapshot to yield an encrypted snapshot, you cannot copy an encrypted snapshot to yield an unencrypted one.

Scenario	Description	Supported
1	Unencrypted-to-unencrypted	Yes
2	Encrypted-to-encrypted	Yes
3	Unencrypted-to-encrypted	Yes
4	Encrypted-to-unencrypted	No

Note

Encrypting during the `CopyImage` action applies only to Amazon EBS-backed AMIs. Because an instance store-backed AMI does not rely on snapshots, you cannot use copying to change its encryption status.

By default (i.e., without specifying encryption parameters), the backing snapshot of an AMI is copied with its original encryption status. Copying an AMI backed by an unencrypted snapshot results in an identical target snapshot that is also unencrypted. If the source AMI is backed by an encrypted snapshot, copying it results in an identical target snapshot that is encrypted by the same customer master key (CMK). Copying an AMI backed by multiple snapshots preserves, by default, the source encryption status in each target snapshot.

If you specify encryption parameters while copying an AMI, you can encrypt or re-encrypt its backing snapshots. The following example shows a non-default case that supplies encryption parameters to the `CopyImage` action in order to change the target AMI's encryption state.

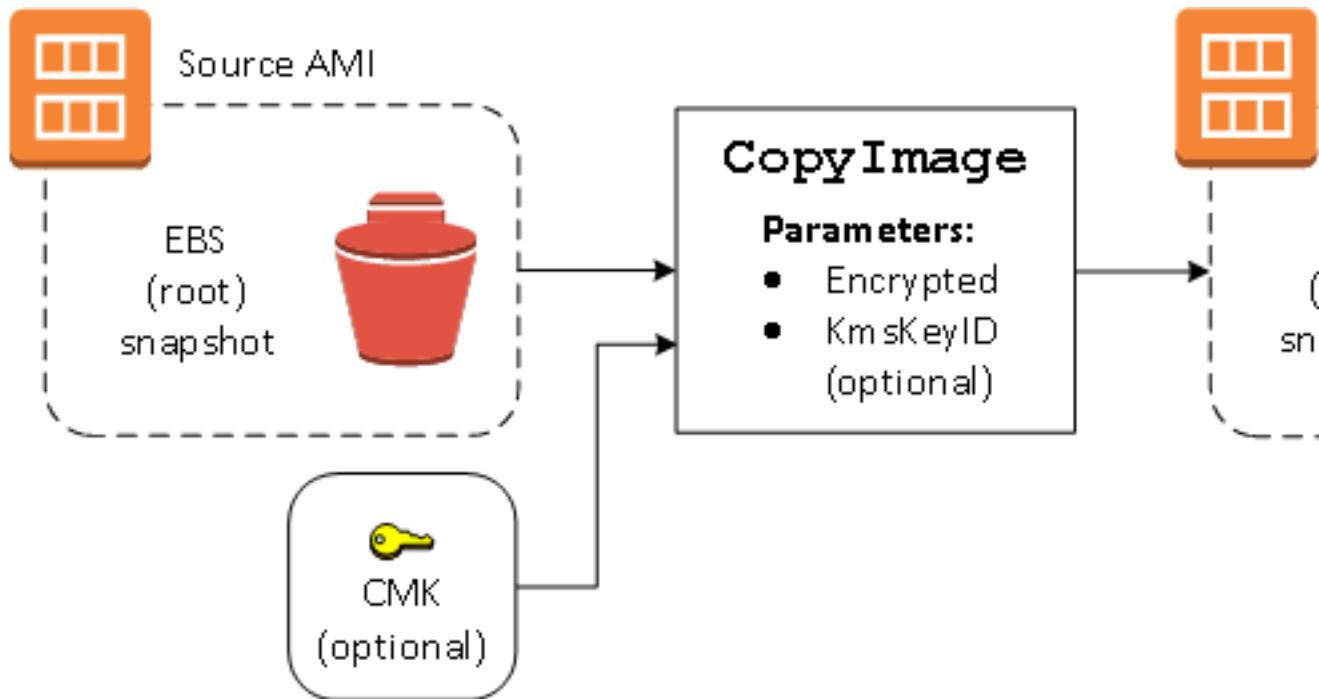
Copy an unencrypted source AMI to an encrypted target AMI

In this scenario, an AMI backed by an unencrypted root snapshot is copied to an AMI with an encrypted root snapshot. The `CopyImage` action is invoked with two encryption parameters, including a CMK. As a result, the encryption status of the root snapshot changes, so that the target AMI is backed by a root snapshot containing the same data as the source snapshot, but encrypted using the specified key. You

incur storage costs for the snapshots in both AMIs, as well as charges for any instances you launch from either AMI.

Note

Enabling [encryption by default \(p. 1124\)](#) has the same effect as setting the `Encrypted` parameter to `true` for all snapshots in the AMI.



Setting the `Encrypted` parameter encrypts the single snapshot for this instance. If you do not specify the `KmsKeyId` parameter, the default CMK is used to encrypt the snapshot copy.

For more information about copying AMIs with encrypted snapshots, see [Using encryption with EBS-backed AMIs \(p. 157\)](#).

Copying an AMI

You can copy an AMI as follows.

Prerequisite

Create or obtain an AMI backed by an Amazon EBS snapshot. Note that you can use the Amazon EC2 console to search a wide variety of AMIs provided by AWS. For more information, see [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#) and [Finding an AMI](#).

To copy an AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the console navigation bar, select the Region that contains the AMI. In the navigation pane, choose **Images, AMIs** to display the list of AMIs available to you in the Region.
3. Select the AMI to copy and choose **Actions, Copy AMI**.
4. In the **Copy AMI** dialog box, specify the following information and then choose **Copy AMI**:
 - **Destination region:** The Region in which to copy the AMI.
 - **Name:** A name for the new AMI. You can include operating system information in the name, as we do not provide this information when displaying details about the AMI.

- **Description:** By default, the description includes information about the source AMI so that you can distinguish a copy from its original. You can change this description as needed.
 - **Encryption:** Select this field to encrypt the target snapshots, or to re-encrypt them using a different key. If you have enabled [encryption by default](#), the **Encryption** option is set and cannot be unset from the AMI console.
 - **Master Key:** The KMS key to used to encrypt the target snapshots.
5. We display a confirmation page to let you know that the copy operation has been initiated and to provide you with the ID of the new AMI.

To check on the progress of the copy operation immediately, follow the provided link. To check on the progress later, choose **Done**, and then when you are ready, use the navigation bar to switch to the target Region (if applicable) and locate your AMI in the list of AMIs.

The initial status of the target AMI is **Pending** and the operation is complete when the status is **Available**.

To copy an AMI using the AWS CLI

You can copy an AMI using the [copy-image](#) command. You must specify both the source and destination Regions. You specify the source Region using the `--source-region` parameter. You can specify the destination Region using either the `--region` parameter or an environment variable. For more information, see [Configuring the AWS Command Line Interface](#).

When you encrypt a target snapshot during copying, you must specify these additional parameters: `--encrypted` and `--kms-key-id`.

To copy an AMI using the Tools for Windows PowerShell

You can copy an AMI using the [Copy-EC2Image](#) command. You must specify both the source and destination Regions. You specify the source Region using the `-SourceRegion` parameter. You can specify the destination Region using either the `-Region` parameter or the `Set-AWSDefaultRegion` command. For more information, see [Specifying AWS Regions](#).

When you encrypt a target snapshot during copying, you must specify these additional parameters: `-Encrypted` and `-KmsKeyId`.

Stopping a pending AMI copy operation

You can stop a pending AMI copy as follows.

To stop an AMI copy operation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the destination Region from the Region selector.
3. In the navigation pane, choose **AMIs**.
4. Select the AMI to stop copying and choose **Actions, Deregister**.
5. When asked for confirmation, choose **Continue**.

To stop an AMI copy operation using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [deregister-image](#) (AWS CLI)
- [Unregister-EC2Image](#) (AWS Tools for Windows PowerShell)

Obtaining billing information

You can determine the platform details and billing information associated with an Amazon Machine Image (AMI) before you launch an On-Demand Instance or Spot Instance, or purchase a Reserved Instance. For Spot Instances, you can use the platform details to confirm that the AMI is supported for Spot Instances. When purchasing a Reserved Instance, you can make sure that, for **Platform**, you select the correct value that maps to **Platform details** on the AMI. By knowing the billing information before launching an instance or purchasing a Reserved Instance, you reduce the chance of erroneously launching instances from incorrect AMIs and incurring unplanned costs.

For more information about instance pricing, see [Amazon EC2 pricing](#).

Contents

- [AMI billing information fields \(p. 169\)](#)
- [Platform details and usage operation values \(p. 169\)](#)
- [Viewing platform details and usage operation values \(p. 170\)](#)
- [Confirm billing information on your bill \(p. 171\)](#)

AMI billing information fields

The following fields provide billing information associated with an AMI:

Platform details

The platform details associated with the billing code of the AMI. For example, `Red Hat Enterprise Linux`.

Usage operation

The operation of the Amazon EC2 instance and the billing code that is associated with the AMI. For example, `RunInstances:0010`. **Usage operation** corresponds to the `lineitem/Operation` column on your AWS Cost and Usage Report (CUR) and in the [AWS Price List API](#). For the list of **Usage operation** codes, see [Platform details and usage operation values \(p. 169\)](#) in the following section.

You can view these fields on the **Instances** or **AMIs** page in the Amazon EC2 console, or in the response that is returned by the [describe-images](#) command.

Platform details and usage operation values

The following table lists the platform details and usage operation values that can be displayed on the **Instances** or **AMIs** page in the Amazon EC2 console, or in the response that is returned by the [describe-images](#) command.

Platform details	Usage operation **
Linux/UNIX	RunInstances
Red Hat BYOL Linux	RunInstances:00g0
Red Hat Enterprise Linux	RunInstances:0010
SQL Server Enterprise	RunInstances:0100
SQL Server Standard	RunInstances:0004

Platform details	Usage operation **
SQL Server Web	RunInstances:0200
SUSE Linux	RunInstances:000g
Windows	RunInstances:0002
Windows BYOL	RunInstances:0800
Windows with SQL Server Enterprise *	RunInstances:0102
Windows with SQL Server Standard *	RunInstances:0006
Windows with SQL Server Web *	RunInstances:0202

* If two software licenses are associated with an AMI, the **Platform details** field shows both.

** If you are running Spot Instances, the [lineitem/Operation](#) on your AWS Cost and Usage Report might be different from the **Usage operation** value that is listed here. For example, if [lineitem/Operation](#) displays RunInstances:0010:SV006, it means that Amazon EC2 is running Red Hat Enterprise Linux Spot Instance-hour in US East (Virginia) in VPC Zone #6.

Viewing platform details and usage operation values

You can view the platform details and usage operation values associated with an AMI from the AMI or from the instance. You can view these values in the Amazon EC2 console or by using the AWS CLI.

From the AMI

To view the platform details and usage operation associated with an AMI (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**, and then select an AMI.
3. On the **Details** tab, check the values for **Platform details** and **Usage operation**.

To view the platform details and usage operation associated with an AMI (AWS CLI)

Use the [describe-images](#) command.

```
$ aws ec2 describe-images --image-ids ami-0123456789EXAMPLE
```

The following example output shows the `PlatformDetails` and `UsageOperation` fields. In this example, the `ami-0123456789EXAMPLE` platform is Red Hat Enterprise Linux and the usage operation and billing code is `RunInstances:0010`.

```
{
  "Images": [
    {
      "VirtualizationType": "hvm",
      "Description": "Provided by Red Hat, Inc.",
      "Hypervisor": "xen",
      "EnaSupport": true,
      "SriovNetSupport": "simple",
      "ImageId": "ami-0123456789EXAMPLE",
```

```
"State": "available",
"BlockDeviceMappings": [
    {
        "DeviceName": "/dev/sda1",
        "Ebs": {
            "SnapshotId": "snap-111222333444aaabb",
            "DeleteOnTermination": true,
            "VolumeType": "gp2",
            "VolumeSize": 10,
            "Encrypted": false
        }
    }
],
"Architecture": "x86_64",
"ImageLocation": "123456789012/RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2",
"RootDeviceType": "ebs",
"OwnerId": "123456789012",
"PlatformDetails": "Red Hat Enterprise Linux",
"UsageOperation": "RunInstances:0010",
"RootDeviceName": "/dev/sda1",
"CreationDate": "2019-05-10T13:17:12.000Z",
"Public": true,
"ImageType": "machine",
"Name": "RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2"
}
]
```

From the instance

To view the platform details and usage operation associated with an AMI (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and then select an instance.
3. On the **Details** tab, check the values for **Platform details** and **Usage operation**.

To view the platform details and usage operation associated with an AMI (console)

After you have launched an instance, you can find the billing information by inspecting the `billingProducts` field in the instance metadata. For more information, see [Instance identity documents \(p. 684\)](#). Alternatively, you can use the `describe-instances` command to obtain the AMI ID for the instance, and then use the `describe-images` command, as described in the preceding procedure, to obtain the billing information from the `PlatformDetails` and `UsageOperation` fields in the response.

Confirm billing information on your bill

To ensure that you're not incurring unplanned costs, you can confirm that the billing information for an instance in your AWS Cost and Usage Report (CUR) matches the billing information associated with the AMI that you used to launch the instance. To confirm the billing information, find the instance ID in your CUR and check the corresponding value in the `lineitem/Operation` column. The value should match the value for **Usage operation** associated with the AMI.

For example, the AMI, `ami-0123456789EXAMPLE`, has the following billing information: **Platform details** = Red Hat Enterprise Linux and **Usage operation** = `RunInstances:0010`. If you launched an instance using this AMI, you can find the instance ID in your CUR and check the corresponding value in the `lineitem/Operation` column. In this example, the value should be `RunInstances:0010`.

Deregistering your Linux AMI

You can deregister an AMI when you have finished using it. After you deregister an AMI, you can't use it to launch new instances.

When you deregister an AMI, it doesn't affect any instances that you've already launched from the AMI. You'll continue to incur usage costs for these instances. Therefore, if you are finished with these instances, you should terminate them.

The procedure that you'll use to clean up your AMI depends on whether it is backed by Amazon EBS or instance store. For more information, see [Determining the root device type of your AMI \(p. 100\)](#).

Note

An AMI must be owned by your account in order to deregister it.

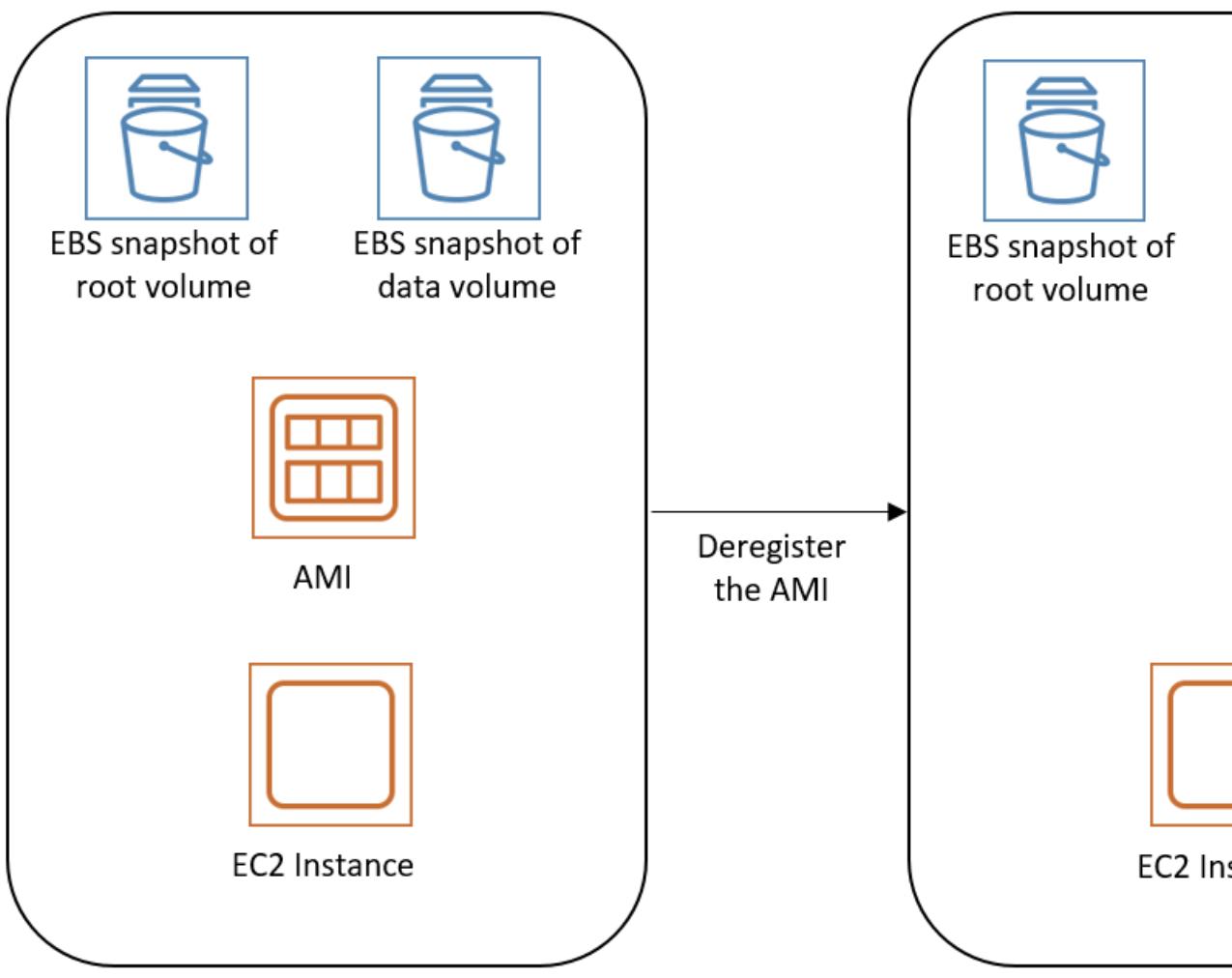
Contents

- [Cleaning up your Amazon EBS-backed AMI \(p. 172\)](#)
- [Cleaning up your instance store-backed AMI \(p. 174\)](#)

Cleaning up your Amazon EBS-backed AMI

When you deregister an Amazon EBS-backed AMI, it doesn't affect the snapshot(s) that were created for the volume(s) of the instance during the AMI creation process. You'll continue to incur storage costs for the snapshots. Therefore, if you are finished with the snapshots, you should delete them.

The following diagram illustrates the process for cleaning up your Amazon EBS-backed AMI.



Your AMI, its snapshots, and an instance launched from the AMI

To clean up your Amazon EBS-backed AMI

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**. Select the AMI, and take note of its ID — this can help you find the correct snapshot in the next step. Choose **Actions**, and then **Deregister**. When prompted for confirmation, choose **Continue**.

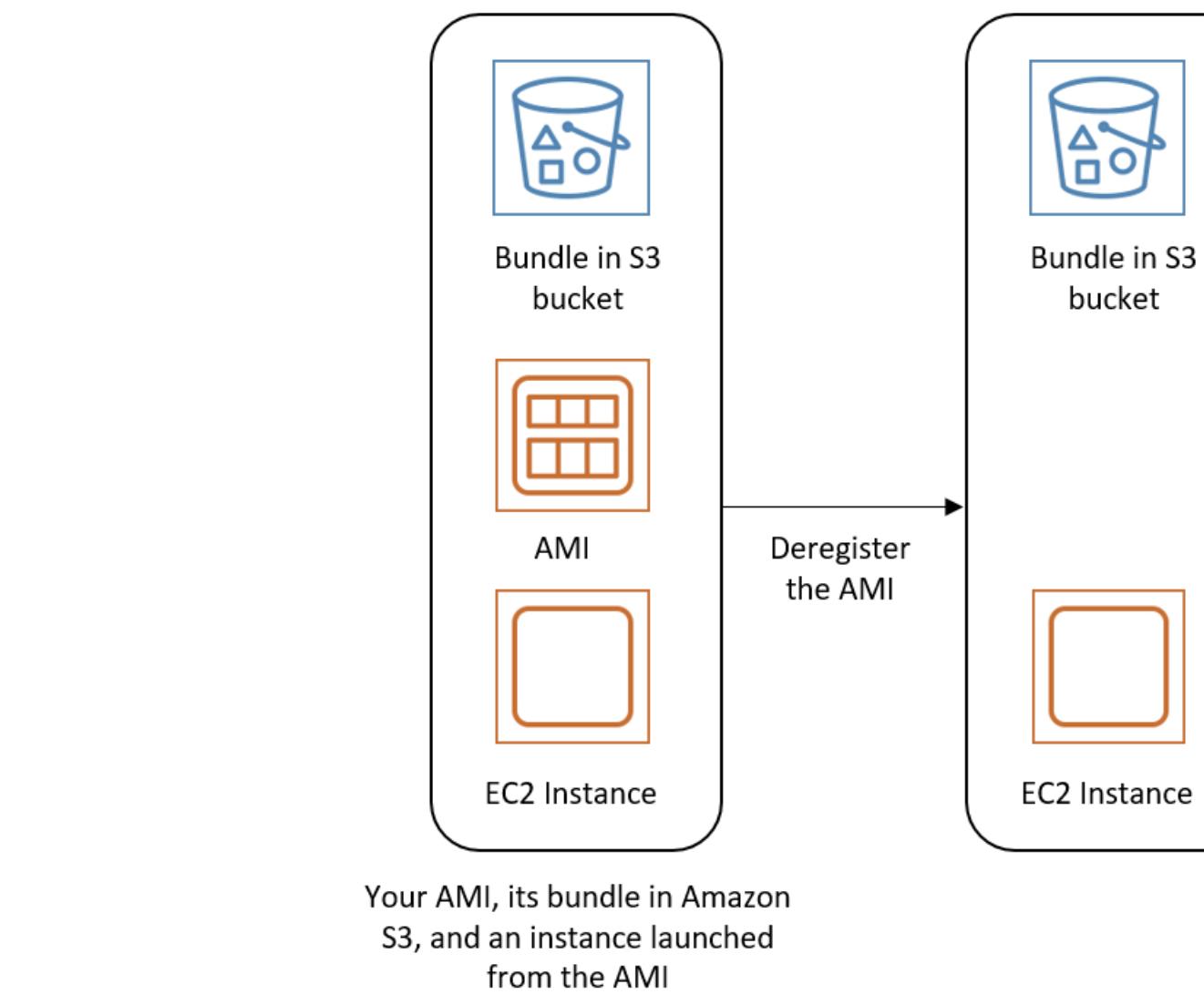
Note
It may take a few minutes before the console removes the AMI from the list. Choose **Refresh** to refresh the status.
3. In the navigation pane, choose **Snapshots**, and select the snapshot (look for the AMI ID in the **Description** column). Choose **Actions**, and then choose **Delete Snapshot**. When prompted for confirmation, choose **Yes, Delete**.

4. (Optional) If you are finished with an instance that you launched from the AMI, terminate it. In the navigation pane, choose **Instances**. Select the instance, choose **Actions**, then **Instance State**, and then **Terminate**. When prompted for confirmation, choose **Yes, Terminate**.

Cleaning up your instance store-backed AMI

When you deregister an instance store-backed AMI, it doesn't affect the files that you uploaded to Amazon S3 when you created the AMI. You'll continue to incur usage costs for these files in Amazon S3. Therefore, if you are finished with these files, you should delete them.

The following diagram illustrates the process for cleaning up your instance store-backed AMI.



To clean up your instance store-backed AMI

1. Deregister the AMI using the [deregister-image](#) command as follows.

```
aws ec2 deregister-image --image-id ami_id
```

2. Delete the bundle in Amazon S3 using the [ec2-delete-bundle \(p. 148\)](#) (AMI tools) command as follows.

```
ec2-delete-bundle -b myawsbucket/myami -a your_access_key_id -s your_secret_access_key  
-p image
```

3. (Optional) If you are finished with an instance that you launched from the AMI, you can terminate it using the [terminate-instances](#) command as follows.

```
aws ec2 terminate-instances --instance-ids instance_id
```

4. (Optional) If you are finished with the Amazon S3 bucket that you uploaded the bundle to, you can delete the bucket. To delete an Amazon S3 bucket, open the Amazon S3 console, select the bucket, choose **Actions**, and then choose **Delete**.

Amazon Linux

Amazon Linux is provided by Amazon Web Services (AWS). It is designed to provide a stable, secure, and high-performance execution environment for applications running on Amazon EC2. It also includes packages that enable easy integration with AWS, including launch configuration tools and many popular AWS libraries and tools. AWS provides ongoing security and maintenance updates for all instances running Amazon Linux. Many applications developed on CentOS (and similar distributions) run on Amazon Linux.

Contents

- [Amazon Linux availability \(p. 175\)](#)
- [Connecting to an Amazon Linux instance \(p. 176\)](#)
- [Identifying Amazon Linux images \(p. 176\)](#)
- [AWS command line tools \(p. 177\)](#)
- [Package repository \(p. 178\)](#)
- [Extras library \(Amazon Linux 2\) \(p. 180\)](#)
- [Accessing source packages for reference \(p. 181\)](#)
- [cloud-init \(p. 181\)](#)
- [Subscribing to Amazon Linux notifications \(p. 183\)](#)
- [Running Amazon Linux 2 as a virtual machine on premises \(p. 184\)](#)
- [Kernel Live Patching on Amazon Linux 2 \(p. 188\)](#)

Amazon Linux availability

AWS provides Amazon Linux 2 and the Amazon Linux AMI. If you are migrating from another Linux distribution to Amazon Linux, we recommend that you migrate to Amazon Linux 2.

The last version of the Amazon Linux AMI, 2018.03, reaches the end of standard support on December 31, 2020. For more information, see the following blog post: [Amazon Linux AMI end of life](#). If you are currently using the Amazon Linux AMI, we recommend that you migrate to Amazon Linux 2. To migrate to Amazon Linux 2, launch an instance or create a virtual machine using the current Amazon Linux 2 image. Install your applications, plus any required packages. Test your application, and make any changes required for it to run on Amazon Linux 2.

For more information, see [Amazon Linux 2](#) and [Amazon Linux AMI](#). For Amazon Linux Docker container images, see [amazonlinux](#) on Docker Hub.

Connecting to an Amazon Linux instance

Amazon Linux does not allow remote root SSH by default. Also, password authentication is disabled to prevent brute-force password attacks. To enable SSH logins to an Amazon Linux instance, you must provide your key pair to the instance at launch. You must also set the security group used to launch your instance to allow SSH access. By default, the only account that can log in remotely using SSH is `ec2-user`; this account also has `sudo` privileges. If you enable remote root login, be aware that it is less secure than relying on key pairs and a secondary user.

Identifying Amazon Linux images

Each image contains a unique `/etc/image-id` file that identifies it. This file contains the following information about the image:

- `image_name`, `image_version`, `image_arch` — Values from the build recipe that Amazon used to construct the image.
- `image_stamp` — A unique, random hex value generated during image creation.
- `image_date` — The UTC time of image creation, in `YYYYMMDDhhmmss` format
- `recipe_name`, `recipe_id` — The name and ID of the build recipe Amazon used to construct the image.

Amazon Linux contains an `/etc/system-release` file that specifies the current release that is installed. This file is updated using `yum` and is part of the `system-release` RPM.

Amazon Linux also contains a machine-readable version of `/etc/system-release` that follows the CPE specification; see `/etc/system-release-cpe`.

Amazon Linux 2

The following is an example of `/etc/image-id` for the current version of Amazon Linux 2:

```
[ec2-user ~]$ cat /etc/image-id
image_name="amzn2-ami-hvm"
image_version="2"
image_arch="x86_64"
image_file="amzn2-ami-hvm-2.0.20180810-x86_64.xfs.gpt"
image_stamp="8008-2abd"
image_date="20180811020321"
recipe_name="amzn2 ami"
recipe_id="c652686a-2415-9819-65fb-4dee-9792-289d-1e2846bd"
```

The following is an example of `/etc/system-release` for the current version of Amazon Linux 2:

```
[ec2-user ~]$ cat /etc/system-release
Amazon Linux 2
```

The following is an example of `/etc/os-release` for Amazon Linux 2:

```
[ec2-user ~]$ cat /etc/os-release
NAME="Amazon Linux"
VERSION="2"
```

```
ID="amzn"  
ID_LIKE="centos rhel fedora"  
VERSION_ID="2"  
PRETTY_NAME="Amazon Linux 2"  
ANSI_COLOR="0;33"  
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2"  
HOME_URL="https://amazonlinux.com/"
```

Amazon Linux AMI

The following is an example of `/etc/image-id` for the current Amazon Linux AMI:

```
[ec2-user ~]$ cat /etc/image-id  
image_name="amzn-ami-hvm"  
image_version="2018.03"  
image_arch="x86_64"  
image_file="amzn-ami-hvm-2018.03.0.20180811-x86_64.ext4.gpt"  
image_stamp="cc81-f2f3"  
image_date="20180811012746"  
recipe_name="amzn ami"  
recipe_id="5b283820-dc60-a7ea-d436-39fa-439f-02ea-5c802dbd"
```

The following is an example of `/etc/system-release` for the current Amazon Linux AMI:

```
[ec2-user ~]$ cat /etc/system-release  
Amazon Linux AMI release 2018.03
```

AWS command line tools

The following command line tools for AWS integration and usage are included in the Amazon Linux AMI, or in the default repositories for Amazon Linux 2. For the complete list of packages in the Amazon Linux AMI, see [Amazon Linux AMI 2017.09 Packages](#).

- aws-amitools-ec2
- aws-apitools-as
- aws-apitools-cfn
- aws-apitools-elb
- aws-apitools-mon
- aws-cfn-bootstrap
- aws-cli

Amazon Linux 2 and the minimal versions of Amazon Linux (`amzn-ami-minimal-*` and `amzn2-ami-minimal-*`) do not always contain all of these packages; however, you can install them from the default repositories using the following command:

```
[ec2-user ~]$ sudo yum install -y package_name
```

For instances launched using IAM roles, a simple script has been included to prepare `AWS_CREDENTIAL_FILE`, `JAVA_HOME`, `AWS_PATH`, `PATH`, and product-specific environment variables after a credential file has been installed to simplify the configuration of these tools.

Also, to allow the installation of multiple versions of the API and AMI tools, we have placed symbolic links to the desired versions of these tools in `/opt/aws`, as described here:

/opt/aws/bin

Symbolic links to /bin directories in each of the installed tools directories.

/opt/aws/{apitools|amitools}

Products are installed in directories of the form *name-version* and a symbolic link *name* that is attached to the most recently installed version.

/opt/aws/{apitools|amitools}/*name*/environment.sh

Used by /etc/profile.d/aws-apitools-common.sh to set product-specific environment variables, such as EC2_HOME.

Package repository

Amazon Linux 2 and the Amazon Linux AMI are designed to be used with online package repositories hosted in each Amazon EC2 AWS Region. These repositories provide ongoing updates to packages in Amazon Linux 2 and the Amazon Linux AMI, as well as access to hundreds of additional common open-source server applications. The repositories are available in all Regions and are accessed using **yum** update tools. Hosting repositories in each Region enables us to deploy updates quickly and without any data transfer charges.

Amazon Linux 2 and the Amazon Linux AMI are updated regularly with security and feature enhancements. If you do not need to preserve data or customizations for your instances, you can simply launch new instances using the current AMI. If you need to preserve data or customizations for your instances, you can maintain those instances through the Amazon Linux package repositories. These repositories contain all the updated packages. You can choose to apply these updates to your running instances. Older versions of the AMI and update packages continue to be available for use, even as new versions are released.

Important

Your instance must have access to the internet in order to access the repository.

To install packages, use the following command:

```
[ec2-user ~]$ sudo yum install package
```

For the Amazon Linux AMI, access to the Extra Packages for Enterprise Linux (EPEL) repository is configured, but it is not enabled by default. Amazon Linux 2 is not configured to use the EPEL repository. EPEL provides third-party packages in addition to those that are in the repositories. The third-party packages are not supported by AWS. You can enable the EPEL repository with the following commands:

- For Amazon Linux 2:

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

- For the Amazon Linux AMI:

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

If you find that Amazon Linux does not contain an application you need, you can simply install the application directly on your Amazon Linux instance. Amazon Linux uses RPMs and **yum** for package management, and that is likely the simplest way to install new applications. You should always check to see if an application is available in our central Amazon Linux repository first, because many applications are available there. These applications can easily be added to your Amazon Linux instance.

To upload your applications onto a running Amazon Linux instance, use **scp** or **sftp** and then configure the application by logging on to your instance. Your applications can also be uploaded during the instance launch by using the **PACKAGE_SETUP** action from the built-in cloud-init package. For more information, see [cloud-init \(p. 181\)](#).

Security updates

Security updates are provided using the package repositories as well as updated AMI security alerts are published in the [Amazon Linux Security Center](#). For more information about AWS security policies or to report a security problem, go to the [AWS Security Center](#).

Amazon Linux is configured to download and install critical or important security updates at launch time. We recommend that you make the necessary updates for your use case after launch. For example, you may want to apply all updates (not just security updates) at launch, or evaluate each update and apply only the ones applicable to your system. This is controlled using the following cloud-init setting: `repo_upgrade`. The following snippet of cloud-init configuration shows how you can change the settings in the user data text you pass to your instance initialization:

```
#cloud-config
repo_upgrade: security
```

The possible values for `repo_upgrade` are as follows:

critical

Apply outstanding critical security updates.

important

Apply outstanding critical and important security updates.

medium

Apply outstanding critical, important, and medium security updates.

low

Apply all outstanding security updates, including low-severity security updates.

security

Apply outstanding critical or important updates that Amazon marks as security updates.

bugfix

Apply updates that Amazon marks as bug fixes. Bug fixes are a larger set of updates, which include security updates and fixes for various other minor bugs.

all

Apply all applicable available updates, regardless of their classification.

none

Do not apply any updates to the instance on startup.

The default setting for `repo_upgrade` is `security`. That is, if you don't specify a different value in your user data, by default, Amazon Linux performs the security upgrades at launch for any packages installed at that time. Amazon Linux also notifies you of any updates to the installed packages by listing the number of available updates upon login using the `/etc/motd` file. To install these updates, you need to run **sudo yum upgrade** on the instance.

Repository configuration

With Amazon Linux, AMIs are treated as snapshots in time, with a repository and update structure that always gives you the latest packages when you run **yum update -y**.

The repository structure is configured to deliver a continuous flow of updates that enable you to roll from one version of Amazon Linux to the next. For example, if you launch an instance from an older version of the Amazon Linux AMI (such as 2017.09 or earlier) and run **yum update -y**, you end up with the latest packages.

You can disable rolling updates by enabling the *lock-on-launch* feature. The lock-on-launch feature locks your instance to receive updates only from the specified release of the AMI. For example, you can launch a 2017.09 AMI and have it receive only the updates that were released prior to the 2018.03 AMI, until you are ready to migrate to the 2018.03 AMI.

Important

If you lock to a version of the repositories that is not the latest, you do not receive further updates. To receive a continuous flow of updates, you must use the latest AMI, or consistently update your AMI with the repositories pointed to latest.

To enable lock-on-launch in new instances, launch it with the following user data passed to cloud-init:

```
#cloud-config
repo_releasever: 2017.09
```

To lock existing instances to their current AMI version

1. Edit `/etc/yum.conf`.
2. Comment out `releasever=latest`.
3. To clear the cache, run **yum clean all**.

Extras library (Amazon Linux 2)

With Amazon Linux 2, you can use the Extras Library to install application and software updates on your instances. These software updates are known as *topics*. You can install a specific version of a topic or omit the version information to use the most recent version.

To list the available topics, use the following command:

```
[ec2-user ~]$ amazon-linux-extras list
```

To enable a topic and install the latest version of its package to ensure freshness, use the following command:

```
[ec2-user ~]$ sudo amazon-linux-extras install topic
```

To enable topics and install specific versions of their packages to ensure stability, use the following command:

```
[ec2-user ~]$ sudo amazon-linux-extras install topic=version topic=version
```

To remove a package installed from a topic, use the following command:

```
[ec2-user ~]$ sudo yum remove $(yum list installed | grep amzn2extra-topic | awk '{ print $1 }')
```

Note

This command does not remove packages that were installed as dependencies of the extra.

To disable a topic and make the packages inaccessible to the yum package manager, use the following command:

```
[ec2-user ~]$ sudo amazon-linux-extras disable topic
```

Important

This command is intended for advanced users. Improper usage of this command could cause package compatibility conflicts.

Accessing source packages for reference

You can view the source of packages you have installed on your instance for reference purposes by using tools provided in Amazon Linux. Source packages are available for all of the packages included in Amazon Linux and the online package repository. Simply determine the package name for the source package you want to install and use the **yumdownloader --source** command to view source within your running instance. For example:

```
[ec2-user ~]$ yumdownloader --source bash
```

The source RPM can be unpacked, and, for reference, you can view the source tree using standard RPM tools. After you finish debugging, the package is available for use.

cloud-init

The cloud-init package is an open-source application built by Canonical that is used to bootstrap Linux images in a cloud computing environment, such as Amazon EC2. Amazon Linux contains a customized version of cloud-init. It enables you to specify actions that should happen to your instance at boot time. You can pass desired actions to cloud-init through the user data fields when launching an instance. This means you can use common APIs for many use cases and configure them dynamically at startup. Amazon Linux also uses cloud-init to perform initial configuration of the ec2-user account.

For more information, see the [cloud-init documentation](#).

Amazon Linux uses the cloud-init actions found in /etc/cloud/cloud.cfg.d and /etc/cloud/cloud.cfg. You can create your own cloud-init action files in /etc/cloud/cloud.cfg.d. All files in this directory are read by cloud-init. They are read in lexical order, and later files overwrite values in earlier files.

The cloud-init package performs these (and other) common configuration tasks for instances at boot:

- Set the default locale.
- Set the hostname.
- Parse and handle user data.
- Generate host private SSH keys.
- Add a user's public SSH keys to .ssh/authorized_keys for easy login and administration.
- Prepare the repositories for package management.
- Handle package actions defined in user data.
- Execute user scripts found in user data.

- Mount instance store volumes, if applicable.
 - By default, the `ephemeral0` instance store volume is mounted at `/media/ephemeral0` if it is present and contains a valid file system; otherwise, it is not mounted.
 - By default, any swap volumes associated with the instance are mounted (only for `m1.small` and `c1.medium` instance types).
- You can override the default instance store volume mount with the following cloud-init directive:

```
#cloud-config
mounts:
- [ ephemeral0 ]
```

For more control over mounts, see [Mounts](#) in the cloud-init documentation.

- Instance store volumes that support TRIM are not formatted when an instance launches, so you must partition and format them before you can mount them. For more information, see [Instance store volume TRIM support \(p. 1201\)](#). You can use the `disk_setup` module to partition and format your instance store volumes at boot. For more information, see [Disk Setup](#) in the cloud-init documentation.

Supported user-data formats

The cloud-init package supports user-data handling of a variety of formats:

- Gzip
 - If user-data is gzip compressed, cloud-init decompresses the data and handles it appropriately.
- MIME multipart
 - Using a MIME multipart file, you can specify more than one type of data. For example, you could specify both a user-data script and a cloud-config type. Each part of the multipart file can be handled by cloud-init if it is one of the supported formats.
- Base64 decoding
 - If user-data is base64-encoded, cloud-init determines if it can understand the decoded data as one of the supported types. If it understands the decoded data, it decodes the data and handles it appropriately. If not, it returns the base64 data intact.
- User-Data script
 - Begins with `#!` or `Content-Type: text/x-shellscript`.
 - The script is executed by `/etc/init.d/cloud-init-user-scripts` during the first boot cycle. This occurs late in the boot process (after the initial configuration actions are performed).
- Include file
 - Begins with `#include` or `Content-Type: text/x-include-url`.
 - This content is an include file. The file contains a list of URLs, one per line. Each of the URLs is read, and their content passed through this same set of rules. The content read from the URL can be gzipped, MIME-multi-part, or plaintext.
- Cloud Config Data
 - Begins with `#cloud-config` or `Content-Type: text/cloud-config`.
 - This content is cloud-config data. For a commented example of supported configuration formats, see the examples.
- Upstart job
 - Begins with `#upstart-job` or `Content-Type: text/upstart-job`.
 - This content is stored in a file in `/etc/init`, and upstart consumes the content as per other upstart jobs.
- Cloud Boothook

- Begins with `#cloud-boothook` or `Content-Type: text/cloud-boothook`.
 - This content is booothook data. It is stored in a file under `/var/lib/cloud` and then executed immediately.
 - This is the earliest "hook" available. There is no mechanism provided for running it only one time. The booothook must take care of this itself. It is provided with the instance ID in the environment variable `INSTANCE_ID`. Use this variable to provide a once-per-instance set of booothook data.

Subscribing to Amazon Linux notifications

To be notified when new AMIs are released, you can subscribe using Amazon SNS.

To subscribe to Amazon Linux notifications

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
 2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must select the Region in which the SNS notification that you are subscribing to was created.
 3. In the navigation pane, choose **Subscriptions**, **Create subscription**.
 4. For the **Create subscription** dialog box, do the following:
 - a. [Amazon Linux 2] For **Topic ARN**, copy and paste the following Amazon Resource Name (ARN):
arn:aws:sns:us-east-1:137112412989:amazon-linux-2-ami-updates.
 - b. [Amazon Linux] For **Topic ARN**, copy and paste the following Amazon Resource Name (ARN):
arn:aws:sns:us-east-1:137112412989:amazon-linux-ami-updates.
 - c. For **Protocol**, choose **Email**.
 - d. For **Endpoint**, enter an email address that you can use to receive the notifications.
 - e. Choose **Create subscription**.
 5. You receive a confirmation email with the subject line "AWS Notification - Subscription Confirmation". Open the email and choose **Confirm subscription** to complete your subscription.

Whenever AMIs are released, we send notifications to the subscribers of the corresponding topic. To stop receiving these notifications, use the following procedure to unsubscribe.

To unsubscribe from Amazon Linux notifications

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
 2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must use the Region in which the SNS notification was created.
 3. In the navigation pane, choose **Subscriptions**, select the subscription, and choose **Actions, Delete subscriptions**.
 4. When prompted for confirmation, choose **Delete**.

Amazon Linux AMI SNS message format

The schema for the SNS message is as follows.

```
"ReleaseVersion": {  
    "description": "Major release (ex. 2018.03)",  
    "type": "string"  
},  
"ImageVersion": {  
    "description": "Full release (ex. 2018.03.0.20180412)",  
    "type": "string"  
},  
"ReleaseNotes": {  
    "description": "Human-readable string with extra information",  
    "type": "string"  
},  
"Regions": {  
    "type": "object",  
    "description": "Each key will be a region name (ex. us-east-1)",  
    "additionalProperties": {  
        "type": "array",  
        "items": {  
            "type": "object",  
            "properties": {  
                "Name": {  
                    "description": "AMI Name (ex. amzn-ami-hvm-2018.03.0.20180412-x86_64-gp2)",  
                    "type": "string"  
                },  
                "ImageId": {  
                    "description": "AMI Name (ex. ami-467ca739)",  
                    "type": "string"  
                }  
            },  
            "required": [  
                "Name",  
                "ImageId"  
            ]  
        }  
    }  
},  
"required": [  
    "ReleaseVersion",  
    "ImageVersion",  
    "ReleaseNotes",  
    "Regions"  
]  
}  
},  
"required": [  
    "v1"  
]  
}
```

Running Amazon Linux 2 as a virtual machine on premises

Use the Amazon Linux 2 virtual machine (VM) images for on-premises development and testing. These images are available for use on the following virtualization platforms:

- VMWare
- KVM
- VirtualBox (Oracle VM)
- Microsoft Hyper-V

To use the Amazon Linux 2 virtual machine images with one of the supported virtualization platforms, do the following:

- [Step 1: Prepare the seed.iso boot image \(p. 185\)](#)
- [Step 2: Download the Amazon Linux 2 VM image \(p. 186\)](#)
- [Step 3: Boot and connect to your new VM \(p. 187\)](#)

Step 1: Prepare the seed.iso boot image

The `seed.iso` boot image includes the initial configuration information that is needed to boot your new VM, such as the network configuration, host name, and user data.

Note

The `seed.iso` boot image includes only the configuration information required to boot the VM. It does not include the Amazon Linux 2 operating system files.

To generate the `seed.iso` boot image, you need two configuration files:

- `meta-data`—This file includes the hostname and static network settings for the VM.
- `user-data`—This file configures user accounts, and specifies their passwords, key pairs, and access mechanisms. By default, the Amazon Linux 2 VM image creates a `ec2-user` user account. You use the `user-data` configuration file to set the password for the default user account.

To create the `seed.iso` boot disc

1. Create a new folder named `seedconfig` and navigate into it.
2. Create the `meta-data` configuration file.
 - a. Create a new file named `meta-data`.
 - b. Open the `meta-data` file using your preferred editor and add the following.

```
local-hostname: vm_hostname
# eth0 is the default network interface enabled in the image. You can configure
static network settings with an entry like the following.
network-interfaces: |
    auto eth0
    iface eth0 inet static
        address 192.168.1.10
        network 192.168.1.0
        netmask 255.255.255.0
        broadcast 192.168.1.255
        gateway 192.168.1.254
```

Replace `vm_hostname` with a VM host name of your choice, and configure the network settings as required.

- c. Save and close the `meta-data` configuration file.

For an example `meta-data` configuration file that specifies a VM hostname (`amazonlinux.onprem`), configures the default network interface (`eth0`), and specifies static IP addresses for the necessary network devices, see the [sample Seed.iso file](#).

3. Create the `user-data` configuration file.
 - a. Create a new file named `user-data`.
 - b. Open the `user-data` file using your preferred editor and add the following.

```
#cloud-config
# vim:syntax=yaml
users:
  # A user by the name `ec2-user` is created in the image by default.
  - default
chpasswd:
  list: |
    ec2-user:plain_text_password
# In the above line, do not add any spaces after 'ec2-user:'.
```

Replace *plain_text_password* with a password of your choice for the default ec2-user user account.

- c. (Optional) By default, cloud-init applies network settings each time the VM boots. Add the following to prevent cloud-init from applying network settings at each boot, and to retain the network settings applied during the first boot.

```
# NOTE: Cloud-init applies network settings on every boot by default. To retain
network settings from first
boot, add following 'write_files' section:
write_files:
  - path: /etc/cloud/cloud.cfg.d/80_disable_network_after_firstboot.cfg
    content: |
      # Disable network configuration after first boot
      network:
        config: disabled
```

- d. Save and close the user-data configuration file.

You can also create additional user accounts and specify their access mechanisms, passwords, and key pairs. For more information about the supported directives, see [Modules](#). For an example user-data file that creates three additional users and specifies a custom password for the default ec2-user user account, see the [sample Seed.iso file](#).

4. Create the seed.iso boot image using the meta-data and user-data configuration files.

For Linux, use a tool such as **genisoimage**. Navigate into the seedconfig folder, and execute the following command.

```
$ genisoimage -output seed.iso -volid cidata -joliet -rock user-data meta-data
```

For macOS, use a tool such as **hdiutil**. Navigate one level up from the seedconfig folder, and execute the following command.

```
$ hdiutil makehybrid -o seed.iso -hfs -joliet -iso -default-volume-name cidata
seedconfig/
```

Step 2: Download the Amazon Linux 2 VM image

We offer a different Amazon Linux 2 VM image for each of the supported virtualization platforms. Download the correct VM image for your chosen platform:

- [VMWare](#)
- [KVM](#)
- [Oracle VirtualBox](#)
- [Microsoft Hyper-V](#)

Step 3: Boot and connect to your new VM

To boot and connect to your new VM, you must have the `seed.iso` boot image (created in Step 1) and an Amazon Linux 2 VM image (downloaded in Step 2). The steps vary depending on your chosen VM platform.

VMWare vSphere

The VM image for VMware is made available in the OVF format.

To boot the VM using VMWare vSphere

1. Create a new datastore for the `seed.iso` file, or add it to an existing datastore.
2. Deploy the OVF template, but do not start the VM yet.
3. In the **Navigator** panel, right-click the new virtual machine and choose **Edit Settings**.
4. On the **Virtual Hardware** tab, for **New device**, choose **CD/DVD Drive**, and then choose **Add**.
5. For **New CD/DVD Drive**, choose **Datastore ISO File**. Select the datastore to which you added the `seed.iso` file, browse to and select the `seed.iso` file, and then choose **OK**.
6. For **New CD/DVD Drive**, select **Connect**, and then choose **OK**.

After you have associated the datastore with the VM, you should be able to boot it.

KVM

To boot the VM using KVM

1. Open the **Create new VM** wizard.
2. For Step 1, choose **Import existing disk image**.
3. For Step 2, browse to and select the VM image. For **OS type** and **Version**, choose **Linux** and **Red Hat Enterprise Linux 7.0** respectively.
4. For Step 3, specify the amount of RAM and the number of CPUs to use.
5. For Step 4, enter a name for the new VM and select **Customize configuration before install**, and choose **Finish**.
6. In the Configuration window for the VM, choose **Add Hardware**.
7. In the **Add New Virtual Hardware** window, choose **Storage**.
8. In the Storage configuration, choose **Select or create custom storage**. For **Device type**, choose **CDROM device**. Choose **Manage**, **Browse Local**, and then navigate to and select the `seed.iso` file. Choose **Finish**.
9. Choose **Begin Installation**.

Oracle VirtualBox

To boot the VM using Oracle VirtualBox

1. Open Oracle VirtualBox and choose **New**.
2. For **Name**, enter a descriptive name for the virtual machine, and for **Type** and **Version**, select **Linux** and **Red Hat (64-bit)** respectively. Choose **Continue**.
3. For **Memory size**, specify the amount of memory to allocate to the virtual machine, and then choose **Continue**.
4. For **Hard disk**, choose **Use an existing virtual hard disk file**, browse to and open the VM image, and then choose **Create**.
5. Before you start the VM, you must load the `seed.iso` file in the virtual machine's virtual optical drive:

- a. Select the new VM, choose **Settings**, and then choose **Storage**.
- b. In the **Storage Devices** list, under **Controller: IDE**, choose the *Empty* optical drive.
- c. In the **Attributes** section for the optical drive, choose the browse button, select **Choose Virtual Optical Disk File**, and then select the `seed.iso` file. Choose **OK** to apply the changes and close the Settings.

After you have added the `seed.iso` file to the virtual optical drive, you should be able to start the VM.

Microsoft Hyper-V

The VM image for Microsoft Hyper-V is compressed into a zip file. You must extract the contents of the zip file.

To boot the VM using Microsoft Hyper-V

1. Open the **New Virtual Machine Wizard**.
2. When prompted to select a generation, select **Generation 1**.
3. When prompted to configure the network adapter, for **Connection** choose **External**.
4. When prompted to connect a virtual hard disk, choose **Use an existing virtual hard disk**, choose **Browse**, and then navigate to and select the VM image. Choose **Finish** to create the VM.
5. Right-click the new VM and choose **Settings**. In the **Settings** window, under **IDE Controller 1**, choose **DVD Drive**.
6. For the DVD drive, choose **Image file** and then browse to and select the `seed.iso` file.
7. Apply the changes and start the VM.

After the VM has booted, log in using one of the user accounts that is defined in the user-data configuration file. For virtualization platforms other than VMWare, you can disconnect the `seed.iso` boot image from the VM after you have logged in for the first time.

Kernel Live Patching on Amazon Linux 2

Kernel Live Patching for Amazon Linux 2 enables you to apply security vulnerability and critical bug patches to a running Linux kernel, without reboots or disruptions to running applications. This allows you to benefit from improved service and application availability, while keeping your infrastructure secure and up to date.

AWS releases two types of kernel live patches for Amazon Linux 2:

- **Security updates**—Include updates for Linux common vulnerabilities and exposures (CVE). These updates are typically rated as *important* or *critical* using the Amazon Linux Security Advisory ratings. They generally map to a Common Vulnerability Scoring System (CVSS) score of 7 and higher. In some cases, AWS might provide updates before a CVE is assigned. In these cases, the patches might appear as bug fixes.
- **Bug fixes**—Include fixes for critical bugs and stability issues that are not associated with CVEs.

AWS provides kernel live patches for an Amazon Linux 2 kernel version for up to 3 months after its release. After the 3-month period, you must update to a later kernel version to continue to receive kernel live patches.

Amazon Linux 2 kernel live patches are made available as signed RPM packages in the existing Amazon Linux 2 repositories. The patches can be installed on individual instances using existing **yum** workflows, or they can be installed on a group of managed instances using AWS Systems Manager.

Kernel Live Patching on Amazon Linux 2 is provided at no additional cost.

Topics

- [Supported configurations and prerequisites \(p. 189\)](#)
- [Working with Kernel Live Patching \(p. 189\)](#)
- [Limitations \(p. 193\)](#)
- [Frequently asked questions \(p. 193\)](#)

Supported configurations and prerequisites

Kernel Live Patching is supported on Amazon EC2 instances and [on-premises virtual machines \(p. 184\)](#) running Amazon Linux 2.

To use Kernel Live Patching on Amazon Linux 2, you must use:

- A 64-bit (x86_64) architecture that is supported by Amazon Linux 2
- Amazon Linux 2 with kernel version 4.14.165-131.185 or later

Note

The 64-bit ARM (arm64) architecture is not supported.

Working with Kernel Live Patching

You can enable and use Kernel Live Patching on individual instances using the command line on the instance itself, or you can enable and use Kernel Live Patching on a group of managed instances using AWS Systems Manager.

The following sections explain how to enable and use Kernel Live Patching on individual instances using the command line.

For more information about enabling and using Kernel Live Patching on a group of managed instances, see [Use Kernel Live Patching on Amazon Linux 2 instances](#) in the *AWS Systems Manager User Guide*.

Topics

- [Enabling Kernel Live Patching \(p. 189\)](#)
- [Viewing the available kernel live patches \(p. 190\)](#)
- [Applying kernel live patches \(p. 191\)](#)
- [View the applied kernel live patches \(p. 192\)](#)
- [Disabling Kernel Live Patching \(p. 192\)](#)

Enabling Kernel Live Patching

Kernel Live Patching is disabled by default on Amazon Linux 2. To use live patching, you must install the `yum` plugin for Kernel Live Patching and enable the live patching functionality.

Prerequisites

Kernel Live Patching requires `binutils`. If you do not have `binutils` installed, install it using the following command:

```
$ sudo yum install binutils
```

To enable Kernel Live Patching

1. Kernel live patches are available for Amazon Linux 2 with kernel version 4.14.165-131.185 or later. To check your kernel version, run the following command.

```
$ sudo yum list kernel
```

2. If you already have a supported kernel version, skip this step. If you do not have a supported kernel version, run the following commands to update the kernel to the latest version and to reboot the instance.

```
$ sudo yum install -y kernel
```

```
$ sudo reboot
```

3. Install the **yum** plugin for Kernel Live Patching.

```
$ sudo yum install -y yum-plugin-kernel-livepatch
```

4. Enable the **yum** plugin for Kernel Live Patching.

```
$ sudo yum kernel-livepatch enable -y
```

This command also installs the latest version of the kernel live patch RPM from the configured repositories.

5. To confirm that the **yum** plugin for kernel live patching has installed successfully, run the following command.

```
$ rpm -qa | grep kernel-livepatch
```

When you enable Kernel Live Patching, an empty kernel live patch RPM is automatically applied. If Kernel Live Patching was successfully enabled, this command returns a list that includes the initial empty kernel live patch RPM.

6. Update and start the **kpatch** service. This service loads all of the kernel live patches upon initialization or at boot.

```
$ sudo yum update kpatch-runtime
```

```
$ sudo systemctl enable kpatch.service
```

7. Configure the Amazon Linux 2 Kernel Live Patching repository, which contains the kernel live patches.

```
$ sudo amazon-linux-extras enable livepatch
```

Viewing the available kernel live patches

Amazon Linux security alerts are published to the Amazon Linux Security Center. For more information about the Amazon Linux 2 security alerts, which include alerts for kernel live patches, see the [Amazon Linux Security Center](#). Kernel live patches are prefixed with ALASLIVEPATCH. The Amazon Linux Security Center might not list kernel live patches that address bugs.

You can also discover the available kernel live patches for advisories and CVEs using the command line.

To list all available kernel live patches for advisories

Use the following command.

```
$ yum updateinfo list
```

The following shows example output.

```
Loaded plugins: extras_suggestions, kernel-livepatch, langpacks, priorities, update-motd
ALAS2LIVEPATCH-2020-002 important/Sec. kernel-
livepatch-4.14.165-133.209-1.0-3.amzn2.x86_64
ALAS2LIVEPATCH-2020-005 medium/Sec. kernel-livepatch-4.14.165-133.209-1.0-4.amzn2.x86_64
updateinfo list done
```

To list all available kernel live patches for CVEs

Use the following command.

```
$ yum updateinfo list cves
```

The following shows example output.

```
Loaded plugins: extras_suggestions, kernel-livepatch, langpacks, priorities, update-
motd
motd
alas2-core/2/x86_64 | 2.4 kB 00:00:00
CVE-2019-15918 important/Sec. kernel-livepatch-4.14.165-133.209-1.0-3.amzn2.x86_64
CVE-2019-20096 important/Sec. kernel-livepatch-4.14.165-133.209-1.0-3.amzn2.x86_64
CVE-2020-8648 medium/Sec. kernel-livepatch-4.14.165-133.209-1.0-4.amzn2.x86_64
updateinfo list done
```

Applying kernel live patches

You apply kernel live patches using the **yum** package manager in the same way that you would apply regular updates. The **yum** plugin for Kernel Live Patching manages the kernel live patches that are to be applied and eliminates the need to reboot.

Tip

We recommend that you update your kernel regularly using Kernel Live Patching to ensure that it remains secure and up to date.

You can choose to apply a specific kernel live patch, or to apply any available kernel live patches along with your regular security updates.

To apply a specific kernel live patch

1. Get the kernel live patch version using one of the commands described in [Viewing the available kernel live patches \(p. 190\)](#).
2. Apply the kernel live patch for your Amazon Linux 2 kernel.

```
$ sudo yum install kernel-livepatch-<kernel_version>.x86_64
```

For example, the following command applies a kernel live patch for Amazon Linux 2 kernel version 4.14.165-133.209.

```
$ sudo yum install kernel-livepatch-4.14.165-133.209-1.0-4.amzn2.x86_64
```

To apply any available kernel live patches along with your regular security updates

Use the following command.

```
$ sudo yum update --security
```

Omit the --security option to include bug fixes.

Important

- The kernel version is not updated after applying kernel live patches. The version is only updated to the new version after the instance is rebooted.
- An Amazon Linux 2 kernel receives kernel live patches for a period of three months. After the three month period has lapsed, no new kernel live patches are released for that kernel version. To continue to receive kernel live patches after the three-month period, you must reboot the instance to move to the new kernel version, which will then continue receiving kernel live patches for the next three months. To check the support window for your kernel version, run `yum kernel-livepatch supported`.

View the applied kernel live patches

To view the applied kernel live patches

Use the following command.

```
$ kpatch list
```

The command returns a list of the loaded and installed security update kernel live patches. The following is example output.

```
Loaded patch modules:  
livepatch_cifs_lease_buffer_len [enabled]  
livepatch_CVE_2019_20096 [enabled]  
livepatch_CVE_2020_8648 [enabled]  
  
Installed patch modules:  
livepatch_cifs_lease_buffer_len (4.14.165-133.209.amzn2.x86_64)  
livepatch_CVE_2019_20096 (4.14.165-133.209.amzn2.x86_64)  
livepatch_CVE_2020_8648 (4.14.165-133.209.amzn2.x86_64)
```

Note

A single kernel live patch can include and install multiple live patches.

Disabling Kernel Live Patching

If you no longer need to use Kernel Live Patching, you can disable it at any time.

To disable Kernel Live Patching

1. Remove the RPM packages for the applied kernel live patches.

```
$ sudo yum kernel-livepatch disable
```

2. Uninstall the `yum` plugin for Kernel Live Patching.

```
$ sudo yum remove yum-plugin-kernel-livepatch
```

3. Reboot the instance.

```
$ sudo reboot
```

Limitations

Kernel Live Patching has the following limitations:

- While applying a kernel live patch, you can't perform hibernation, use advanced debugging tools (such as SystemTap, kprobes, and eBPF-based tools), or access ftrace output files used by the Kernel Live Patching infrastructure.
- Amazon Linux 2 instances with 64-bit ARM (arm64) architecture are not supported.

Frequently asked questions

For frequently asked questions about Kernel Live Patching for Amazon Linux 2, see the [Amazon Linux 2 Kernel Live Patching FAQ](#).

User provided kernels

If you need a custom kernel on your Amazon EC2 instances, you can start with an AMI that is close to what you want, compile the custom kernel on your instance, and update the bootloader to point to the new kernel. This process varies depending on the virtualization type that your AMI uses. For more information, see [Linux AMI virtualization types \(p. 102\)](#).

Contents

- [HVM AMIs \(GRUB\) \(p. 193\)](#)
- [Paravirtual AMIs \(PV-GRUB\) \(p. 193\)](#)

HVM AMIs (GRUB)

HVM instance volumes are treated like actual physical disks. The boot process is similar to that of a bare metal operating system with a partitioned disk and bootloader, which enables it to work with all currently supported Linux distributions. The most common bootloader is GRUB or GRUB2.

By default, GRUB does not send its output to the instance console because it creates an extra boot delay. For more information, see [Instance console output \(p. 1278\)](#). If you are installing a custom kernel, you should consider enabling GRUB output.

You don't need to specify a fallback kernel, but we recommend that you have a fallback when you test a new kernel. GRUB can fall back to another kernel in the event that the new kernel fails. Having a fallback kernel enables the instance to boot even if the new kernel isn't found.

The legacy GRUB for Amazon Linux uses `/boot/grub/menu.lst`. GRUB2 for Amazon Linux 2 uses `/etc/default/grub`. For more information about updating the default kernel in the bootloader, see the documentation for your Linux distribution.

Paravirtual AMIs (PV-GRUB)

Amazon Machine Images that use paravirtual (PV) virtualization use a system called *PV-GRUB* during the boot process. PV-GRUB is a paravirtual bootloader that runs a patched version of GNU GRUB 0.97. When

you start an instance, PV-GRUB starts the boot process and then chain loads the kernel specified by your image's `menu.1st` file.

PV-GRUB understands standard `grub.conf` or `menu.1st` commands, which allows it to work with all currently supported Linux distributions. Older distributions such as Ubuntu 10.04 LTS, Oracle Enterprise Linux or CentOS 5.x require a special "ec2" or "xen" kernel package, while newer distributions include the required drivers in the default kernel package.

Most modern paravirtual AMIs use a PV-GRUB AKI by default (including all of the paravirtual Linux AMIs available in the Amazon EC2 Launch Wizard Quick Start menu), so there are no additional steps that you need to take to use a different kernel on your instance, provided that the kernel you want to use is compatible with your distribution. The best way to run a custom kernel on your instance is to start with an AMI that is close to what you want and then to compile the custom kernel on your instance and modify the `menu.1st` file to boot with that kernel.

You can verify that the kernel image for an AMI is a PV-GRUB AKI. Run the following [describe-images](#) command (substituting your kernel image ID) check whether the `Name` field starts with `pv-grub`:

```
aws ec2 describe-images --filters Name=image-id,Values=aki-880531cd
```

Contents

- [Limitations of PV-GRUB \(p. 194\)](#)
- [Configuring GRUB for paravirtual AMIs \(p. 194\)](#)
- [Amazon PV-GRUB Kernel Image IDs \(p. 195\)](#)
- [Updating PV-GRUB \(p. 197\)](#)

Limitations of PV-GRUB

PV-GRUB has the following limitations:

- You can't use the 64-bit version of PV-GRUB to start a 32-bit kernel or vice versa.
- You can't specify an Amazon ramdisk image (ARI) when using a PV-GRUB AKI.
- AWS has tested and verified that PV-GRUB works with these file system formats: EXT2, EXT3, EXT4, JFS, XFS, and ReiserFS. Other file system formats might not work.
- PV-GRUB can boot kernels compressed using the gzip, bzip2, lzo, and xz compression formats.
- Cluster AMIs don't support or need PV-GRUB, because they use full hardware virtualization (HVM). While paravirtual instances use PV-GRUB to boot, HVM instance volumes are treated like actual disks, and the boot process is similar to the boot process of a bare metal operating system with a partitioned disk and bootloader.
- PV-GRUB versions 1.03 and earlier don't support GPT partitioning; they support MBR partitioning only.
- If you plan to use a logical volume manager (LVM) with Amazon EBS volumes, you need a separate boot partition outside of the LVM. Then you can create logical volumes with the LVM.

Configuring GRUB for paravirtual AMIs

To boot PV-GRUB, a GRUB `menu.1st` file must exist in the image; the most common location for this file is `/boot/grub/menu.1st`.

The following is an example of a `menu.1st` configuration file for booting an AMI with a PV-GRUB AKI. In this example, there are two kernel entries to choose from: Amazon Linux 2018.03 (the original kernel for this AMI), and Vanilla Linux 4.16.4 (a newer version of the Vanilla Linux kernel from <https://www.kernel.org/>). The Vanilla entry was copied from the original entry for this AMI, and the `kernel`

and `initrd` paths were updated to the new locations. The `default 0` parameter points the bootloader to the first entry it sees (in this case, the Vanilla entry), and the `fallback 1` parameter points the bootloader to the next entry if there is a problem booting the first.

```
default 0
fallback 1
timeout 0
hiddenmenu

title Vanilla Linux 4.16.4
root (hd0)
kernel /boot/vmlinuz-4.16.4 root=LABEL=/ console=hvc0
initrd /boot/initrd.img-4.16.4

title Amazon Linux 2018.03 (4.14.26-46.32.amzn1.x86_64)
root (hd0)
kernel /boot/vmlinuz-4.14.26-46.32.amzn1.x86_64 root=LABEL=/ console=hvc0
initrd /boot/initramfs-4.14.26-46.32.amzn1.x86_64.img
```

You don't need to specify a fallback kernel in your `menu.lst` file, but we recommend that you have a fallback when you test a new kernel. PV-GRUB can fall back to another kernel in the event that the new kernel fails. Having a fallback kernel allows the instance to boot even if the new kernel isn't found.

PV-GRUB checks the following locations for `menu.lst`, using the first one it finds:

- `(hd0)/boot/grub`
- `(hd0,0)/boot/grub`
- `(hd0,0)/grub`
- `(hd0,1)/boot/grub`
- `(hd0,1)/grub`
- `(hd0,2)/boot/grub`
- `(hd0,2)/grub`
- `(hd0,3)/boot/grub`
- `(hd0,3)/grub`

Note that PV-GRUB 1.03 and earlier only check one of the first two locations in this list.

Amazon PV-GRUB Kernel Image IDs

PV-GRUB AKIs are available in all Amazon EC2 regions. There are AKIs for both 32-bit and 64-bit architecture types. Most modern AMIs use a PV-GRUB AKI by default.

We recommend that you always use the latest version of the PV-GRUB AKI, as not all versions of the PV-GRUB AKI are compatible with all instance types. Use the following [describe-images](#) command to get a list of the PV-GRUB AKIs for the current region:

```
aws ec2 describe-images --owners amazon --filters Name=name,Values=pv-grub-* .gz
```

PV-GRUB is the only AKI available in the `ap-southeast-2` Region. You should verify that any AMI you want to copy to this Region is using a version of PV-GRUB that is available in this Region.

The following are the current AKI IDs for each Region. Register new AMIs using an `hd0` AKI.

Note

We continue to provide `hd00` AKIs for backward compatibility in Regions where they were previously available.

ap-northeast-1, Asia Pacific (Tokyo)

Image ID	Image Name
aki-f975a998	pv-grub-hd0_1.05-i386.gz
aki-7077ab11	pv-grub-hd0_1.05-x86_64.gz

ap-southeast-1, Asia Pacific (Singapore) Region

Image ID	Image Name
aki-17a40074	pv-grub-hd0_1.05-i386.gz
aki-73a50110	pv-grub-hd0_1.05-x86_64.gz

ap-southeast-2, Asia Pacific (Sydney)

Image ID	Image Name
aki-ba5665d9	pv-grub-hd0_1.05-i386.gz
aki-66506305	pv-grub-hd0_1.05-x86_64.gz

eu-central-1, Europe (Frankfurt)

Image ID	Image Name
aki-1419e57b	pv-grub-hd0_1.05-i386.gz
aki-931fe3fc	pv-grub-hd0_1.05-x86_64.gz

eu-west-1, Europe (Ireland)

Image ID	Image Name
aki-1c9fd86f	pv-grub-hd0_1.05-i386.gz
aki-dc9ed9af	pv-grub-hd0_1.05-x86_64.gz

sa-east-1, South America (São Paulo)

Image ID	Image Name
aki-7cd34110	pv-grub-hd0_1.05-i386.gz
aki-912fbcf9	pv-grub-hd0_1.05-x86_64.gz

us-east-1, US East (N. Virginia)

Image ID	Image Name
aki-04206613	pv-grub-hd0_1.05-i386.gz

Image ID	Image Name
aki-5c21674b	pv-grub-hd0_1.05-x86_64.gz

us-gov-west-1, AWS GovCloud (US-West)

Image ID	Image Name
aki-5ee9573f	pv-grub-hd0_1.05-i386.gz
aki-9ee55bff	pv-grub-hd0_1.05-x86_64.gz

us-west-1, US West (N. California)

Image ID	Image Name
aki-43cf8123	pv-grub-hd0_1.05-i386.gz
aki-59cc8239	pv-grub-hd0_1.05-x86_64.gz

us-west-2, US West (Oregon)

Image ID	Image Name
aki-7a69931a	pv-grub-hd0_1.05-i386.gz
aki-70cb0e10	pv-grub-hd0_1.05-x86_64.gz

Updating PV-GRUB

We recommend that you always use the latest version of the PV-GRUB AKI, as not all versions of the PV-GRUB AKI are compatible with all instance types. Also, older versions of PV-GRUB are not available in all regions, so if you copy an AMI that uses an older version to a Region that does not support that version, you will be unable to boot instances launched from that AMI until you update the kernel image. Use the following procedures to check your instance's version of PV-GRUB and update it if necessary.

To check your PV-GRUB version

- Find the kernel ID for your instance.

```
aws ec2 describe-instance-attribute --instance-id instance_id --attribute kernel --region region

{
    "InstanceId": "instance_id",
    "KernelId": "aki-70cb0e10"
}
```

The kernel ID for this instance is aki-70cb0e10.

- View the version information of that kernel ID.

```
aws ec2 describe-images --image-ids aki-70cb0e10 --region region
```

```
{
```

```
"Images": [  
    {  
        "VirtualizationType": "paravirtual",  
        "Name": "pv-grub-hd0_1.05-x86_64.gz",  
        "..."  
        "Description": "PV-GRUB release 1.05, 64-bit"  
    }  
]
```

This kernel image is PV-GRUB 1.05. If your PV-GRUB version is not the newest version (as shown in [Amazon PV-GRUB Kernel Image IDs \(p. 195\)](#)), you should update it using the following procedure.

To update your PV-GRUB version

If your instance is using an older version of PV-GRUB, you should update it to the latest version.

1. Identify the latest PV-GRUB AKI for your Region and processor architecture from [Amazon PV-GRUB Kernel Image IDs \(p. 195\)](#).
2. Stop your instance. Your instance must be stopped to modify the kernel image used.

```
aws ec2 stop-instances --instance-ids instance_id --region region
```

3. Modify the kernel image used for your instance.

```
aws ec2 modify-instance-attribute --instance-id instance_id --kernel kernel_id --region region
```

4. Restart your instance.

```
aws ec2 start-instances --instance-ids instance_id --region region
```

Using the MATE desktop environment provided with Amazon Linux 2

The [MATE desktop environment](#) is pre-installed and pre-configured in the AMI with the following description: Amazon Linux 2 with .Net Core, PowerShell, Mono, and MATE Desktop Environment. The environment provides an intuitive graphical user interface for administering Amazon Linux 2 instances without using the command line. The interface uses graphical representations, such as icons, windows, toolbars, folders, wallpapers, and desktop widgets. Built-in, GUI-based tools are available to perform common tasks. For example, there are tools for adding and removing software, applying updates, organizing files, launching programs, and monitoring system health.

Complete the following procedure to use the MATE desktop environment.

To configure Remote Desktop Protocol (RDP) connections and set up a password

1. Use the following `describe-images` command to get the ID of the AMI for Amazon Linux 2 that includes MATE in the AMI name.

```
aws ec2 describe-images --filters "Name=name,Values=amzn2*MATE*" --query 'Images[*].ImageId' --output text
```

The following is example output:

```
ami-0abcdef1234567890
```

2. Launch an EC2 instance with the AMI that you located in the previous step. Configure the security group to allow for inbound TCP traffic to port 3389. For more information about configuring security groups, see [Security groups for your VPC](#). This configuration enables you to use an RDP client to connect to the instance.
3. Connect to the instance using [SSH](#). Run the following command to set the password for ec2-user.

```
[ec2-user ~]$ sudo passwd ec2-user
```

4. Open an RDP client on the computer from which you will connect to the instance (for example, Remote Desktop Connection on a computer running Microsoft Windows). Enter ec2-user as the user name and enter the password that you set in the previous step.

To disable the MATE Desktop Environment on your EC2 instance

You can turn off the GUI environment at any time by running one of the following commands.

```
[ec2-user ~]$ sudo systemctl disable xrdp
```

```
[ec2-user ~]$ sudo systemctl stop xrdp
```

To enable the MATE Desktop Environment on your EC2 instance

To turn the GUI back on, you can run one of the following commands.

```
[ec2-user ~]$ sudo systemctl enable xrdp
```

```
[ec2-user ~]$ sudo systemctl start xrdp
```

Amazon EC2 instances

If you're new to Amazon EC2, see the following topics to get started:

- [What is Amazon EC2? \(p. 1\)](#)
- [Setting up with Amazon EC2 \(p. 26\)](#)
- [Tutorial: Getting started with Amazon EC2 Linux instances \(p. 30\)](#)
- [Instance lifecycle \(p. 493\)](#)

Before you launch a production environment, you need to answer the following questions.

Q. What instance type best meets my needs?

Amazon EC2 provides different instance types to enable you to choose the CPU, memory, storage, and networking capacity that you need to run your applications. For more information, see [Instance types \(p. 200\)](#).

Q. What purchasing option best meets my needs?

Amazon EC2 supports On-Demand Instances (the default), Spot Instances, and Reserved Instances. For more information, see [Instance purchasing options \(p. 303\)](#).

Q. Which type of root volume meets my needs?

Each instance is backed by Amazon EBS or backed by instance store. Select an AMI based on which type of root volume you need. For more information, see [Storage for the root device \(p. 100\)](#).

Q. Can I remotely manage a fleet of EC2 instances and machines in my hybrid environment?

AWS Systems Manager enables you to remotely and securely manage the configuration of your Amazon EC2 instances, and your on-premises instances and virtual machines (VMs) in hybrid environments, including VMs from other cloud providers. For more information, see the [AWS Systems Manager User Guide](#).

Instance types

When you launch an instance, the *instance type* that you specify determines the hardware of the host computer used for your instance. Each instance type offers different compute, memory, and storage capabilities and are grouped in instance families based on these capabilities. Select an instance type based on the requirements of the application or software that you plan to run on your instance.

Amazon EC2 provides each instance with a consistent and predictable amount of CPU capacity, regardless of its underlying hardware.

Amazon EC2 dedicates some resources of the host computer, such as CPU, memory, and instance storage, to a particular instance. Amazon EC2 shares other resources of the host computer, such as the network and the disk subsystem, among instances. If each instance on a host computer tries to use as much of one of these shared resources as possible, each receives an equal share of that resource. However, when a resource is underused, an instance can consume a higher share of that resource while it's available.

Each instance type provides higher or lower minimum performance from a shared resource. For example, instance types with high I/O performance have a larger allocation of shared resources. Allocating a larger share of shared resources also reduces the variance of I/O performance. For most applications, moderate I/O performance is more than enough. However, for applications that require greater or more consistent I/O performance, consider an instance type with higher I/O performance.

Contents

- [Available instance types \(p. 201\)](#)
- [Hardware specifications \(p. 204\)](#)
- [AMI virtualization types \(p. 204\)](#)
- [Instances built on the Nitro System \(p. 205\)](#)
- [Networking and storage features \(p. 205\)](#)
- [Instance limits \(p. 208\)](#)
- [General purpose instances \(p. 208\)](#)
- [Compute optimized instances \(p. 253\)](#)
- [Memory optimized instances \(p. 260\)](#)
- [Storage optimized instances \(p. 272\)](#)
- [Linux accelerated computing instances \(p. 278\)](#)
- [Finding an Amazon EC2 instance type \(p. 293\)](#)
- [Changing the instance type \(p. 294\)](#)
- [Getting recommendations for an instance type \(p. 300\)](#)

Available instance types

Amazon EC2 provides a wide selection of instance types optimized for different use cases. To determine which instance types meet your requirements, such as supported Regions, compute resources, or storage resources, see [Finding an Amazon EC2 instance type \(p. 293\)](#).

Current generation instances

For the best performance, we recommend that you use the following instance types when you launch new instances. For more information, see [Amazon EC2 Instance Types](#).

Type	Sizes	Use case
A1	a1.medium a1.large a1.xlarge a1.2xlarge a1.4xlarge a1.metal	General purpose (p. 208)
C4	c4.large c4.xlarge c4.2xlarge c4.4xlarge c4.8xlarge	Compute optimized (p. 253)
C5	c5.large c5.xlarge c5.2xlarge c5.4xlarge c5.9xlarge c5.12xlarge c5.18xlarge c5.24xlarge c5.metal	Compute optimized (p. 253)
C5a	c5a.large c5a.xlarge c5a.2xlarge c5a.4xlarge c5a.8xlarge c5a.12xlarge c5a.16xlarge c5a.24xlarge	Compute optimized (p. 253)
C5ad	c5ad.large c5ad.xlarge c5ad.2xlarge c5ad.4xlarge c5ad.8xlarge c5ad.12xlarge c5ad.16xlarge c5ad.24xlarge	Compute optimized (p. 253)
C5d	c5d.large c5d.xlarge c5d.2xlarge c5d.4xlarge c5d.9xlarge c5d.12xlarge c5d.18xlarge c5d.24xlarge c5d.metal	Compute optimized (p. 253)
C5n	c5n.large c5n.xlarge c5n.2xlarge c5n.4xlarge c5n.9xlarge c5n.18xlarge c5n.metal	Compute optimized (p. 253)
C6g	c6g.medium c6g.large c6g.xlarge c6g.2xlarge c6g.4xlarge c6g.8xlarge c6g.12xlarge c6g.16xlarge c6g.metal	Compute optimized (p. 253)

Type	Sizes	Use case
C6gd	c6gd.medium c6gd.large c6gd.xlarge c6gd.2xlarge c6gd.4xlarge c6gd.8xlarge c6gd.12xlarge c6gd.16xlarge c6gd.metal	Compute optimized (p. 253)
D2	d2.xlarge d2.2xlarge d2.4xlarge d2.8xlarge	Storage optimized (p. 272)
F1	f1.2xlarge f1.4xlarge f1.16xlarge	Accelerated computing (p. 278)
G3	g3s.xlarge g3.4xlarge g3.8xlarge g3.16xlarge	Accelerated computing (p. 278)
G4	g4dn.xlarge g4dn.2xlarge g4dn.4xlarge g4dn.8xlarge g4dn.12xlarge g4dn.16xlarge g4dn.metal	Accelerated computing (p. 278)
H1	h1.2xlarge h1.4xlarge h1.8xlarge h1.16xlarge	Storage optimized (p. 272)
I3	i3.large i3.xlarge i3.2xlarge i3.4xlarge i3.8xlarge i3.16xlarge i3.metal	Storage optimized (p. 272)
I3en	i3en.large i3en.xlarge i3en.2xlarge i3en.3xlarge i3en.6xlarge i3en.12xlarge i3en.24xlarge i3en.metal	Storage optimized (p. 272)
Inf1	inf1.xlarge inf1.2xlarge inf1.6xlarge inf1.24xlarge	Accelerated computing (p. 278)
M4	m4.large m4.xlarge m4.2xlarge m4.4xlarge m4.10xlarge m4.16xlarge	General purpose (p. 208)
M5	m5.large m5.xlarge m5.2xlarge m5.4xlarge m5.8xlarge m5.12xlarge m5.16xlarge m5.24xlarge m5.metal	General purpose (p. 208)
M5a	m5a.large m5a.xlarge m5a.2xlarge m5a.4xlarge m5a.8xlarge m5a.12xlarge m5a.16xlarge m5a.24xlarge	General purpose (p. 208)
M5ad	m5ad.large m5ad.xlarge m5ad.2xlarge m5ad.4xlarge m5ad.8xlarge m5ad.12xlarge m5ad.16xlarge m5ad.24xlarge	General purpose (p. 208)
M5d	m5d.large m5d.xlarge m5d.2xlarge m5d.4xlarge m5d.8xlarge m5d.12xlarge m5d.16xlarge m5d.24xlarge m5d.metal	General purpose (p. 208)
M5dn	m5dn.large m5dn.xlarge m5dn.2xlarge m5dn.4xlarge m5dn.8xlarge m5dn.12xlarge m5dn.16xlarge m5dn.24xlarge	General purpose (p. 208)
M5n	m5n.large m5n.xlarge m5n.2xlarge m5n.4xlarge m5n.8xlarge m5n.12xlarge m5n.16xlarge m5n.24xlarge	General purpose (p. 208)
M6g	m6g.medium m6g.large m6g.xlarge m6g.2xlarge m6g.4xlarge m6g.8xlarge m6g.12xlarge m6g.16xlarge m6g.metal	General purpose (p. 208)
M6gd	m6gd.medium m6gd.large m6gd.xlarge m6gd.2xlarge m6gd.4xlarge m6gd.8xlarge m6gd.12xlarge m6gd.16xlarge m6gd.metal	General purpose (p. 208)
P2	p2.xlarge p2.8xlarge p2.16xlarge	Accelerated computing (p. 278)

Type	Sizes	Use case
P3	p3.2xlarge p3.8xlarge p3.16xlarge	Accelerated computing (p. 278)
P3dn	p3dn.24xlarge	Accelerated computing (p. 278)
R4	r4.large r4.xlarge r4.2xlarge r4.4xlarge r4.8xlarge r4.16xlarge	Memory optimized (p. 260)
R5	r5.large r5.xlarge r5.2xlarge r5.4xlarge r5.8xlarge r5.12xlarge r5.16xlarge r5.24xlarge r5.metal	Memory optimized (p. 260)
R5a	r5a.large r5a.xlarge r5a.2xlarge r5a.4xlarge r5a.8xlarge r5a.12xlarge r5a.16xlarge r5a.24xlarge	Memory optimized (p. 260)
R5ad	r5ad.large r5ad.xlarge r5ad.2xlarge r5ad.4xlarge r5ad.8xlarge r5ad.12xlarge r5ad.16xlarge r5ad.24xlarge	Memory optimized (p. 260)
R5d	r5d.large r5d.xlarge r5d.2xlarge r5d.4xlarge r5d.8xlarge r5d.12xlarge r5d.16xlarge r5d.24xlarge r5d.metal	Memory optimized (p. 260)
R5dn	r5dn.large r5dn.xlarge r5dn.2xlarge r5dn.4xlarge r5dn.8xlarge r5dn.12xlarge r5dn.16xlarge r5dn.24xlarge	Memory optimized (p. 260)
R5n	r5n.large r5n.xlarge r5n.2xlarge r5n.4xlarge r5n.8xlarge r5n.12xlarge r5n.16xlarge r5n.24xlarge	Memory optimized (p. 260)
R6g	r6g.medium r6g.large r6g.xlarge r6g.2xlarge r6g.4xlarge r6g.8xlarge r6g.12xlarge r6g.16xlarge r6g.metal	Memory optimized (p. 260)
R6gd	r6gd.medium r6gd.large r6gd.xlarge r6gd.2xlarge r6gd.4xlarge r6gd.8xlarge r6gd.12xlarge r6gd.16xlarge r6gd.metal	Memory optimized (p. 260)
T2	t2.nano t2.micro t2.small t2.medium t2.large t2.xlarge t2.2xlarge	General purpose (p. 208)
T3	t3.nano t3.micro t3.small t3.medium t3.large t3.xlarge t3.2xlarge	General purpose (p. 208)
T3a	t3a.nano t3a.micro t3a.small t3a.medium t3a.large t3a.xlarge t3a.2xlarge	General purpose (p. 208)
T4g	t4g.nano t4g.micro t4g.small t4g.medium t4g.large t4g.xlarge t4g.2xlarge	General purpose (p. 208)
u-xtb1	u-6tb1.metal u-9tb1.metal u-12tb1.metal u-18tb1.metal u-24tb1.metal	Memory optimized (p. 260)
X1	x1.16xlarge x1.32xlarge	Memory optimized (p. 260)
X1e	x1e.xlarge x1e.2xlarge x1e.4xlarge x1e.8xlarge x1e.16xlarge x1e.32xlarge	Memory optimized (p. 260)
z1d	z1d.large z1d.xlarge z1d.2xlarge z1d.3xlarge z1d.6xlarge z1d.12xlarge z1d.metal	Memory optimized (p. 260)

Previous generation instances

Amazon Web Services offers previous generation instance types for users who have optimized their applications around them and have yet to upgrade. We encourage you to use current generation instance types to get the best performance, but we continue to support the following previous generation instance types. For more information about which current generation instance type would be a suitable upgrade, see [Previous Generation Instances](#).

Type	Sizes
C1	c1.medium c1.xlarge
C3	c3.large c3.xlarge c3.2xlarge c3.4xlarge c3.8xlarge
G2	g2.2xlarge g2.8xlarge
I2	i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge
M1	m1.small m1.medium m1.large m1.xlarge
M2	m2.xlarge m2.2xlarge m2.4xlarge
M3	m3.medium m3.large m3.xlarge m3.2xlarge
R3	r3.large r3.xlarge r3.2xlarge r3.4xlarge r3.8xlarge
T1	t1.micro

Hardware specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instance Types](#).

To determine which instance type best meets your needs, we recommend that you launch an instance and use your own benchmark application. Because you pay by the instance second, it's convenient and inexpensive to test multiple instance types before making a decision.

If your needs change, even after you make a decision, you can resize your instance later. For more information, see [Changing the instance type \(p. 294\)](#).

Note

Amazon EC2 instances typically run on 64-bit virtual Intel processors as specified in the instance type product pages. For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instance Types](#). However, confusion may result from industry naming conventions for 64-bit CPUs. Chip manufacturer Advanced Micro Devices (AMD) introduced the first commercially successful 64-bit architecture based on the Intel x86 instruction set. Consequently, the architecture is widely referred to as AMD64 regardless of the chip manufacturer. Windows and several Linux distributions follow this practice. This explains why the internal system information on an Ubuntu or Windows EC2 instance displays the CPU architecture as AMD64 even though the instances are running on Intel hardware.

AMI virtualization types

The virtualization type of your instance is determined by the AMI that you use to launch it. Current generation instance types support hardware virtual machine (HVM) only. Some previous generation instance types support paravirtual (PV) and some AWS regions support PV instances. For more information, see [Linux AMI virtualization types \(p. 102\)](#).

For best performance, we recommend that you use an HVM AMI. In addition, HVM AMIs are required to take advantage of enhanced networking. HVM virtualization uses hardware-assist technology provided by the AWS platform. With HVM virtualization, the guest VM runs as if it were on a native hardware platform, except that it still uses PV network and storage drivers for improved performance.

Instances built on the Nitro System

The Nitro System is a collection of AWS-built hardware and software components that enable high performance, high availability, and high security. In addition, the Nitro System provides bare metal capabilities that eliminate virtualization overhead and support workloads that require full access to host hardware. For more information, see [AWS Nitro System](#).

Nitro components

The following components are part of the Nitro System:

- Nitro card
 - Local NVMe storage volumes
 - Networking hardware support
 - Management
 - Monitoring
 - Security
- Nitro security chip, integrated into the motherboard
- Nitro hypervisor - A lightweight hypervisor that manages memory and CPU allocation and delivers performance that is indistinguishable from bare metal for most workloads.

Instance types

The following instances are built on the Nitro System:

- Virtualized: A1, C5, C5a, C5ad, C5d, C5n, C6g, C6gd, G4, I3en, Inf1, M5, M5a, M5ad, M5d, M5dn, M5n, M6g, M6gd, p3dn.24xlarge, R5, R5a, R5ad, R5d, R5dn, R5n, R6g, R6gd, T3, T3a, T4g, and z1d
- Bare metal: a1.metal, c5.metal, c5d.metal, c5n.metal, c6g.metal, c6gd.metal, i3.metal, i3en.metal, m5.metal, m5d.metal, m6g.metal, m6gd.metal, r5.metal, r5d.metal, r6g.metal, r6gd.metal, u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, u-18tb1.metal, u-24tb1.metal, and z1d.metal

Learn more

For more information, see the following videos:

- [AWS re:Invent 2017: The Amazon EC2 Nitro System Architecture](#)
- [AWS re:Invent 2017: Amazon EC2 Bare Metal Instances](#)
- [AWS re:Invent 2019: Powering next-gen Amazon EC2: Deep dive into the Nitro system](#)
- [AWS re:Inforce 2019: Security Benefits of the Nitro Architecture](#)

Networking and storage features

When you select an instance type, this determines the networking and storage features that are available. To describe an instance type, use the [describe-instance-types](#) command.

Networking features

- IPv6 is supported on all current generation instance types and the C3, R3, and I2 previous generation instance types.
- To maximize the networking and bandwidth performance of your instance type, you can do the following:
 - Launch supported instance types into a cluster placement group to optimize your instances for high performance computing (HPC) applications. Instances in a common cluster placement group can benefit from high-bandwidth, low-latency networking. For more information, see [Placement groups \(p. 869\)](#).
 - Enable enhanced networking for supported current generation instance types to get significantly higher packet per second (PPS) performance, lower network jitter, and lower latencies. For more information, see [Enhanced networking on Linux \(p. 813\)](#).
- Current generation instance types that are enabled for enhanced networking have the following networking performance attributes:
 - Traffic within the same Region over private IPv4 or IPv6 can support 5 Gbps for single-flow traffic and up to 25 Gbps for multi-flow traffic (depending on the instance type).
 - Traffic to and from Amazon S3 buckets within the same Region over the public IP address space or through a VPC endpoint can use all available instance aggregate bandwidth.
 - The maximum transmission unit (MTU) supported varies across instance types. All Amazon EC2 instance types support standard Ethernet V2 1500 MTU frames. All current generation instances support 9001 MTU, or jumbo frames, and some previous generation instances support them as well. For more information, see [Network maximum transmission unit \(MTU\) for your EC2 instance \(p. 881\)](#).

Storage features

- Some instance types support EBS volumes and instance store volumes, while other instance types support only EBS volumes. Some instance types that support instance store volumes use solid state drives (SSD) to deliver very high random I/O performance. Some instance types support NVMe instance store volumes. Some instance types support NVMe EBS volumes. For more information, see [Amazon EBS and NVMe on Linux instances \(p. 1136\)](#) and [NVMe SSD volumes \(p. 1200\)](#).
- To obtain additional, dedicated capacity for Amazon EBS I/O, you can launch some instance types as EBS-optimized instances. Some instance types are EBS-optimized by default. For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#).

Summary of networking and storage features

The following table summarizes the networking and storage features supported by current generation instance types.

	EBS only	NVMe EBS	Instance store	Placement group	Enhanced networking
A1	Yes	Yes	No	Yes	ENI
C4	Yes	No	No	Yes	Intel 82599 VF
C5	Yes	Yes	No	Yes	ENI
C5a	Yes	Yes	No	Yes	ENI
C5ad	No	Yes	NVMe *	Yes	ENI
C5d	No	Yes	NVMe *	Yes	ENI

	EBS only	NVMe EBS	Instance store	Placement group	Enhanced networking
C5n	Yes	Yes	No	Yes	ENA
C6g	Yes	Yes	No	Yes	ENA
C6gd	No	Yes	NVME *	Yes	ENA
D2	No	No	HDD	Yes	Intel 82599 VF
F1	No	No	NVMe *	Yes	ENA
G3	Yes	No	No	Yes	ENA
G4	No	Yes	NVMe *	Yes	ENA
HS1	No	No	HDD *	Yes	ENA
I3	No	No	NVMe *	Yes	ENA
I3en	No	Yes	NVMe *	Yes	ENA
Inf1	Yes	Yes	No	Yes	ENA
M4	Yes	No	No	Yes	m4.16xlarge: ENA All other sizes: Intel 82599 VF
M5	Yes	Yes	No	Yes	ENA
M5a	Yes	Yes	No	Yes	ENA
M5ad	No	Yes	NVMe *	Yes	ENA
M5d	No	Yes	NVMe *	Yes	ENA
M5dn	No	Yes	NVMe *	Yes	ENA
M5n	Yes	Yes	No	Yes	ENA
M6g	Yes	Yes	No	Yes	ENA
M6gd	No	Yes	NVME *	Yes	ENA
P2	Yes	No	No	Yes	ENA
P3	Yes	No	No	Yes	ENA
P3dn	No	Yes	NVMe *	Yes	ENA
R4	Yes	No	No	Yes	ENA
R5	Yes	Yes	No	Yes	ENA
R5a	Yes	Yes	No	Yes	ENA
R5ad	No	Yes	NVMe *	Yes	ENA
R5d	No	Yes	NVMe *	Yes	ENA

	EBS only	NVMe EBS	Instance store	Placement group	Enhanced networking
R5dn	No	Yes	NVMe *	Yes	ENA
R5n	Yes	Yes	No	Yes	ENA
R6g	Yes	Yes	No	Yes	ENA
R6gd	No	Yes	NVME *	Yes	ENA
T2	Yes	No	No	No	No
T3	Yes	Yes	No	No	ENA
T3a	Yes	Yes	No	No	ENA
T4g	Yes	Yes	No	No	ENA
u-xtb1.metal	Yes	Yes	No	No	ENA
X1	No	No	SSD *	Yes	ENA
X1e	No	No	SSD *	Yes	ENA
z1d	No	Yes	NVMe *	Yes	ENA

* The root device volume must be an Amazon EBS volume.

The following table summarizes the networking and storage features supported by previous generation instance types.

	Instance store	Placement group	Enhanced networking
C3	SSD	Yes	Intel 82599 VF
G2	SSD	Yes	No
I2	SSD	Yes	Intel 82599 VF
M3	SSD	No	No
R3	SSD	Yes	Intel 82599 VF

Instance limits

There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some instance types.

For more information about the default limits, see [How many instances can I run in Amazon EC2?](#)

For more information about viewing your current limits or requesting an increase in your current limits, see [Amazon EC2 service quotas \(p. 1241\)](#).

General purpose instances

General purpose instances provide a balance of compute, memory, and networking resources, and can be used for a wide range of workloads.

A1 instances

These instances are ideally suited for scale-out workloads that are supported by the Arm ecosystem. These instances are well-suited for the following:

- Web servers
- Containerized microservices

Bare metal instances, such as `a1.metal`, provide your applications with direct access to physical resources of the host server, such as processors and memory.

For more information, see [AWS Graviton Processor](#) and [Amazon EC2 A1 Instances](#).

M5 and M5a instances

These instances provide an ideal cloud infrastructure, offering a balance of compute, memory, and networking resources for a broad range of applications that are deployed in the cloud. They are well-suited for the following:

- Small and midsize databases
- Data processing tasks that require additional memory
- Caching fleets
- Backend servers for SAP, Microsoft SharePoint, cluster computing, and other enterprise applications

For more information, see [Amazon EC2 M5 and M5a Instances](#).

Bare metal instances, such as `m5.metal`, provide your applications with direct access to physical resources of the host server, such as processors and memory. These instances are well suited for the following:

- Workloads that require access to low-level hardware features (for example, Intel VT) that are not available or fully supported in virtualized environments
- Applications that require a non-virtualized environment for licensing or support

M6g and M6gd instances

These instances are powered by AWS Graviton2 processors and deliver balanced compute, memory, and networking for a broad range a general purpose workloads. They are well suited for the following:

- Application servers
- Microservices
- Gaming servers
- Midsize data stores
- Caching fleets

Bare metal instances, such as `m6g.metal`, provide your applications with direct access to physical resources of the host server, such as processors and memory.

For more information, see [Amazon EC2 M6g Instances](#).

T2, T3, T3a, and T4g instances

These instances provide a baseline level of CPU performance with the ability to burst to a higher level when required by your workload. An Unlimited instance can sustain high CPU performance for any

period of time whenever required. For more information, see [Burstable performance instances \(p. 219\)](#). These instances are well-suited for the following:

- Websites and web applications
- Code repositories
- Development, build, test, and staging environments
- Microservices

For more information, see [Amazon EC2 T2 Instances](#), [Amazon EC2 T3 Instances](#), and [Amazon EC2 T4g Instances](#).

Contents

- [Hardware specifications \(p. 210\)](#)
- [Instance performance \(p. 214\)](#)
- [Network performance \(p. 214\)](#)
- [SSD I/O performance \(p. 215\)](#)
- [Instance features \(p. 217\)](#)
- [Release notes \(p. 218\)](#)
- [Burstable performance instances \(p. 219\)](#)

Hardware specifications

The following is a summary of the hardware specifications for general purpose instances.

Instance type	Default vCPUs	Memory (GiB)
a1.medium	1	2
a1.large	2	4
a1.xlarge	4	8
a1.2xlarge	8	16
a1.4xlarge	16	32
a1.metal	16	32
m4.large	2	8
m4.xlarge	4	16
m4.2xlarge	8	32
m4.4xlarge	16	64
m4.10xlarge	40	160
m4.16xlarge	64	256
m5.large	2	8
m5.xlarge	4	16
m5.2xlarge	8	32

Instance type	Default vCPUs	Memory (GiB)
m5.4xlarge	16	64
m5.8xlarge	32	128
m5.12xlarge	48	192
m5.16xlarge	64	256
m5.24xlarge	96	384
m5.metal	96	384
m5a.large	2	8
m5a.xlarge	4	16
m5a.2xlarge	8	32
m5a.4xlarge	16	64
m5a.8xlarge	32	128
m5a.12xlarge	48	192
m5a.16xlarge	64	256
m5a.24xlarge	96	384
m5ad.large	2	8
m5ad.xlarge	4	16
m5ad.2xlarge	8	32
m5ad.4xlarge	16	64
m5ad.8xlarge	32	128
m5ad.12xlarge	48	192
m5ad.16xlarge	64	256
m5ad.24xlarge	96	384
m5d.large	2	8
m5d.xlarge	4	16
m5d.2xlarge	8	32
m5d.4xlarge	16	64
m5d.8xlarge	32	128
m5d.12xlarge	48	192
m5d.16xlarge	64	256
m5d.24xlarge	96	384
m5d.metal	96	384

Instance type	Default vCPUs	Memory (GiB)
m5dn.large	2	8
m5dn.xlarge	4	16
m5dn.2xlarge	8	32
m5dn.4xlarge	16	64
m5dn.8xlarge	32	128
m5dn.12xlarge	48	192
m5dn.16xlarge	64	256
m5dn.24xlarge	96	384
m5n.large	2	8
m5n.xlarge	4	16
m5n.2xlarge	8	32
m5n.4xlarge	16	64
m5n.8xlarge	32	128
m5n.12xlarge	48	192
m5n.16xlarge	64	256
m5n.24xlarge	96	384
m6g.medium	1	4
m6g.large	2	8
m6g.xlarge	4	16
m6g.2xlarge	8	32
m6g.4xlarge	16	64
m6g.8xlarge	32	128
m6g.12xlarge	48	192
m6g.16xlarge	64	256
m6g.metal	64	256
m6gd.medium	1	4
m6gd.large	2	8
m6gd.xlarge	4	16
m6gd.2xlarge	8	32
m6gd.4xlarge	16	64
m6gd.8xlarge	32	128

Instance type	Default vCPUs	Memory (GiB)
m6gd.12xlarge	48	192
m6gd.16xlarge	64	256
m6gd.metal	64	256
t2.nano	1	0.5
t2.micro	1	1
t2.small	1	2
t2.medium	2	4
t2.large	2	8
t2.xlarge	4	16
t2.2xlarge	8	32
t3.nano	2	0.5
t3.micro	2	1
t3.small	2	2
t3.medium	2	4
t3.large	2	8
t3.xlarge	4	16
t3.2xlarge	8	32
t3a.nano	2	0.5
t3a.micro	2	1
t3a.small	2	2
t3a.medium	2	4
t3a.large	2	8
t3a.xlarge	4	16
t3a.2xlarge	8	32
t4g.nano	2	0.5
t4g.micro	2	1
t4g.small	2	2
t4g.medium	2	4
t4g.large	2	8
t4g.xlarge	4	16
t4g.2xlarge	8	32

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instance Types](#).

For more information about specifying CPU options, see [Optimizing CPU options \(p. 632\)](#).

Instance performance

EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your instance. Some general purpose instances are EBS-optimized by default at no additional cost. For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#).

Some general purpose instance types provide the ability to control processor C-states and P-states on Linux. C-states control the sleep levels that a core can enter when it is inactive, while P-states control the desired performance (in CPU frequency) from a core. For more information, see [Processor state control for your EC2 instance \(p. 622\)](#).

Network performance

You can enable enhanced networking on supported instance types to provide lower latencies, lower network jitter, and higher packet-per-second (PPS) performance. Most applications do not consistently need a high level of network performance, but can benefit from access to increased bandwidth when they send or receive data. For more information, see [Enhanced networking on Linux \(p. 813\)](#).

The following is a summary of network performance for general purpose instances that support enhanced networking.

Instance type	Network performance	Enhanced networking
t2.nano t2.micro t2.small t2.medium t2.large t2.xlarge t2.2xlarge	Up to 1 Gbps	Not supported
t3.nano t3.micro t3.small t3.medium t3.large t3.xlarge t3.2xlarge t3a.nano t3a.micro t3a.small t3a.medium t3a.large t3a.xlarge t3a.2xlarge t4g.nano t4g.micro t4g.small t4g.medium t4g.large t4g.xlarge t4g.2xlarge	Up to 5 Gbps †	ENAs (p. 814)
m4.large	Moderate	Intel 82599 VF (p. 826)
m4.xlarge m4.2xlarge m4.4xlarge	High	Intel 82599 VF (p. 826)
a1.4xlarge and smaller a1.metal m5.4xlarge and smaller m5a.8xlarge and smaller m5ad.8xlarge and smaller m5d.4xlarge and smaller m6g.4xlarge and	Up to 10 Gbps †	ENAs (p. 814)

Instance type	Network performance	Enhanced networking
smaller m6gd.4xlarge and smaller		
m4.10xlarge	10 Gbps	Intel 82599 VF (p. 826)
m5.8xlarge m5a.12xlarge m5ad.12xlarge m5d.8xlarge	10 Gbps	ENAv (p. 814)
m5.12xlarge m5a.16xlarge m5ad.16xlarge m5d.12xlarge m6g.8xlarge m6gd.8xlarge	12 Gbps	ENAv (p. 814)
m5.16xlarge m5a.24xlarge m5ad.24xlarge m5d.16xlarge m6g.12xlarge m6gd.12xlarge	20 Gbps	ENAv (p. 814)
m5dn.4xlarge and smaller m5n.4xlarge and smaller	Up to 25 Gbps †	ENAv (p. 814)
m4.16xlarge m5.24xlarge m5.metal m5d.24xlarge m5d.metal m5dn.8xlarge m5n.8xlarge m6g.16xlarge m6g.metal m6gd.16xlarge m6gd.metal	25 Gbps	ENAv (p. 814)
m5dn.12xlarge m5n.12xlarge	50 Gbps	ENAv (p. 814)
m5dn.16xlarge m5n.16xlarge	75 Gbps	ENAv (p. 814)
m5dn.24xlarge m5n.24xlarge	100 Gbps	ENAv (p. 814)

† These instances use a network I/O credit mechanism to allocate network bandwidth to instances based on average bandwidth utilization. They accrue credits when their bandwidth is below their baseline bandwidth, and can use these credits when they perform network data transfers. For more information, open a support case and ask about baseline bandwidth for the specific instance types that you are interested in.

SSD I/O performance

If you use a Linux AMI with kernel version 4.4 or later and use all the SSD-based instance store volumes available to your instance, you get the IOPS (4,096 byte block size) performance listed in the following table (at queue depth saturation). Otherwise, you get lower IOPS performance.

Instance Size	100% Random Read IOPS	Write IOPS
m5ad.large *	30,000	15,000
m5ad.xlarge *	59,000	29,000

Instance Size	100% Random Read IOPS	Write IOPS
m5ad.2xlarge *	117,000	57,000
m5ad.4xlarge *	234,000	114,000
m5ad.8xlarge	466,666	233,333
m5ad.12xlarge	700,000	340,000
m5ad.16xlarge	933,333	466,666
m5ad.24xlarge	1,400,000	680,000
m5d.large *	30,000	15,000
m5d.xlarge *	59,000	29,000
m5d.2xlarge *	117,000	57,000
m5d.4xlarge *	234,000	114,000
m5d.8xlarge	466,666	233,333
m5d.12xlarge	700,000	340,000
m5d.16xlarge	933,333	466,666
m5d.24xlarge	1,400,000	680,000
m5d.metal	1,400,000	680,000
m5dn.large *	30,000	15,000
m5dn.xlarge *	59,000	29,000
m5dn.2xlarge *	117,000	57,000
m5dn.4xlarge *	234,000	114,000
m5dn.8xlarge	466,666	233,333
m5dn.12xlarge	700,000	340,000
m5dn.16xlarge	933,333	466,666
m5dn.24xlarge	1,400,000	680,000
m6gd.medium	13,438	5,625
m6gd.large	26,875	11,250
m6gd.xlarge	53,750	22,500
m6gd.2xlarge	107,500	45,000
m6gd.4xlarge	215,000	90,000
m6gd.8xlarge	430,000	180,000
m6gd.12xlarge	645,000	270,000
m6gd.16xlarge	860,000	360,000

Instance Size	100% Random Read IOPS	Write IOPS
m6gd.metal	860,000	360,000

* For these instances, you can get up to the specified performance.

As you fill the SSD-based instance store volumes for your instance, the number of write IOPS that you can achieve decreases. This is due to the extra work the SSD controller must do to find available space, rewrite existing data, and erase unused space so that it can be rewritten. This process of garbage collection results in internal write amplification to the SSD, expressed as the ratio of SSD write operations to user write operations. This decrease in performance is even larger if the write operations are not in multiples of 4,096 bytes or not aligned to a 4,096-byte boundary. If you write a smaller amount of bytes or bytes that are not aligned, the SSD controller must read the surrounding data and store the result in a new location. This pattern results in significantly increased write amplification, increased latency, and dramatically reduced I/O performance.

SSD controllers can use several strategies to reduce the impact of write amplification. One such strategy is to reserve space in the SSD instance storage so that the controller can more efficiently manage the space available for write operations. This is called *over-provisioning*. The SSD-based instance store volumes provided to an instance don't have any space reserved for over-provisioning. To reduce write amplification, we recommend that you leave 10% of the volume unpartitioned so that the SSD controller can use it for over-provisioning. This decreases the storage that you can use, but increases performance even if the disk is close to full capacity.

For instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see [Instance store volume TRIM support \(p. 1201\)](#).

Instance features

The following is a summary of features for general purpose instances:

	EBS only	NVMe EBS	Instance store	Placement group
A1	Yes	Yes	No	Yes
M4	Yes	No	No	Yes
M5	Yes	Yes	No	Yes
M5a	Yes	Yes	No	Yes
M5ad	No	Yes	NVMe *	Yes
M5d	No	Yes	NVMe *	Yes
M5dn	No	Yes	NVMe *	Yes
M5n	Yes	Yes	No	Yes
M6g	Yes	Yes	No	Yes
M6gd	No	Yes	NVMe *	Yes
T2	Yes	No	No	No
T3	Yes	Yes	No	No

	EBS only	NVMe EBS	Instance store	Placement group
T3a	Yes	Yes	No	No
T4g	Yes	Yes	No	No

* The root device volume must be an Amazon EBS volume.

For more information, see the following:

- [Amazon EBS and NVMe on Linux instances \(p. 1136\)](#)
- [Amazon EC2 instance store \(p. 1188\)](#)
- [Placement groups \(p. 869\)](#)

Release notes

- M5, M5d, and T3 instances feature a 3.1 GHz Intel Xeon Platinum 8000 series processor from either the first generation (Skylake-SP) or second generation (Cascade Lake).
- M5a, M5ad, and T3a instances feature a 2.5 GHz AMD EPYC 7000 series processor.
- A1 instances feature a 2.3 GHz AWS Graviton processor based on 64-bit Arm architecture.
- M6g and M6gd instances feature an AWS Graviton2 processor based on 64-bit Arm architecture.
- T4g instances feature an AWS Graviton2 processor based on 64-bit Arm architecture.
- M4, M5, M5a, M5ad, M5d, t2.large and larger, and t3.large and larger, and t3a.large and larger instance types require 64-bit HVM AMIs. They have high-memory, and require a 64-bit operating system to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. In addition, you must use an HVM AMI to take advantage of enhanced networking.
- Instances built on the [Nitro System \(p. 205\)](#) have the following requirements:
 - [NVMe drivers \(p. 1136\)](#) must be installed
 - [Elastic Network Adapter \(ENA\) drivers \(p. 814\)](#) must be installed

The following Linux AMIs meet these requirements:

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (with `linux-aws` kernel) or later
- Red Hat Enterprise Linux 7.4 or later
- SUSE Linux Enterprise Server 12 SP2 or later
- CentOS 7.4.1708 or later
- FreeBSD 11.1 or later
- Debian GNU/Linux 9 or later
- Instances with an [AWS Graviton Processor](#) have the following requirements:
 - Use an AMI for the 64-bit Arm architecture.
 - Support booting through UEFI with ACPI tables and support ACPI hot-plug of PCI devices.

The following Linux AMIs meet these requirements:

- Amazon Linux 2 (64-bit Arm)
- Ubuntu 16.04 or later (64-bit Arm)
- Red Hat Enterprise Linux 8.0 or later (64-bit Arm)
- SUSE Linux Enterprise Server 15 or later (64-bit Arm)

- Debian 10 or later (64-bit Arm)
- Instances built on the Nitro System support a maximum of 28 attachments, including network interfaces, EBS volumes, and NVMe instance store volumes. For more information, see [Nitro System volume limits \(p. 1209\)](#).
- Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
- To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support. Amazon Linux 2 and the latest versions of the Amazon Linux AMI support PCIe native hotplug, but earlier versions do not. You must enable the following Linux kernel configuration options:

```
CONFIG_HOTPLUG_PCI_PCIE=y  
CONFIG_PCIEASPM=y
```

- Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
- Instances built on the Nitro System should have system-logind or acpid installed to support clean shutdown through API requests.
- There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see [How many instances can I run in Amazon EC2?](#) in the Amazon EC2 FAQ.

Burstable performance instances

Burstable performance instances are designed to provide a baseline level of CPU performance with the ability to burst to a higher level when required by your workload. Burstable performance instances are well suited for a wide range of general-purpose applications. Examples include microservices, low-latency interactive applications, small and medium databases, virtual desktops, development, build, and stage environments, code repositories, and product prototypes.

Burstable performance instances are the only instance types that use credits for CPU usage. For more information about instance pricing and additional hardware details, see [Amazon EC2 Pricing](#) and [Amazon EC2 Instance Types](#).

If your account is less than 12 months old, you can use a `t2.micro` instance for free (or a `t3.micro` instance in Regions where `t2.micro` is unavailable) within certain usage limits. For more information, see [AWS Free Tier](#).

Contents

- [Burstable performance instance requirements \(p. 219\)](#)
- [Best practices \(p. 220\)](#)
- [CPU credits and baseline utilization for burstable performance instances \(p. 220\)](#)
- [Unlimited mode for burstable performance instances \(p. 223\)](#)
- [Standard mode for burstable performance instances \(p. 231\)](#)
- [Working with burstable performance instances \(p. 245\)](#)
- [Monitoring your CPU credits \(p. 250\)](#)

Burstable performance instance requirements

The following are the requirements for these instances:

- The supported instance families are: T2, T3, T3a, and T4g.
- The supported purchasing options are: On-Demand Instances, Reserved Instances, Dedicated Instances, and Spot Instances. These instances are not supported as Scheduled Instances or on a Dedicated Host. For more information, see [Instance purchasing options \(p. 303\)](#).
- Ensure that the instance size you choose passes the minimum memory requirements of your operating system and applications. Operating systems with graphical user interfaces that consume significant memory and CPU resources (for example, Windows) might require a t2.micro or larger instance size for many use cases. As the memory and CPU requirements of your workload grow over time, you can scale to larger instance sizes of the same instance type, or another instance type.
- For additional requirements, see [General Purpose Instances Release Notes \(p. 218\)](#).

Best practices

Follow these best practices to get the maximum benefit from burstable performance instances.

- **Use a recommended AMI** – Use an AMI that provides the required drivers. For more information, see [Release notes \(p. 218\)](#).
- **Turn on instance recovery** – Create a CloudWatch alarm that monitors an EC2 instance and automatically recovers it if it becomes impaired for any reason. For more information, see [Adding recover actions to Amazon CloudWatch alarms \(p. 742\)](#).

CPU credits and baseline utilization for burstable performance instances

Traditional Amazon EC2 instance types provide fixed CPU utilization, while burstable performance instances provide a baseline level of CPU utilization with the ability to burst CPU utilization above the baseline level. The baseline utilization and ability to burst are governed by CPU credits.

A CPU credit provides for 100% utilization of a full CPU core for one minute. Other combinations of number of vCPUs, utilization, and time can also equate to one CPU credit. For example, one CPU credit is equal to one vCPU running at 50% utilization for two minutes, or two vCPUs running at 25% utilization for two minutes.

Contents

- [Earning CPU credits \(p. 220\)](#)
- [CPU credit earn rate \(p. 222\)](#)
- [CPU credit accrual limit \(p. 222\)](#)
- [Accrued CPU credits life span \(p. 223\)](#)
- [Baseline utilization \(p. 223\)](#)

Earning CPU credits

Each burstable performance instance continuously earns (at a millisecond-level resolution) a set rate of CPU credits per hour, depending on the instance size. The accounting process for whether credits are accrued or spent also happens at a millisecond-level resolution, so you don't have to worry about overspending CPU credits; a short burst of CPU uses a small fraction of a CPU credit.

If a burstable performance instance uses fewer CPU resources than is required for baseline utilization (such as when it is idle), the unspent CPU credits are accrued in the CPU credit balance. If a burstable performance instance needs to burst above the baseline utilization level, it spends the accrued credits. The more credits that a burstable performance instance has accrued, the more time it can burst beyond its baseline when more CPU utilization is needed.

The following table lists the burstable performance instance types, the rate at which CPU credits are earned per hour, the maximum number of earned CPU credits that an instance can accrue, the number of vCPUs per instance, and the baseline utilization as a percentage of a full core (using a single vCPU).

Instance type	CPU credits earned per hour	Maximum earned credits that can be accrued*	vCPUs	Baseline utilization per vCPU
T2				
t2.nano	3	72	1	5%
t2.micro	6	144	1	10%
t2.small	12	288	1	20%
t2.medium	24	576	2	20%**
t2.large	36	864	2	30%**
t2.xlarge	54	1296	4	22.5%**
t2.2xlarge	81.6	1958.4	8	17%**
T3				
t3.nano	6	144	2	5%**
t3.micro	12	288	2	10%**
t3.small	24	576	2	20%**
t3.medium	24	576	2	20%**
t3.large	36	864	2	30%**
t3.xlarge	96	2304	4	40%**
t3.2xlarge	192	4608	8	40%**
T3a				
t3a.nano	6	144	2	5%**
t3a.micro	12	288	2	10%**
t3a.small	24	576	2	20%**
t3a.medium	24	576	2	20%**
t3a.large	36	864	2	30%**
t3a.xlarge	96	2304	4	40%**
t3a.2xlarge	192	4608	8	40%**
T4g				
t4g.nano	6	144	2	5%**
t4g.micro	12	288	2	10%**

Instance type	CPU credits earned per hour	Maximum earned credits that can be accrued*	vCPUs	Baseline utilization per vCPU
t4g.small	24	576	2	20%**
t4g.medium	24	576	2	20%**
t4g.large	36	864	2	30%**
t4g.xlarge	96	2304	4	40%**
t4g.2xlarge	192	4608	8	40%**

* The number of credits that can be accrued is equivalent to the number of credits that can be earned in a 24-hour period.

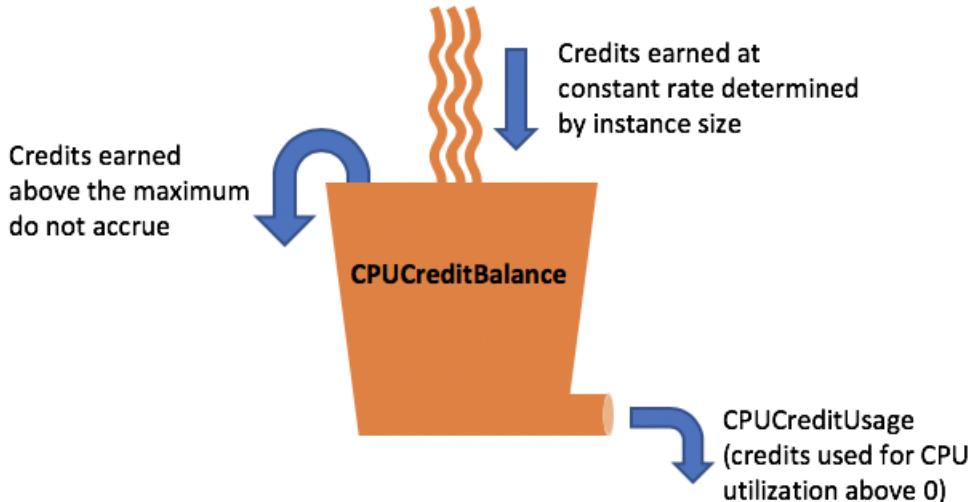
** The percentage baseline utilization in the table is per vCPU. In CloudWatch, CPU utilization is shown per vCPU. For example, the CPU utilization for a t3.large instance operating at the baseline level is shown as 30% in CloudWatch CPU metrics. For information about how to calculate the baseline utilization, see [Baseline utilization \(p. 223\)](#).

CPU credit earn rate

The number of CPU credits earned per hour is determined by the instance size. For example, a t3.nano earns six credits per hour, while a t3.small earns 24 credits per hour. The preceding table lists the credit earn rate for all instances.

CPU credit accrual limit

While earned credits never expire on a running instance, there is a limit to the number of earned credits that an instance can accrue. The limit is determined by the CPU credit balance limit. After the limit is reached, any new credits that are earned are discarded, as indicated by the following image. The full bucket indicates the CPU credit balance limit, and the spillover indicates the newly earned credits that exceed the limit.



The CPU credit balance limit differs for each instance size. For example, a `t3.micro` instance can accrue a maximum of 288 earned CPU credits in the CPU credit balance. The preceding table lists the maximum number of earned credits that each instance can accrue.

T2 Standard instances also earn launch credits. Launch credits do not count towards the CPU credit balance limit. If a T2 instance has not spent its launch credits, and remains idle over a 24-hour period while accruing earned credits, its CPU credit balance appears as over the limit. For more information, see [Launch credits \(p. 232\)](#).

T3 and T4g instances do not earn launch credits. These instances launch as `unlimited` by default, and therefore can burst immediately upon start without any launch credits.

Accrued CPU credits life span

CPU credits on a running instance do not expire.

For T2, the CPU credit balance does not persist between instance stops and starts. If you stop a T2 instance, the instance loses all its accrued credits.

For T3 and T4g, the CPU credit balance persists for seven days after an instance stops and the credits are lost thereafter. If you start the instance within seven days, no credits are lost.

For more information, see `CPUCreditBalance` in the [CloudWatch metrics table \(p. 250\)](#).

Baseline utilization

The *baseline utilization* is the level at which the CPU can be utilized for a net credit balance of zero, when the number CPU credits being earned matches the number of CPU credits being used. Baseline utilization is also known as *the baseline*.

Baseline utilization is expressed as a percentage of vCPU utilization, which is calculated as follows:

$(\text{number of credits earned}/\text{number of vCPUs})/60 \text{ minutes} = \% \text{ baseline utilization}$

For example, a `t3.nano` instance, with 2 vCPUs, earns 6 credits per hour, resulting in a baseline utilization of 5%, which is calculated as follows:

$(6 \text{ credits earned}/2 \text{ vCPUs})/60 \text{ minutes} = 5\% \text{ baseline utilization}$

A `t3.xlarge` instance, with 4 vCPUs, earns 96 credits per hour, resulting in a baseline utilization of 40% ($(96/4)/60$).

Unlimited mode for burstable performance instances

A burstable performance instance configured as `unlimited` can sustain high CPU utilization for any period of time whenever required. The hourly instance price automatically covers all CPU usage spikes if the average CPU utilization of the instance is at or below the baseline over a rolling 24-hour period or the instance lifetime, whichever is shorter.

For the vast majority of general-purpose workloads, instances configured as `unlimited` provide ample performance without any additional charges. If the instance runs at higher CPU utilization for a prolonged period, it can do so for a flat additional rate per vCPU-hour. For information about instance pricing, see [Amazon EC2 Pricing](#) and the section for Unlimited Mode Pricing on the [Amazon EC2 On-Demand Pricing](#) page.

If you use a `t2.micro` or `t3.micro` instance under the [AWS Free Tier](#) offer and use it in `unlimited` mode, charges might apply if your average utilization over a rolling 24-hour period exceeds the [baseline utilization \(p. 223\)](#) of the instance.

T3 and T4g instances launch as `unlimited` by default. If the average CPU usage over a 24-hour period exceeds the baseline, you incur charges for surplus credits. If you launch Spot Instances as `unlimited` and plan to use them immediately and for a short duration, with no idle time for accruing

CPU credits, you incur charges for surplus credits. We recommend that you launch your Spot Instances in standard (p. 231) mode to avoid paying higher costs. For more information, see [Surplus credits can incur charges \(p. 227\)](#) and [Burstable performance instances \(p. 438\)](#).

Contents

- [Unlimited mode concepts \(p. 224\)](#)
 - [How Unlimited burstable performance instances work \(p. 224\)](#)
 - [When to use unlimited mode versus fixed CPU \(p. 224\)](#)
 - [Surplus credits can incur charges \(p. 227\)](#)
 - [No launch credits for T2 Unlimited instances \(p. 227\)](#)
 - [Enabling unlimited mode \(p. 227\)](#)
 - [What happens to credits when switching between Unlimited and Standard \(p. 227\)](#)
 - [Monitoring credit usage \(p. 228\)](#)
- [Unlimited mode examples \(p. 228\)](#)
 - [Example 1: Explaining credit use with T3 Unlimited \(p. 228\)](#)
 - [Example 2: Explaining credit use with T2 Unlimited \(p. 229\)](#)

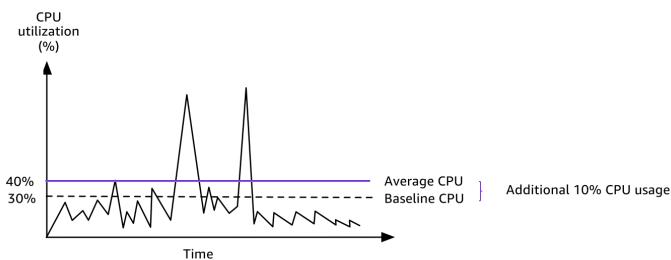
Unlimited mode concepts

The `unlimited` mode is a credit configuration option for burstable performance instances. It can be enabled or disabled at any time for a running or stopped instance. You can set `unlimited` as the default credit option at the account level per AWS Region, per burstable performance instance family, so that all new burstable performance instances in the account launch using the default credit option.

How Unlimited burstable performance instances work

If a burstable performance instance configured as `unlimited` depletes its CPU credit balance, it can spend *surplus* credits to burst beyond the [baseline \(p. 223\)](#). When its CPU utilization falls below the baseline, it uses the CPU credits that it earns to pay down the surplus credits that it spent earlier. The ability to earn CPU credits to pay down surplus credits enables Amazon EC2 to average the CPU utilization of an instance over a 24-hour period. If the average CPU usage over a 24-hour period exceeds the baseline, the instance is billed for the additional usage at a flat additional rate per vCPU-hour.

The following graph shows the CPU usage of a `t3.large`. The baseline CPU utilization for a `t3.large` is 30%. If the instance runs at 30% CPU utilization or less on average over a 24-hour period, there is no additional charge because the cost is already covered by the instance hourly price. However, if the instance runs at 40% CPU utilization on average over a 24-hour period, as shown in the graph, the instance is billed for the additional 10% CPU usage at a flat additional rate per vCPU-hour.



For more information about the baseline utilization per vCPU for each instance type and how many credits each instance type earns, see the [credit table \(p. 221\)](#).

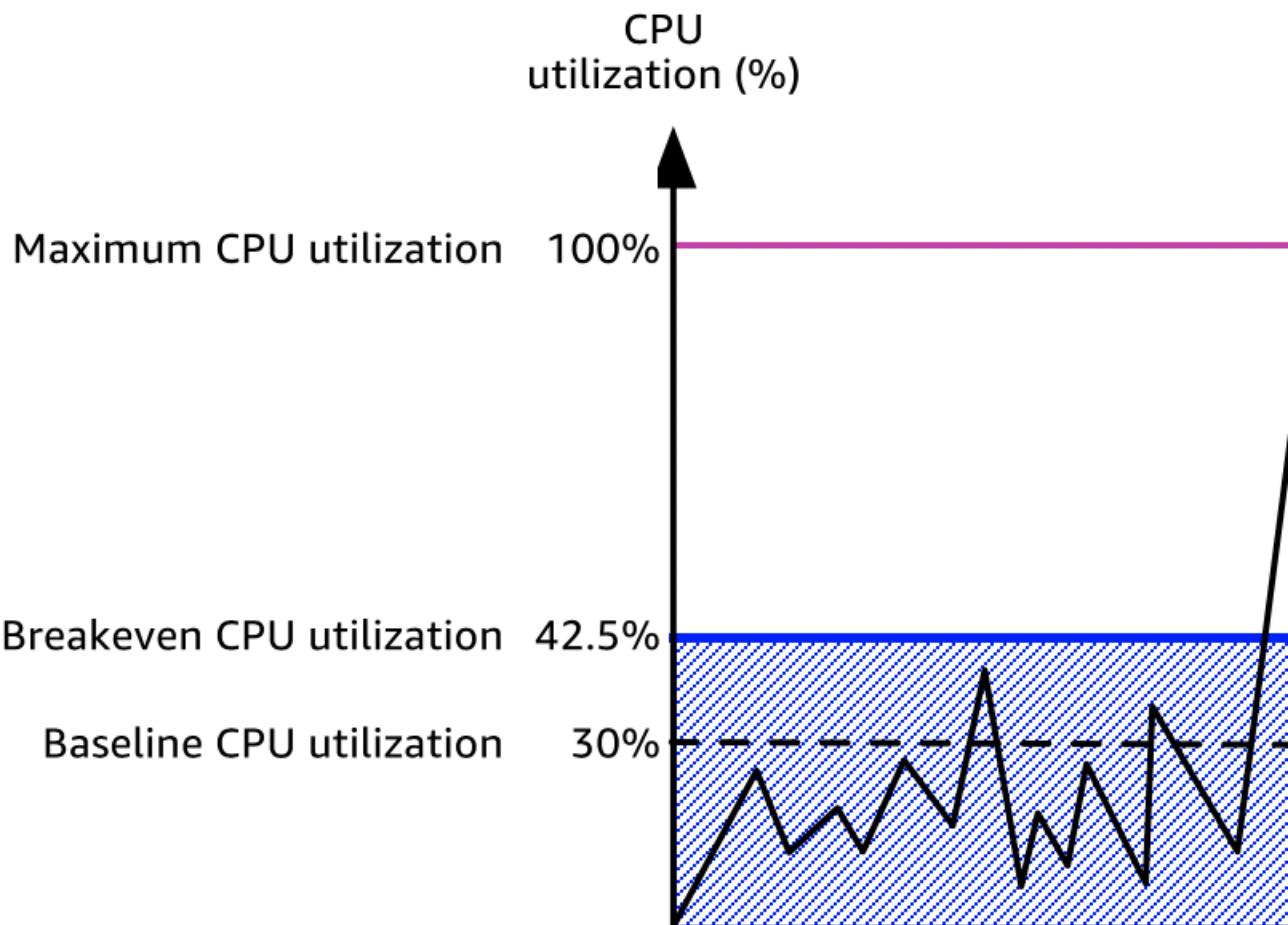
When to use unlimited mode versus fixed CPU

When determining whether you should use a burstable performance instance in `unlimited` mode, such as T3, or a fixed performance instance, such as M5, you need to determine the breakeven CPU

usage. The breakeven CPU usage for a burstable performance instance is the point at which a burstable performance instance costs the same as a fixed performance instance. The breakeven CPU usage helps you determine the following:

- If the average CPU usage over a 24-hour period is at or below the breakeven CPU usage, use a burstable performance instance in **unlimited mode** so that you can benefit from the lower price of a burstable performance instance while getting the same performance as a fixed performance instance.
- If the average CPU usage over a 24-hour period is above the breakeven CPU usage, the burstable performance instance will cost more than the equivalently-sized fixed performance instance. If a T3 instance continuously bursts at 100% CPU, you end up paying approximately 1.5 times the price of an equivalently-sized M5 instance.

The following graph shows the breakeven CPU usage point where a `t3.large` costs the same as an `m5.large`. The breakeven CPU usage point for a `t3.large` is 42.5%. If the average CPU usage is at 42.5%, the cost of running the `t3.large` is the same as an `m5.large`, and is more expensive if the average CPU usage is above 42.5%. If the workload needs less than 42.5% average CPU usage, you can benefit from the lower price of the `t3.large` while getting the same performance as an `m5.large`.



The following table shows how to calculate the breakeven CPU usage threshold so that you can determine when it's less expensive to use a burstable performance instance in `unlimited` mode or a fixed performance instance. The columns in the table are labeled A through K.

Instance type	vCPUs	T3 price*/hour	M5 price*/hour	Price difference	T3 baseline utilization per vCPU (%)	Charge per vCPU hour for surplus credits	Charge per vCPU minute	Additional burst minutes available	Additional CPU % available	Breakeven CPU %
A	B	C	D	E = D - C	F	G	H = G / 60	I = E / H	J = (I / 60) / B	K = F + J
t3.large	2	\$0.0835	\$0.096	\$0.0125	30%	\$0.05	\$0.000833	15	12.5%	42.5%

* Price is based on us-east-1 and Linux OS.

The table provides the following information:

- Column A shows the instance type, `t3.large`.
- Column B shows the number of vCPUs for the `t3.large`.
- Column C shows the price of a `t3.large` per hour.
- Column D shows the price of an `m5.large` per hour.
- Column E shows the price difference between the `t3.large` and the `m5.large`.
- Column F shows the baseline utilization per vCPU of the `t3.large`, which is 30%. At the baseline, the hourly cost of the instance covers the cost of the CPU usage.
- Column G shows the flat additional rate per vCPU-hour that an instance is charged if it bursts at 100% CPU after it has depleted its earned credits.
- Column H shows the flat additional rate per vCPU-minute that an instance is charged if it bursts at 100% CPU after it has depleted its earned credits.
- Column I shows the number of additional minutes that the `t3.large` can burst per hour at 100% CPU while paying the same price per hour as an `m5.large`.
- Column J shows the additional CPU usage (in %) over baseline that the instance can burst while paying the same price per hour as an `m5.large`.
- Column K shows the breakeven CPU usage (in %) that the `t3.large` can burst without paying more than the `m5.large`. Anything above this, and the `t3.large` costs more than the `m5.large`.

The following table shows the breakeven CPU usage (in %) for T3 instance types compared to the similarly-sized M5 instance types.

T3 instance type	Breakeven CPU usage (in %) for T3 compared to M5
<code>t3.large</code>	42.5%
<code>t3.xlarge</code>	52.5%
<code>t3.2xlarge</code>	52.5%

Surplus credits can incur charges

If the average CPU utilization of an instance is at or below the baseline, the instance incurs no additional charges. Because an instance earns a [maximum number of credits \(p. 221\)](#) in a 24-hour period (for example, a `t3.micro` instance can earn a maximum of 288 credits in a 24-hour period), it can spend surplus credits up to that maximum without being charged.

However, if CPU utilization stays above the baseline, the instance cannot earn enough credits to pay down the surplus credits that it has spent. The surplus credits that are not paid down are charged at a flat additional rate per vCPU-hour.

Surplus credits that were spent earlier are charged when any of the following occurs:

- The spent surplus credits exceed the [maximum number of credits \(p. 221\)](#) the instance can earn in a 24-hour period. Spent surplus credits above the maximum are charged at the end of the hour.
- The instance is stopped or terminated.
- The instance is switched from `unlimited` to `standard`.

Spent surplus credits are tracked by the CloudWatch metric `CPUSurplusCreditBalance`. Surplus credits that are charged are tracked by the CloudWatch metric `CPUSurplusCreditsCharged`. For more information, see [Additional CloudWatch metrics for burstable performance instances \(p. 250\)](#).

No launch credits for T2 Unlimited instances

T2 Standard instances receive [launch credits \(p. 232\)](#), but T2 Unlimited instances do not. A T2 Unlimited instance can burst beyond the baseline at any time with no additional charge, as long as its average CPU utilization is at or below the baseline over a rolling 24-hour window or its lifetime, whichever is shorter. As such, T2 Unlimited instances do not require launch credits to achieve high performance immediately after launch.

If a T2 instance is switched from `standard` to `unlimited`, any accrued launch credits are removed from the `CPUCreditBalance` before the remaining `CPUCreditBalance` is carried over.

T3 and T4g instances never receive launch credits.

Enabling unlimited mode

You can switch from `unlimited` to `standard`, and from `standard` to `unlimited`, at any time on a running or stopped instance. For more information, see [Launching a burstable performance instance as Unlimited or Standard \(p. 245\)](#) and [Modifying the credit specification of a burstable performance instance \(p. 248\)](#).

You can set `unlimited` as the default credit option at the account level per AWS Region, per burstable performance instance family, so that all new burstable performance instances in the account launch using the default credit option. For more information, see [Setting the default credit specification for the account \(p. 249\)](#).

You can check whether your burstable performance instance is configured as `unlimited` or `standard` using the Amazon EC2 console or the AWS CLI. For more information, see [Viewing the credit specification of a burstable performance instance \(p. 247\)](#) and [Viewing the default credit specification \(p. 249\)](#).

What happens to credits when switching between Unlimited and Standard

`CPUCreditBalance` is a CloudWatch metric that tracks the number of credits accrued by an instance. `CPUSurplusCreditBalance` is a CloudWatch metric that tracks the number of surplus credits spent by an instance.

When you change an instance configured as `unlimited` to `standard`, the following occurs:

- The `CPUCreditBalance` value remains unchanged and is carried over.

- The `CPUSurplusCreditBalance` value is immediately charged.

When a standard instance is switched to unlimited, the following occurs:

- The `CPUCreditBalance` value containing accrued earned credits is carried over.
- For T2 Standard instances, any launch credits are removed from the `CPUCreditBalance` value, and the remaining `CPUCreditBalance` value containing accrued earned credits is carried over.

Monitoring credit usage

To see if your instance is spending more credits than the baseline provides, you can use CloudWatch metrics to track usage, and you can set up hourly alarms to be notified of credit usage. For more information, see [Monitoring your CPU credits \(p. 250\)](#).

Unlimited mode examples

The following examples explain credit use for instances that are configured as `unlimited`.

Examples

- [Example 1: Explaining credit use with T3 Unlimited \(p. 228\)](#)
- [Example 2: Explaining credit use with T2 Unlimited \(p. 229\)](#)

Example 1: Explaining credit use with T3 Unlimited

In this example, you see the CPU utilization of a `t3.nano` instance launched as `unlimited`, and how it spends *earned* and *surplus* credits to sustain CPU utilization.

A `t3.nano` instance earns 144 CPU credits over a rolling 24-hour period, which it can redeem for 144 minutes of vCPU use. When it depletes its CPU credit balance (represented by the CloudWatch metric `CPUCreditBalance`), it can spend *surplus* CPU credits—that it has *not yet earned*—to burst for as long as it needs. Because a `t3.nano` instance earns a maximum of 144 credits in a 24-hour period, it can spend surplus credits up to that maximum without being charged immediately. If it spends more than 144 CPU credits, it is charged for the difference at the end of the hour.

The intent of the example, illustrated by the following graph, is to show how an instance can burst using surplus credits even after it depletes its `CPUCreditBalance`. The following workflow references the numbered points on the graph:

P1 – At 0 hours on the graph, the instance is launched as `unlimited` and immediately begins to earn credits. The instance remains idle from the time it is launched—CPU utilization is 0%—and no credits are spent. All unspent credits are accrued in the credit balance. For the first 24 hours, `CPUCreditUsage` is at 0, and the `CPUCreditBalance` value reaches its maximum of 144.

P2 – For the next 12 hours, CPU utilization is at 2.5%, which is below the 5% baseline. The instance earns more credits than it spends, but the `CPUCreditBalance` value cannot exceed its maximum of 144 credits.

P3 – For the next 24 hours, CPU utilization is at 7% (above the baseline), which requires a spend of 57.6 credits. The instance spends more credits than it earns, and the `CPUCreditBalance` value reduces to 86.4 credits.

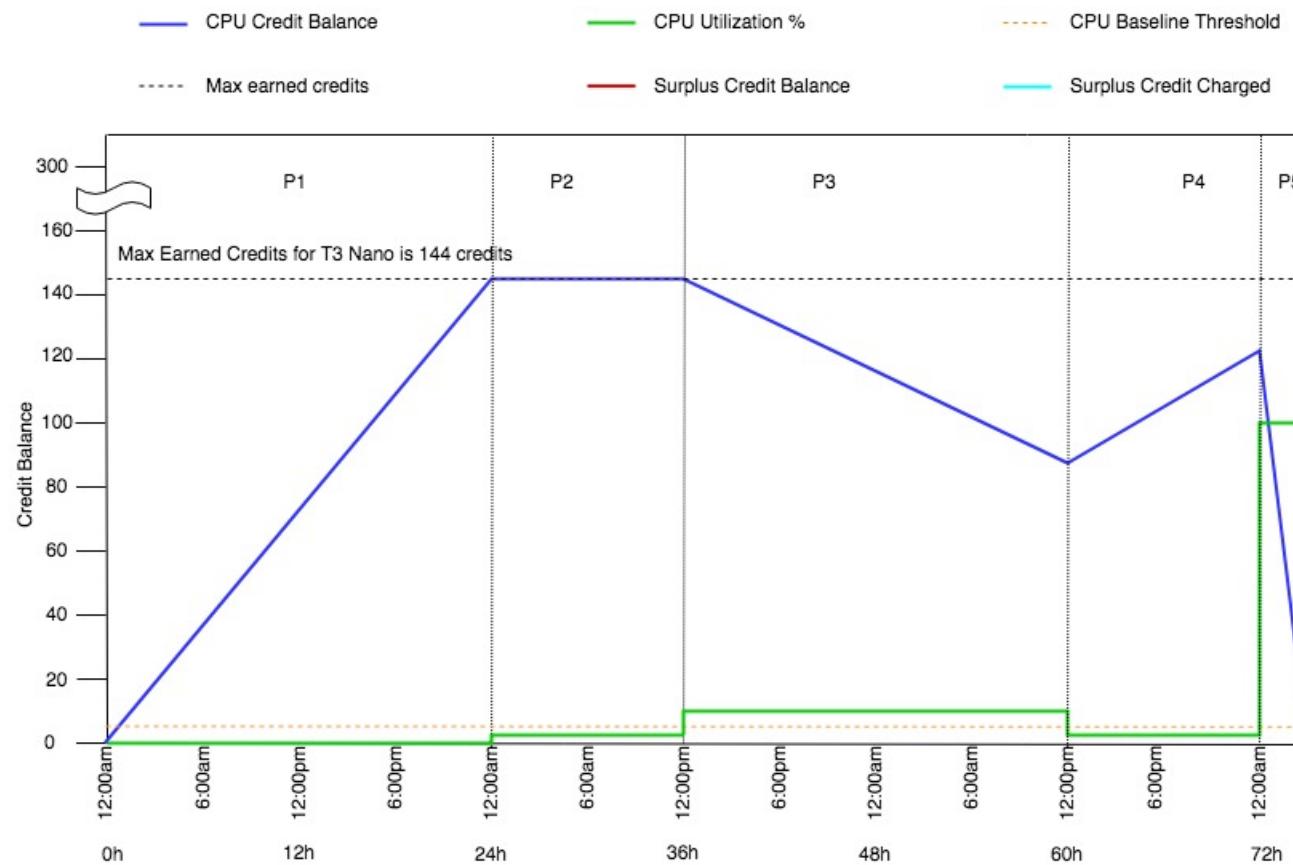
P4 – For the next 12 hours, CPU utilization decreases to 2.5% (below the baseline), which requires a spend of 36 credits. In the same time, the instance earns 72 credits. The instance earns more credits than it spends, and the `CPUCreditBalance` value increases to 122 credits.

P5 – For the next 5 hours, the instance bursts at 100% CPU utilization, and spends a total of 570 credits to sustain the burst. About an hour into this period, the instance depletes its entire `CPUCreditBalance`

of 122 credits, and starts to spend surplus credits to sustain the high CPU utilization, totaling 448 surplus credits in this period ($570 - 122 = 448$). When the `CPUSurplusCreditBalance` value reaches 144 CPU credits (the maximum a `t3.nano` instance can earn in a 24-hour period), any surplus credits spent thereafter cannot be offset by earned credits. The surplus credits spent thereafter amounts to 304 credits ($448 - 144 = 304$), which results in a small additional charge at the end of the hour for 304 credits.

P6 – For the next 13 hours, CPU utilization is at 5% (the baseline). The instance earns as many credits as it spends, with no excess to pay down the `CPUSurplusCreditBalance`. The `CPUSurplusCreditBalance` value remains at 144 credits.

P7 – For the last 24 hours in this example, the instance is idle and CPU utilization is 0%. During this time, the instance earns 144 credits, which it uses to pay down the `CPUSurplusCreditBalance`.



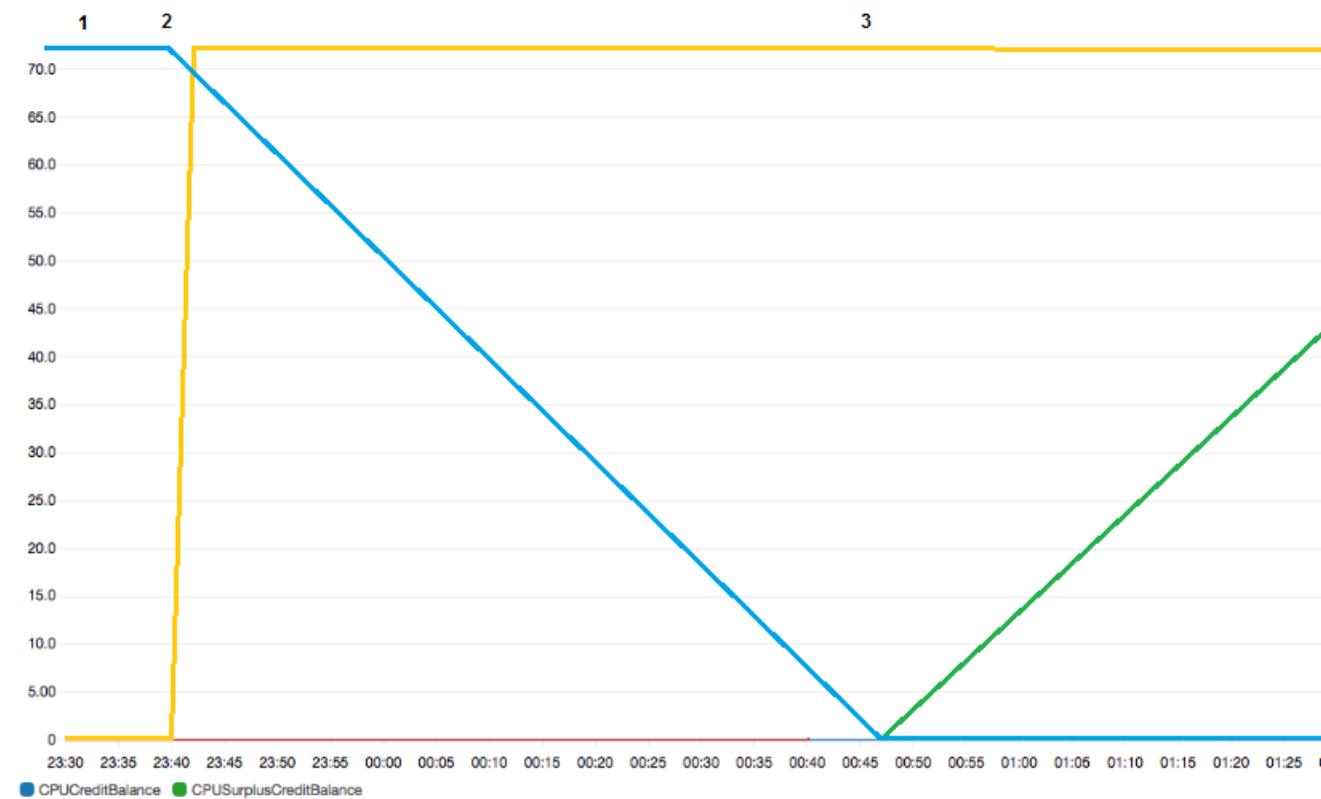
Example 2: Explaining credit use with T2 Unlimited

In this example, you see the CPU utilization of a `t2.nano` instance launched as `unlimited`, and how it spends *earned* and *surplus* credits to sustain CPU utilization.

A `t2.nano` instance earns 72 CPU credits over a rolling 24-hour period, which it can redeem for 72 minutes of vCPU use. When it depletes its CPU credit balance (represented by the CloudWatch metric `CPUCreditBalance`), it can spend *surplus* CPU credits—that it has *not yet earned*—to burst for as long as it needs. Because a `t2.nano` instance earns a maximum of 72 credits in a 24-hour period, it can spend surplus credits up to that maximum without being charged immediately. If it spends more than 72 CPU credits, it is charged for the difference at the end of the hour.

The intent of the example, illustrated by the following graph, is to show how an instance can burst using surplus credits even after it depletes its `CPUCreditBalance`. You can assume that, at the start of the time line in the graph, the instance has an accrued credit balance equal to the maximum number of credits it can earn in 24 hours. The following workflow references the numbered points on the graph:

- 1 – In the first 10 minutes, `CPUCreditUsage` is at 0, and the `CPUCreditBalance` value remains at its maximum of 72.
- 2 – At 23:40, as CPU utilization increases, the instance spends CPU credits and the `CPUCreditBalance` value decreases.
- 3 – At around 00:47, the instance depletes its entire `CPUCreditBalance`, and starts to spend surplus credits to sustain high CPU utilization.
- 4 – Surplus credits are spent until 01:55, when the `CPUSurplusCreditBalance` value reaches 72 CPU credits. This is equal to the maximum a t2.nano instance can earn in a 24-hour period. Any surplus credits spent thereafter cannot be offset by earned credits within the 24-hour period, which results in a small additional charge at the end of the hour.
- 5 – The instance continues to spend surplus credits until around 02:20. At this time, CPU utilization falls below the baseline, and the instance starts to earn credits at 3 credits per hour (or 0.25 credits every 5 minutes), which it uses to pay down the `CPUSurplusCreditBalance`. After the `CPUSurplusCreditBalance` value reduces to 0, the instance starts to accrue earned credits in its `CPUCreditBalance` at 0.25 credits every 5 minutes.



All metrics		Graphed metrics (4)	Graph options
	Label	Details	
■	CPUCreditBalance	EC2 * InstanceId:i-0aa4b948d7eb37d6b * CPUCreditBalance	
■	CPUCreditUsage	EC2 * InstanceId:i-0aa4b948d7eb37d6b * CPUCreditUsage	
■	CPUSurplusCreditBalance	EC2 * InstanceId:i-0aa4b948d7eb37d6b * CPUSurplusCreditBalance	
■	CPUSurplusCreditsCharged	EC2 * InstanceId:i-0aa4b948d7eb37d6b * CPUSurplusCreditsCharged	

Calculating the bill

Surplus credits cost \$0.05 per vCPU-hour. The instance spent approximately 25 surplus credits between 01:55 and 02:20, which is equivalent to 0.42 vCPU-hours.

Additional charges for this instance are $0.42 \text{ vCPU-hours} \times \$0.05/\text{vCPU-hour} = \0.021 , rounded to \$0.02.

Here is the month-end bill for this T2 Unlimited instance:

Amazon Elastic Compute Cloud running Linux/UNIX			
\$0.0058 per On Demand Linux t2.nano Instance Hour		720.000 Hrs	\$4.18
Amazon Elastic Compute Cloud T2 CPU Credits			
\$0.05 per vCPU-Hour of T2 CPU credits		0.420 vCPU-Hours	\$0.02

You can set billing alerts to be notified every hour of any accruing charges, and take action if required.

Standard mode for burstable performance instances

A burstable performance instance configured as `standard` is suited to workloads with an average CPU utilization that is consistently below the baseline CPU utilization of the instance. To burst above the baseline, the instance spends credits that it has accrued in its CPU credit balance. If the instance is running low on accrued credits, CPU utilization is gradually lowered to the baseline level, so that the instance does not experience a sharp performance drop-off when its accrued CPU credit balance is depleted. For more information, see [CPU credits and baseline utilization for burstable performance instances \(p. 220\)](#).

Contents

- [Standard mode concepts \(p. 231\)](#)
 - [How standard burstable performance instances work \(p. 231\)](#)
 - [Launch credits \(p. 232\)](#)
 - [Launch credit limits \(p. 232\)](#)
 - [Differences between launch credits and earned credits \(p. 233\)](#)
- [Standard mode examples \(p. 233\)](#)
 - [Example 1: Explaining credit use with T3 Standard \(p. 233\)](#)
 - [Example 2: Explaining credit use with T2 Standard \(p. 235\)](#)
 - [Period 1: 1 – 24 hours \(p. 236\)](#)
 - [Period 2: 25 – 36 hours \(p. 237\)](#)
 - [Period 3: 37 – 61 hours \(p. 238\)](#)
 - [Period 4: 62 – 72 hours \(p. 239\)](#)
 - [Period 5: 73 – 75 hours \(p. 241\)](#)
 - [Period 6: 76 – 90 hours \(p. 242\)](#)
 - [Period 7: 91 – 96 hours \(p. 244\)](#)

Standard mode concepts

The `standard` mode is a configuration option for burstable performance instances. It can be enabled or disabled at any time for a running or stopped instance. You can set `standard` as the default credit option at the account level per AWS Region, per burstable performance instance family, so that all new burstable performance instances in the account launch using the default credit option.

How standard burstable performance instances work

When a burstable performance instance configured as `standard` is in a running state, it continuously earns (at a millisecond-level resolution) a set rate of earned credits per hour. For T2 Standard, when

the instance is stopped, it loses all its accrued credits, and its credit balance is reset to zero. When it is restarted, it receives a new set of launch credits, and begins to accrue earned credits. For T3 and T4g Standard instances, the CPU credit balance persists for seven days after the instance stops and the credits are lost thereafter. If you start the instance within seven days, no credits are lost.

T2 Standard instances receive two types of CPU credits: *earned credits* and *launch credits*. When a T2 Standard instance is in a running state, it continuously earns (at a millisecond-level resolution) a set rate of earned credits per hour. At start, it has not yet earned credits for a good startup experience; therefore, to provide a good startup experience, it receives launch credits at start, which it spends first while it accrues earned credits.

T3 and T4g Standard instances do not receive launch credits.

Launch credits

T2 Standard instances get 30 launch credits per vCPU at launch or start. For example, a t2.micro instance has one vCPU and gets 30 launch credits, while a t2.xlarge instance has four vCPUs and gets 120 launch credits. Launch credits are designed to provide a good startup experience to allow instances to burst immediately after launch before they have accrued earned credits.

Launch credits are spent first, before earned credits. Unspent launch credits are accrued in the CPU credit balance, but do not count towards the CPU credit balance limit. For example, a t2.micro instance has a CPU credit balance limit of 144 earned credits. If it is launched and remains idle for 24 hours, its CPU credit balance reaches 174 (30 launch credits + 144 earned credits), which is over the limit. However, after the instance spends the 30 launch credits, the credit balance cannot exceed 144. For more information about the CPU credit balance limit for each instance size, see the [credit table \(p. 221\)](#).

The following table lists the initial CPU credit allocation received at launch or start, and the number of vCPUs.

Instance type	Launch credits	vCPUs
t1.micro	15	1
t2.nano	30	1
t2.micro	30	1
t2.small	30	1
t2.medium	60	2
t2.large	60	2
t2.xlarge	120	4
t2.2xlarge	240	8

Launch credit limits

There is a limit to the number of times T2 Standard instances can receive launch credits. The default limit is 100 launches or starts of all T2 Standard instances combined per account, per Region, per rolling 24-hour period. For example, the limit is reached when one instance is stopped and started 100 times within a 24-hour period, or when 100 instances are launched within a 24-hour period, or other combinations that equate to 100 starts. New accounts may have a lower limit, which increases over time based on your usage.

Tip

To ensure that your workloads always get the performance they need, switch to [Unlimited mode for burstable performance instances \(p. 223\)](#) or consider using a larger instance size.

Differences between launch credits and earned credits

The following table lists the differences between launch credits and earned credits.

	Launch credits	Earned credits
Credit earn rate	T2 Standard instances get 30 launch credits per vCPU at launch or start. If a T2 instance is switched from unlimited to standard , it does not get launch credits at the time of switching.	Each T2 instance continuously earns (at a millisecond-level resolution) a set rate of CPU credits per hour, depending on the instance size. For more information about the number of CPU credits earned per instance size, see the credit table (p. 221) .
Credit earn limit	The limit for receiving launch credits is 100 launches or starts of all T2 Standard instances combined per account, per Region, per rolling 24-hour period. New accounts may have a lower limit, which increases over time based on your usage.	A T2 instance cannot accrue more credits than the CPU credit balance limit. If the CPU credit balance has reached its limit, any credits that are earned after the limit is reached are discarded. Launch credits do not count towards the limit. For more information about the CPU credit balance limit for each T2 instance size, see the credit table (p. 221) .
Credit use	Launch credits are spent first, before earned credits.	Earned credits are spent only after all launch credits are spent.
Credit expiration	When a T2 Standard instance is running, launch credits do not expire. When a T2 Standard instance stops or is switched to T2 Unlimited, all launch credits are lost.	When a T2 instance is running, earned credits that have accrued do not expire. When the T2 instance stops, all accrued earned credits are lost.

The number of accrued launch credits and accrued earned credits is tracked by the CloudWatch metric `CPUCreditBalance`. For more information, see `CPUCreditBalance` in the [CloudWatch metrics table \(p. 250\)](#).

Standard mode examples

The following examples explain credit use when instances are configured as **standard**.

Examples

- [Example 1: Explaining credit use with T3 Standard \(p. 233\)](#)
- [Example 2: Explaining credit use with T2 Standard \(p. 235\)](#)

Example 1: Explaining credit use with T3 Standard

In this example, you see how a `t3.nano` instance launched as **standard** earns, accrues, and spends *earned* credits. You see how the credit balance reflects the accrued *earned* credits.

A running `t3.nano` instance earns 144 credits every 24 hours. Its credit balance limit is 144 earned credits. After the limit is reached, new credits that are earned are discarded. For more information about the number of credits that can be earned and accrued, see the [credit table \(p. 221\)](#).

You might launch a T3 Standard instance and use it immediately. Or, you might launch a T3 Standard instance and leave it idle for a few days before running applications on it. Whether an instance is used or remains idle determines if credits are spent or accrued. If an instance remains idle for 24 hours from the

time it is launched, the credit balance reaches its limit, which is the maximum number of earned credits that can be accrued.

This example describes an instance that remains idle for 24 hours from the time it is launched, and walks you through seven periods of time over a 96-hour period, showing the rate at which credits are earned, accrued, spent, and discarded, and the value of the credit balance at the end of each period.

The following workflow references the numbered points on the graph:

P1 – At 0 hours on the graph, the instance is launched as standard and immediately begins to earn credits. The instance remains idle from the time it is launched—CPU utilization is 0%—and no credits are spent. All unspent credits are accrued in the credit balance. For the first 24 hours, `CPUCreditUsage` is at 0, and the `CPUCreditBalance` value reaches its maximum of 144.

P2 – For the next 12 hours, CPU utilization is at 2.5%, which is below the 5% baseline. The instance earns more credits than it spends, but the `CPUCreditBalance` value cannot exceed its maximum of 144 credits. Any credits that are earned in excess of the limit are discarded.

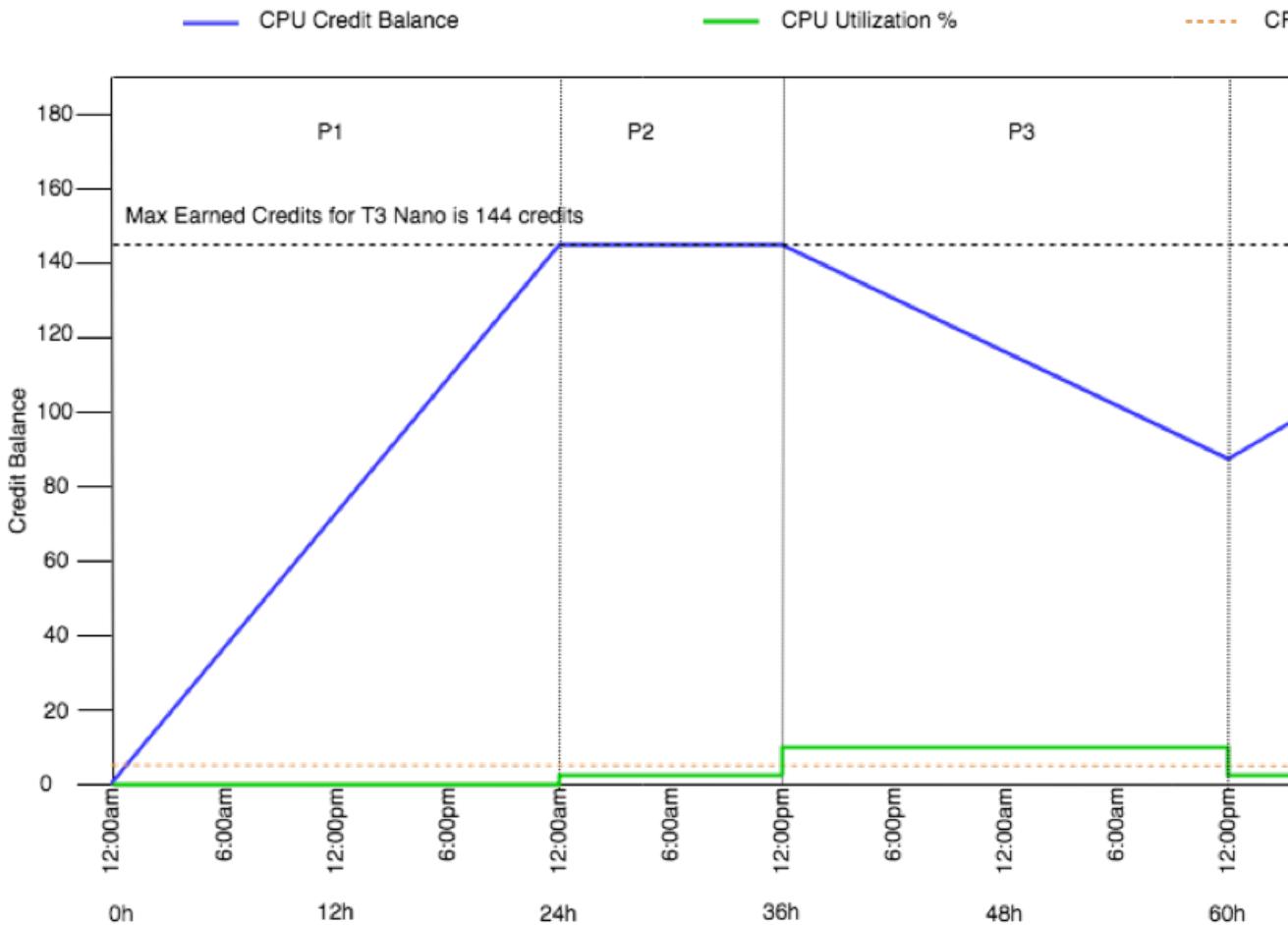
P3 – For the next 24 hours, CPU utilization is at 7% (above the baseline), which requires a spend of 57.6 credits. The instance spends more credits than it earns, and the `CPUCreditBalance` value reduces to 86.4 credits.

P4 – For the next 12 hours, CPU utilization decreases to 2.5% (below the baseline), which requires a spend of 36 credits. In the same time, the instance earns 72 credits. The instance earns more credits than it spends, and the `CPUCreditBalance` value increases to 122 credits.

P5 – For the next two hours, the instance bursts at 100% CPU utilization, and depletes its entire `CPUCreditBalance` value of 122 credits. At the end of this period, with the `CPUCreditBalance` at zero, CPU utilization is forced to drop to the baseline utilization level of 5%. At the baseline, the instance earns as many credits as it spends.

P6 – For the next 14 hours, CPU utilization is at 5% (the baseline). The instance earns as many credits as it spends. The `CPUCreditBalance` value remains at 0.

P7 – For the last 24 hours in this example, the instance is idle and CPU utilization is 0%. During this time, the instance earns 144 credits, which it accrues in its `CPUCreditBalance`.



Example 2: Explaining credit use with T2 Standard

In this example, you see how a `t2.nano` instance launched as standard earns, accrues, and spends *launch* and *earned* credits. You see how the credit balance reflects not only accrued *earned* credits, but also accrued *launch* credits.

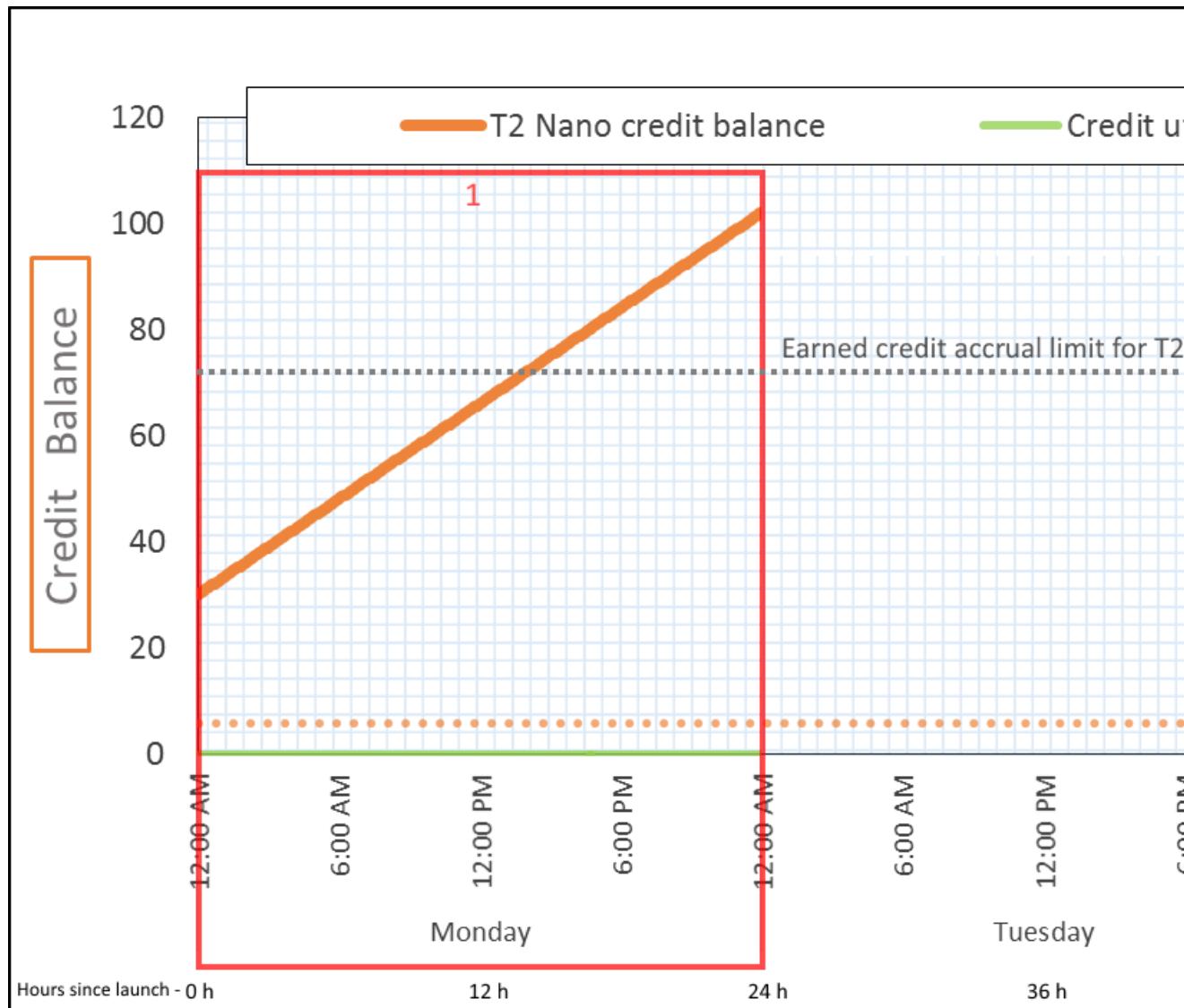
A `t2.nano` instance gets 30 launch credits when it is launched, and earns 72 credits every 24 hours. Its credit balance limit is 72 earned credits; launch credits do not count towards the limit. After the limit is reached, new credits that are earned are discarded. For more information about the number of credits that can be earned and accrued, see the [credit table \(p. 221\)](#). For more information about limits, see [Launch credit limits \(p. 232\)](#).

You might launch a T2 Standard instance and use it immediately. Or, you might launch a T2 Standard instance and leave it idle for a few days before running applications on it. Whether an instance is used or remains idle determines if credits are spent or accrued. If an instance remains idle for 24 hours from the time it is launched, the credit balance appears to exceed its limit because the balance reflects both accrued earned credits and accrued launch credits. However, after CPU is used, the launch credits are spent first. Thereafter, the limit always reflects the maximum number of earned credits that can be accrued.

This example describes an instance that remains idle for 24 hours from the time it is launched, and walks you through seven periods of time over a 96-hour period, showing the rate at which credits are earned, accrued, spent, and discarded, and the value of the credit balance at the end of each period.

Period 1: 1 – 24 hours

At 0 hours on the graph, the T2 instance is launched as standard and immediately gets 30 launch credits. It earns credits while in the running state. The instance remains idle from the time it is launched—CPU utilization is 0%—and no credits are spent. All unspent credits are accrued in the credit balance. At approximately 14 hours after launch, the credit balance is 72 (30 launch credits + 42 earned credits), which is equivalent to what the instance can earn in 24 hours. At 24 hours after launch, the credit balance exceeds 72 credits because the unspent launch credits are accrued in the credit balance—the credit balance is 102 credits: 30 launch credits + 72 earned credits.



Credit Spend Rate	0 credits per 24 hours (0% CPU utilization)
Credit Earn Rate	72 credits per 24 hours
Credit Discard Rate	0 credits per 24 hours
Credit Balance	102 credits (30 launch credits + 72 earned credits)

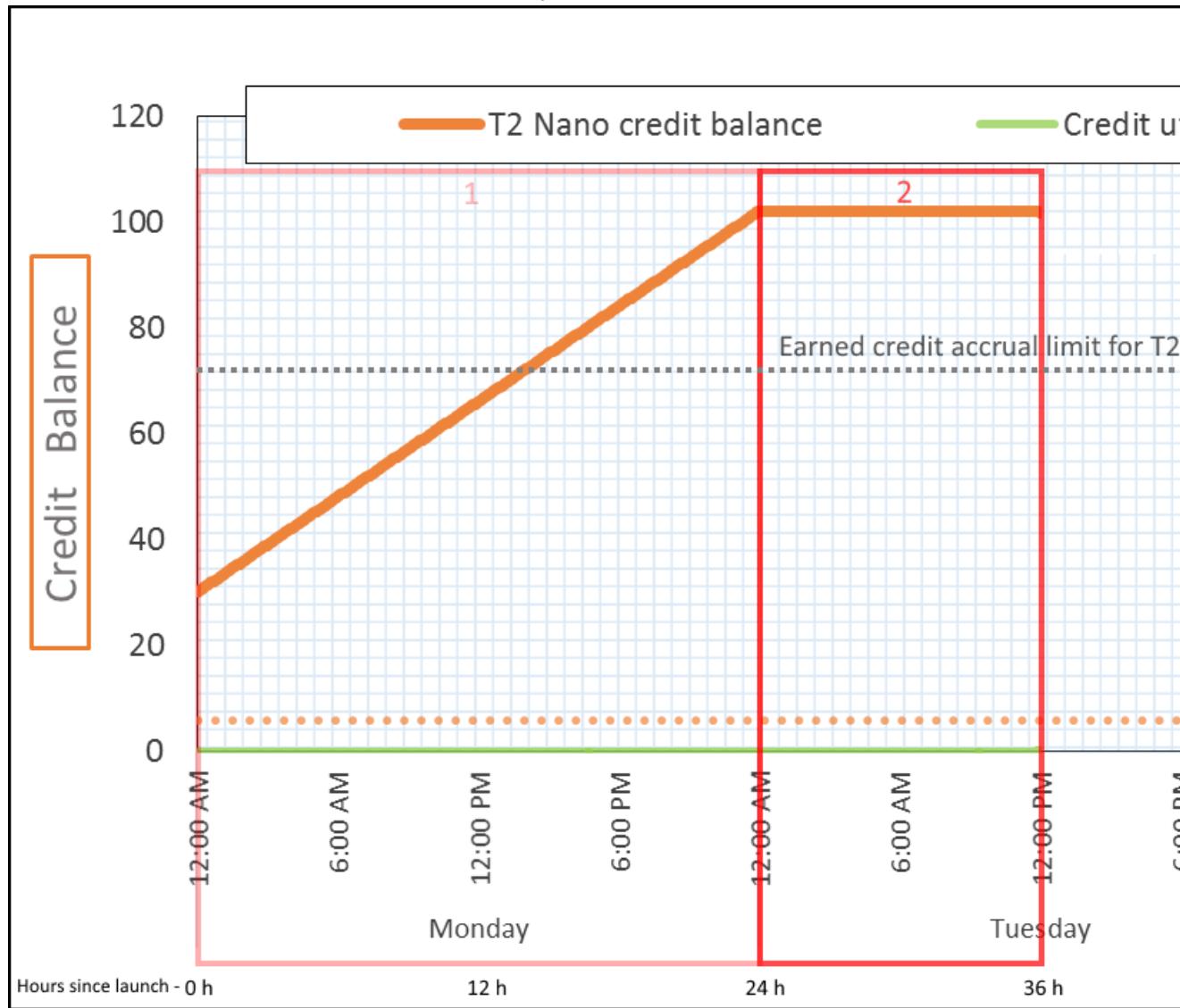
Conclusion

If there is no CPU utilization after launch, the instance accrues more credits than what it can earn in 24 hours (30 launch credits + 72 earned credits = 102 credits).

In a real-world scenario, an EC2 instance consumes a small number of credits while launching and running, which prevents the balance from reaching the maximum theoretical value in this example.

Period 2: 25 – 36 hours

For the next 12 hours, the instance continues to remain idle and earn credits, but the credit balance does not increase. It plateaus at 102 credits (30 launch credits + 72 earned credits). The credit balance has reached its limit of 72 accrued earned credits, so newly earned credits are discarded.



Credit Spend Rate	0 credits per 24 hours (0% CPU utilization)
Credit Earn Rate	72 credits per 24 hours (3 credits per hour)
Credit Discard Rate	72 credits per 24 hours (100% of credit earn rate)

Credit Balance

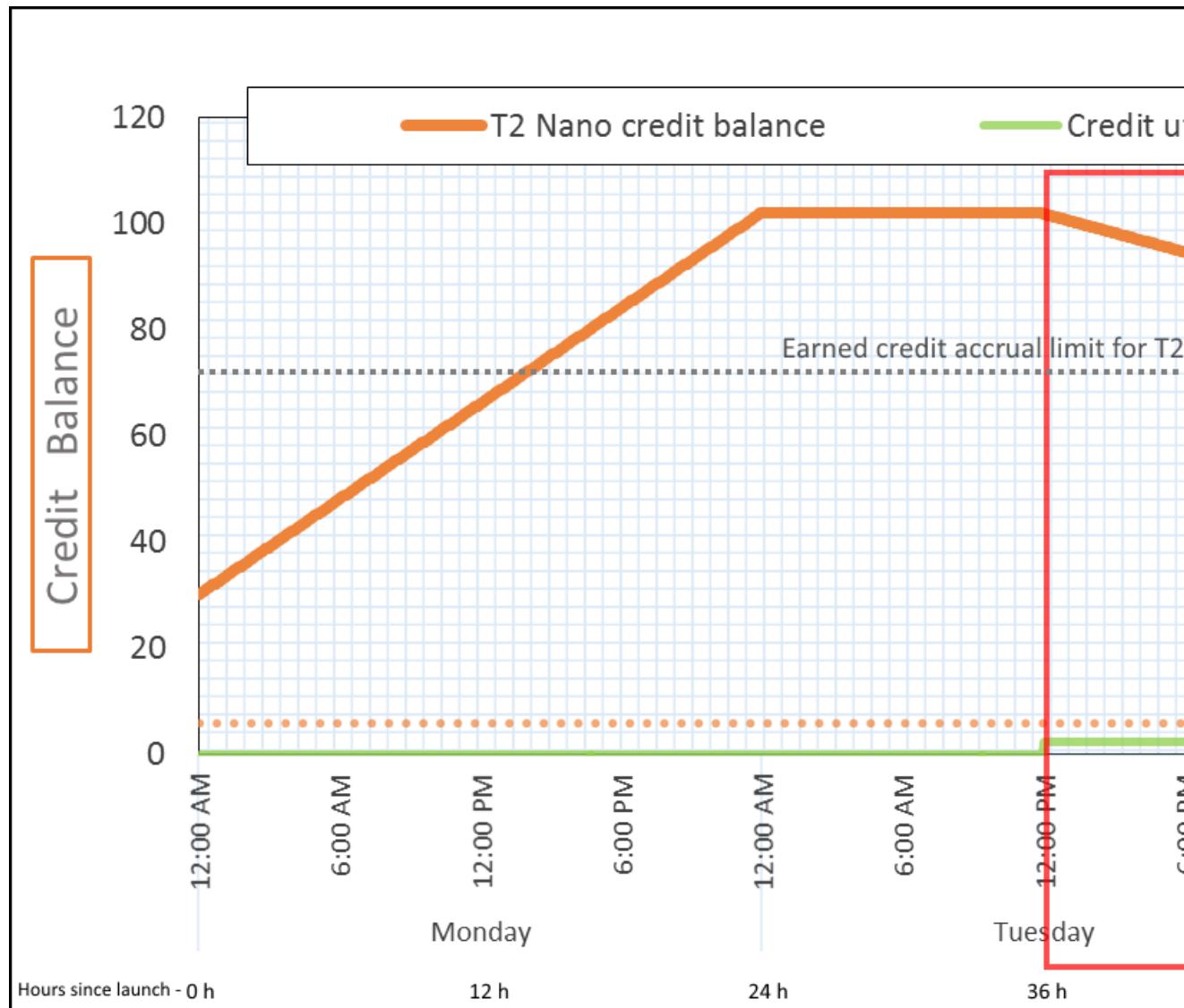
102 credits (30 launch credits + 72 earned credits)
—balance is unchanged

Conclusion

An instance constantly earns credits, but it cannot accrue more earned credits if the credit balance has reached its limit. After the limit is reached, newly earned credits are discarded. Launch credits do not count towards the credit balance limit. If the balance includes accrued launch credits, the balance appears to be over the limit.

Period 3: 37 – 61 hours

For the next 25 hours, the instance uses 2% CPU, which requires 30 credits. In the same period, it earns 75 credits, but the credit balance decreases. The balance decreases because the accrued *launch* credits are spent first, while newly earned credits are discarded because the credit balance is already at its limit of 72 earned credits.



Credit Spend Rate	28.8 credits per 24 hours (1.2 credits per hour, 2% CPU utilization, 40% of credit earn rate)—30 credits over 25 hours
Credit Earn Rate	72 credits per 24 hours
Credit Discard Rate	72 credits per 24 hours (100% of credit earn rate)
Credit Balance	72 credits (30 launch credits were spent; 72 earned credits remain unspent)

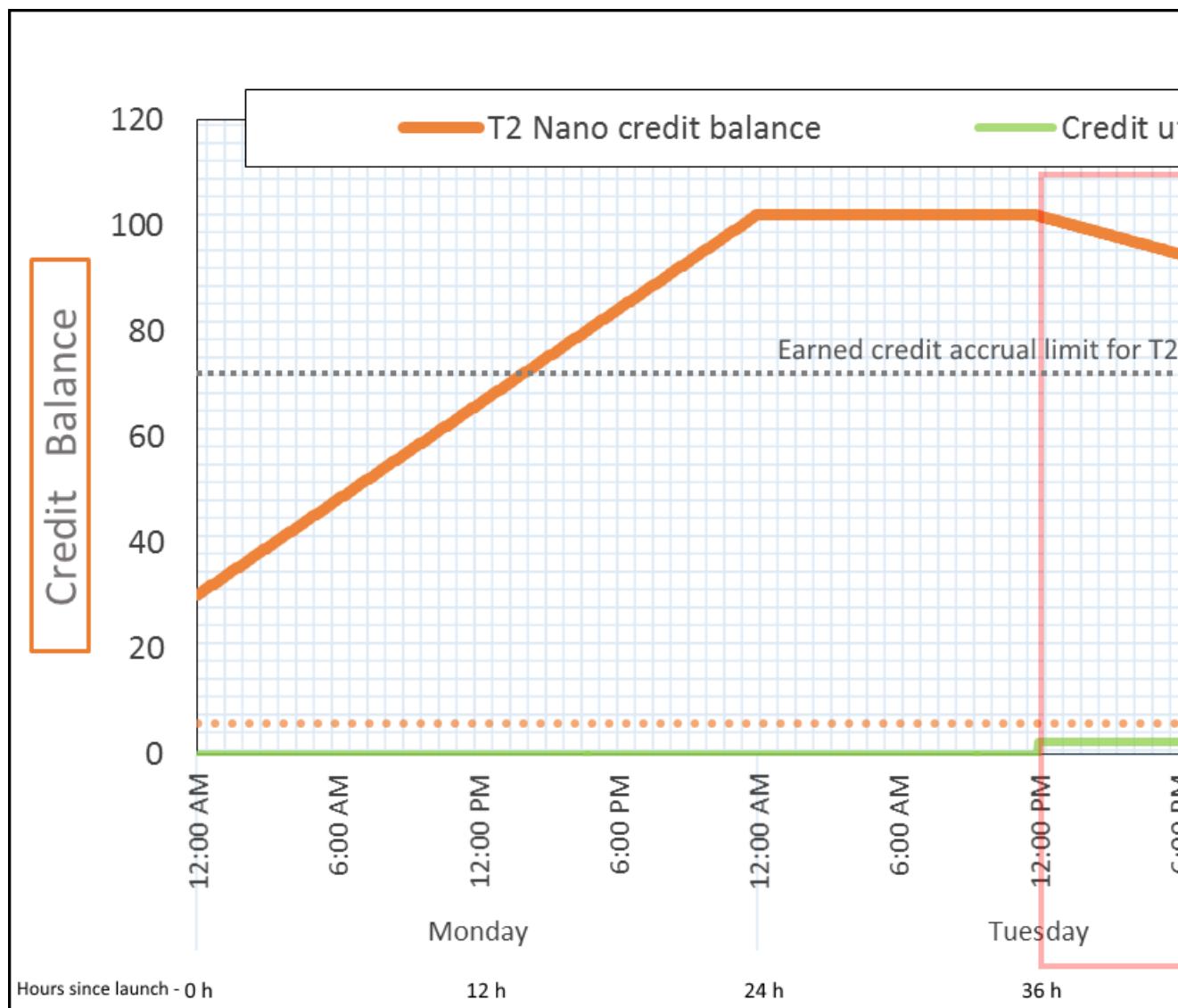
Conclusion

An instance spends launch credits first, before spending earned credits. Launch credits do not count towards the credit limit. After the launch credits are spent, the balance can never go higher than what can be earned in 24 hours. Furthermore, while an instance is running, it cannot get more launch credits.

Period 4: 62 – 72 hours

For the next 11 hours, the instance uses 2% CPU, which requires 13.2 credits. This is the same CPU utilization as in the previous period, but the balance does not decrease. It stays at 72 credits.

The balance does not decrease because the credit earn rate is higher than the credit spend rate. In the time that the instance spends 13.2 credits, it also earns 33 credits. However, the balance limit is 72 credits, so any earned credits that exceed the limit are discarded. The balance plateaus at 72 credits, which is different from the plateau of 102 credits during Period 2, because there are no accrued launch credits.



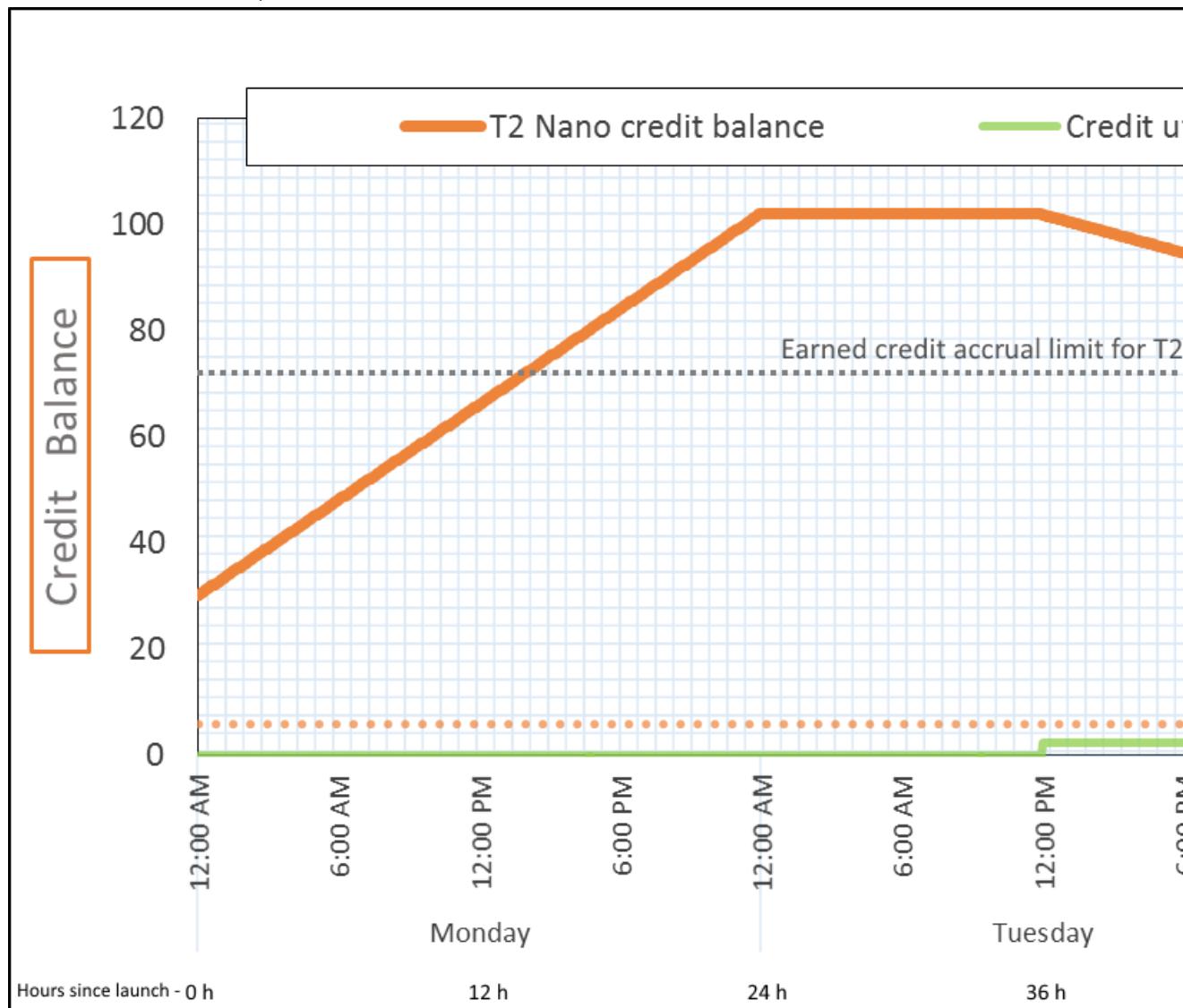
Credit Spend Rate	28.8 credits per 24 hours (1.2 credits per hour, 2% CPU utilization, 40% of credit earn rate)—13.2 credits over 11 hours
Credit Earn Rate	72 credits per 24 hours
Credit Discard Rate	43.2 credits per 24 hours (60% of credit earn rate)
Credit Balance	72 credits (0 launch credits, 72 earned credits)—balance is at its limit

Conclusion

After launch credits are spent, the credit balance limit is determined by the number of credits that an instance can earn in 24 hours. If the instance earns more credits than it spends, newly earned credits over the limit are discarded.

Period 5: 73 – 75 hours

For the next three hours, the instance bursts at 20% CPU utilization, which requires 36 credits. The instance earns nine credits in the same three hours, which results in a net balance decrease of 27 credits. At the end of three hours, the credit balance is 45 accrued earned credits.



Credit Spend Rate	288 credits per 24 hours (12 credits per hour, 20% CPU utilization, 400% of credit earn rate)—36 credits over 3 hours
Credit Earn Rate	72 credits per 24 hours (9 credits over 3 hours)
Credit Discard Rate	0 credits per 24 hours
Credit Balance	45 credits (previous balance (72) - spent credits (36) + earned credits (9))—balance decreases at a rate of 216 credits per 24 hours (spend rate)

	$\frac{288}{24} + \text{earn rate } \frac{72}{24} = \text{balance decrease rate}$ $216/24)$
--	---------------------------------------------------------------------------------------------

Conclusion

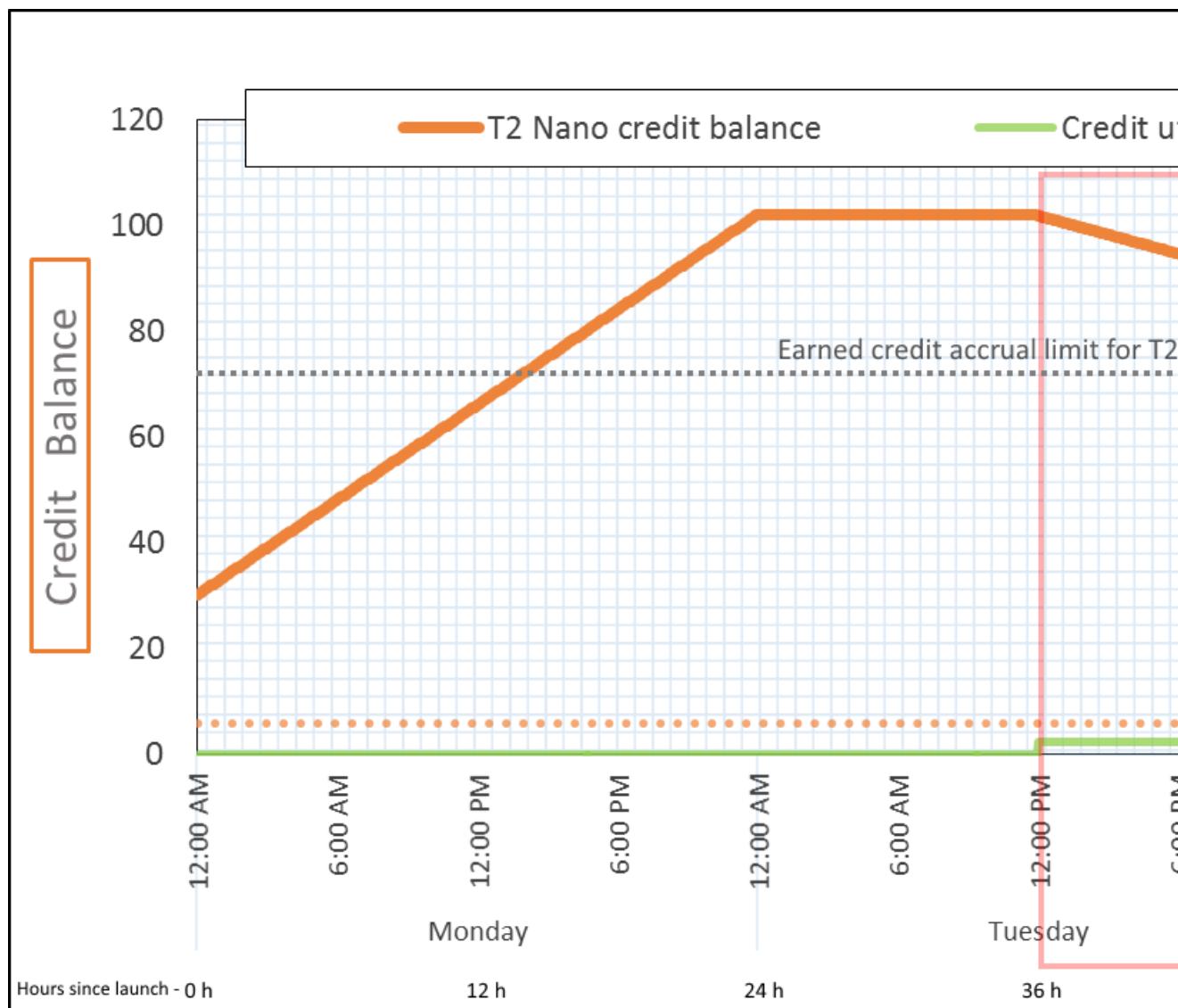
If an instance spends more credits than it earns, its credit balance decreases.

Period 6: 76 – 90 hours

For the next 15 hours, the instance uses 2% CPU, which requires 18 credits. This is the same CPU utilization as in Periods 3 and 4. However, the balance increases in this period, whereas it decreased in Period 3 and plateaued in Period 4.

In Period 3, the accrued launch credits were spent, and any earned credits that exceeded the credit limit were discarded, resulting in a decrease in the credit balance. In Period 4, the instance spent fewer credits than it earned. Any earned credits that exceeded the limit were discarded, so the balance plateaued at its maximum of 72 credits.

In this period, there are no accrued launch credits, and the number of accrued earned credits in the balance is below the limit. No earned credits are discarded. Furthermore, the instance earns more credits than it spends, resulting in an increase in the credit balance.



Credit Spend Rate	28.8 credits per 24 hours (1.2 credits per hour, 2% CPU utilization, 40% of credit earn rate)—18 credits over 15 hours
Credit Earn Rate	72 credits per 24 hours (45 credits over 15 hours)
Credit Discard Rate	0 credits per 24 hours
Credit Balance	72 credits (balance increases at a rate of 43.2 credits per 24 hours—change rate = spend rate 28.8/24 + earn rate 72/24)

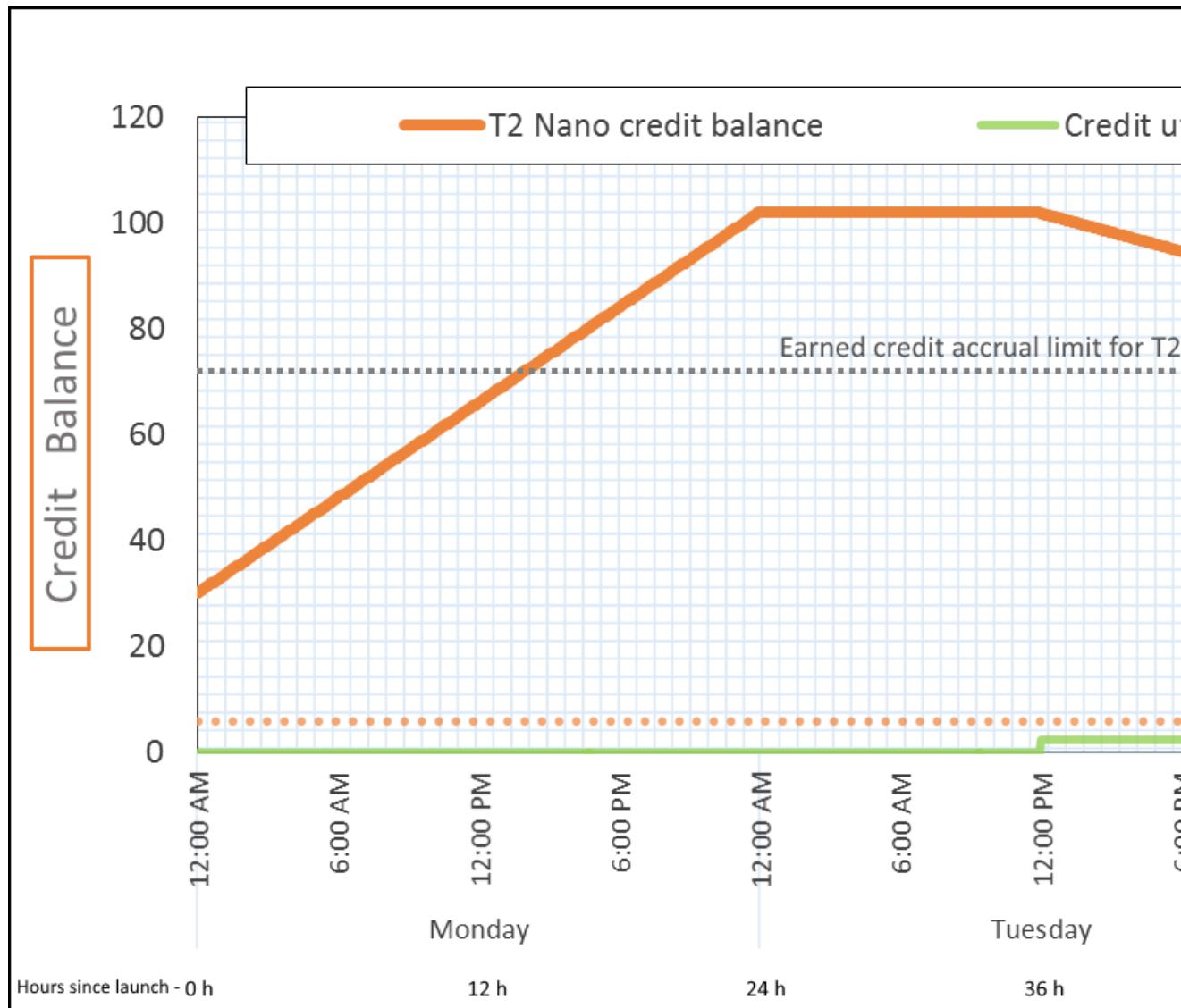
Conclusion

If an instance spends fewer credits than it earns, its credit balance increases.

Period 7: 91 – 96 hours

For the next six hours, the instance remains idle—CPU utilization is 0%—and no credits are spent. This is the same CPU utilization as in Period 2, but the balance does not plateau at 102 credits—it plateaus at 72 credits, which is the credit balance limit for the instance.

In Period 2, the credit balance included 30 accrued launch credits. The launch credits were spent in Period 3. A running instance cannot get more launch credits. After its credit balance limit is reached, any earned credits that exceed the limit are discarded.



Credit Spend Rate	0 credits per 24 hours (0% CPU utilization)
Credit Earn Rate	72 credits per 24 hours
Credit Discard Rate	72 credits per 24 hours (100% of credit earn rate)
Credit Balance	72 credits (0 launch credits, 72 earned credits)

Conclusion

An instance constantly earns credits, but cannot accrue more earned credits if the credit balance limit has been reached. After the limit is reached, newly earned credits are discarded. The credit balance limit is determined by the number of credits that an instance can earn in 24 hours. For more information about credit balance limits, see the [credit table \(p. 221\)](#).

Working with burstable performance instances

The steps for launching, monitoring, and modifying these instances are similar. The key difference is the default credit specification when they launch. If you do not change the default credit specification, the default is that:

- T3 and T4g instances launch as `unlimited`
- T2 instances launch as `standard`

Contents

- [Launching a burstable performance instance as Unlimited or Standard \(p. 245\)](#)
- [Using an Auto Scaling group to launch a burstable performance instance as Unlimited \(p. 246\)](#)
- [Viewing the credit specification of a burstable performance instance \(p. 247\)](#)
- [Modifying the credit specification of a burstable performance instance \(p. 248\)](#)
- [Setting the default credit specification for the account \(p. 249\)](#)
- [Viewing the default credit specification \(p. 249\)](#)

Launching a burstable performance instance as Unlimited or Standard

You can launch your instances as `unlimited` or `standard` using the Amazon EC2 console, an AWS SDK, a command line tool, or with an Auto Scaling group. For more information, see [Using an Auto Scaling group to launch a burstable performance instance as Unlimited \(p. 246\)](#).

Requirements

- You must launch your instances using an Amazon EBS volume as the root device. For more information, see [Amazon EC2 root device volume \(p. 20\)](#).
- For more information about AMI and driver requirements for these instances, see [Release notes \(p. 218\)](#).

To launch a burstable performance instance as Unlimited or Standard (console)

1. Follow the [Launching an instance using the Launch Instance Wizard \(p. 499\)](#) procedure.
2. On the **Choose an Instance Type** page, select an instance type, and choose **Next: Configure Instance Details**.
3. Choose a credit specification.
 - a. To launch a T3 or T4g instance as `standard`, clear **Unlimited**.
 - b. To launch a T2 instance as `unlimited`, select **Unlimited**.
4. Continue as prompted by the wizard. When you've finished reviewing your options on the **Review Instance Launch** page, choose **Launch**. For more information, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#).

To launch a burstable performance instance as Unlimited or Standard (AWS CLI)

Use the [run-instances](#) command to launch your instances. Specify the credit specification using the `--credit-specification CpuCredits=` parameter. Valid credit specifications are `unlimited` and `standard`.

- For T3 and T4g, if you do not include the `--credit-specification` parameter, the instance launches as `unlimited` by default.
- For T2, if you do not include the `--credit-specification` parameter, the instance launches as `standard` by default.

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t3.micro --key-name MyKeyPair --credit-specification "CpuCredits=unlimited"
```

Using an Auto Scaling group to launch a burstable performance instance as Unlimited

When burstable performance instances are launched or started, they require CPU credits for a good bootstrapping experience. If you use an Auto Scaling group to launch your instances, we recommend that you configure your instances as `unlimited`. If you do, the instances use surplus credits when they are automatically launched or restarted by the Auto Scaling group. Using surplus credits prevents performance restrictions.

Creating a launch template

You must use a *launch template* for launching instances as `unlimited` in an Auto Scaling group. A launch configuration does not support launching instances as `unlimited`.

To create a launch template that launches instances as Unlimited (console)

1. Follow the [Creating a Launch Template for an Auto Scaling Group](#) procedure.
2. In **Launch template contents**, for **Instance type**, choose an instance size.
3. To launch instances as `unlimited` in an Auto Scaling group, under **Advanced details**, for **Credit specification**, choose **Unlimited**.
4. When you've finished defining the launch template parameters, choose **Create launch template**. For more information, see [Creating a Launch Template for an Auto Scaling Group](#) in the *Amazon EC2 Auto Scaling User Guide*.

To create a launch template that launches instances as Unlimited (AWS CLI)

Use the `create-launch-template` command and specify `unlimited` as the credit specification.

- For T3 and T4g, if you do not include the `CreditSpecification={CpuCredits=unlimited}` value, the instance launches as `unlimited` by default.
- For T2, if you do not include the `CreditSpecification={CpuCredits=unlimited}` value, the instance launches as `standard` by default.

```
aws ec2 create-launch-template --launch-template-name MyLaunchTemplate --version-description FirstVersion --launch-template-data ImageId=ami-8c1be5f6,InstanceType=t3.medium,CreditSpecification={CpuCredits=unlimited}
```

Associating an Auto Scaling group with a launch template

To associate the launch template with an Auto Scaling group, create the Auto Scaling group using the launch template, or add the launch template to an existing Auto Scaling group.

To create an Auto Scaling group using a launch template (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation bar at the top of the screen, select the same Region that you used when you created the launch template.
3. In the navigation pane, choose **Auto Scaling Groups**, **Create Auto Scaling group**.
4. Choose **Launch Template**, select your launch template, and then choose **Next Step**.
5. Complete the fields for the Auto Scaling group. When you've finished reviewing your configuration settings on the **Review page**, choose **Create Auto Scaling group**. For more information, see [Creating an Auto Scaling Group Using a Launch Template](#) in the *Amazon EC2 Auto Scaling User Guide*.

To create an Auto Scaling group using a launch template (AWS CLI)

Use the [create-auto-scaling-group](#) AWS CLI command and specify the --launch-template parameter.

To add a launch template to an existing Auto Scaling group (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation bar at the top of the screen, select the same Region that you used when you created the launch template.
3. In the navigation pane, choose **Auto Scaling Groups**.
4. From the Auto Scaling group list, select an Auto Scaling group, and choose **Actions**, **Edit**.
5. On the **Details** tab, for **Launch Template**, choose a launch template, and then choose **Save**.

To add a launch template to an existing Auto Scaling group (AWS CLI)

Use the [update-auto-scaling-group](#) AWS CLI command and specify the --launch-template parameter.

[Viewing the credit specification of a burstable performance instance](#)

You can view the credit specification (unlimited or standard) of a running or stopped instance.

New console

To view the credit specification of a burstable instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances**.
3. Select the instance.
4. Choose **Details** and view the **Credit specification** field. The value is either **unlimited** or **standard**.

Old console

To view the credit specification of a burstable instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances**.
3. Select the instance.
4. Choose **Description** and view the **T2/T3 Unlimited** field.
 - If the value is **Enabled**, then your instance is configured as **unlimited**.
 - If the value is **Disabled**, then your instance is configured as **standard**.

To describe the credit specification of a burstable performance instance (AWS CLI)

Use the [describe-instance-credit-specifications](#) command. If you do not specify one or more instance IDs, all instances with the credit specification of `unlimited` are returned, as well as instances that were previously configured with the `unlimited` credit specification. For example, if you resize a T3 instance to an M4 instance, while it is configured as `unlimited`, Amazon EC2 returns the M4 instance.

Example

```
aws ec2 describe-instance-credit-specifications --instance-id i-1234567890abcdef0
```

The following is example output:

```
{  
    "InstanceCreditSpecifications": [  
        {  
            "InstanceId": "i-1234567890abcdef0",  
            "CpuCredits": "unlimited"  
        }  
    ]  
}
```

Modifying the credit specification of a burstable performance instance

You can switch the credit specification of a running or stopped instance at any time between `unlimited` and `standard`.

New console

To modify the credit specification of a burstable performance instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances**.
3. Select the instance. To modify the credit specification for several instances at one time, select all applicable instances.
4. Choose **Actions, Instance settings, Change credit specification**. This option is enabled only if you selected a burstable performance instance.
5. To change the credit specification to `unlimited`, select the check box next to the instance ID. To change the credit specification to `standard`, clear the check box next to the instance ID.

Old console

To modify the credit specification of a burstable performance instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances**.
3. Select the instance. To modify the credit specification for several instances at one time, select all applicable instances.
4. Choose **Actions, Instance Settings, Change T2/T3 Unlimited**. This option is enabled only if you selected a burstable performance instance.
5. The current credit specification appears in parentheses after the instance ID. To change the credit specification to `unlimited`, choose **Enable**. To change the credit specification to `standard`, choose **Disable**.

To modify the credit specification of a burstable performance instance (AWS CLI)

Use the [modify-instance-credit-specification](#) command. Specify the instance and its credit specification using the --instance-credit-specification parameter. Valid credit specifications are **unlimited** and **standard**.

Example

```
aws ec2 modify-instance-credit-specification --region us-east-1 --instance-credit-specification "InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

The following is example output:

```
{  
    "SuccessfulInstanceCreditSpecifications": [  
        {  
            "InstanceId": "i- 1234567890abcdef0"  
        }  
    ],  
    "UnsuccessfulInstanceCreditSpecifications": []  
}
```

Setting the default credit specification for the account

You can set the default credit specification at the account level per AWS Region. You specify the default credit specification per instance family (for example, T2 or T3).

If you use the Launch Instance Wizard in the AWS Management Console to launch instances, the value you select for the credit specification overrides the account-level default credit specification. If you use the AWS CLI to launch instances, all new burstable performance instances in the account launch using the default credit option. The credit specification for existing running or stopped instances is not affected.

The [modify-default-credit-specification](#) API is an asynchronous operation, which works at an AWS Region level and modifies the credit option for each Availability Zone. All zones in a Region are updated within five minutes. But if instances are launched during this operation, they might not get the new credit option until the zone is updated. To verify whether the update has occurred, you can call [get-default-credit-specification](#) and check the default credit specification for updates. For more information, see [Viewing the default credit specification \(p. 249\)](#).

Consideration

The default credit specification for an instance family can be modified only once in a rolling 5-minute period, and up to four times in a rolling 24-hour period.

To set the default credit specification at the account level (AWS CLI)

Use the [modify-default-credit-specification](#) command. Specify the AWS Region, instance family, and the default credit specification using the --cpu-credits parameter. Valid default credit specifications are **unlimited** and **standard**.

```
aws ec2 modify-default-credit-specification --region us-east-1 --instance-family t2 --cpu-credits unlimited
```

Viewing the default credit specification

You can view the default credit specification of a burstable performance instance family at the account level per AWS Region.

To view the default credit specification at the account level (AWS CLI)

Use the [get-default-credit-specification](#) command. Specify the AWS Region and instance family.

```
aws ec2 get-default-credit-specification --region us-east-1 --instance-family t2
```

Monitoring your CPU credits

You can see the credit balance for each instance in the Amazon EC2 per-instance metrics of the CloudWatch console.

Contents

- [Additional CloudWatch metrics for burstable performance instances \(p. 250\)](#)
- [Calculating CPU credit usage \(p. 251\)](#)

Additional CloudWatch metrics for burstable performance instances

Burstable performance instances have these additional CloudWatch metrics, which are updated every five minutes:

- **CPUCreditUsage** – The number of CPU credits spent during the measurement period.
- **CPUCreditBalance** – The number of CPU credits that an instance has accrued. This balance is depleted when the CPU bursts and CPU credits are spent more quickly than they are earned.
- **CPUSurplusCreditBalance** – The number of surplus CPU credits spent to sustain CPU utilization when the **CPUCreditBalance** value is zero.
- **CPUSurplusCreditsCharged** – The number of surplus CPU credits exceeding the [maximum number of CPU credits \(p. 221\)](#) that can be earned in a 24-hour period, and thus attracting an additional charge.

The last two metrics apply only to instances configured as **unlimited**.

The following table describes the CloudWatch metrics for burstable performance instances. For more information, see [List the available CloudWatch metrics for your instances \(p. 717\)](#).

Metric	Description
CPUCreditUsage	<p>The number of CPU credits spent by the instance for CPU utilization. One CPU credit equals one vCPU running at 100% utilization for one minute or an equivalent combination of vCPUs, utilization, and time (for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilization for two minutes).</p> <p>CPU credit metrics are available at a five-minute frequency only. If you specify a period greater than five minutes, use the Sum statistic instead of the Average statistic.</p> <p>Units: Credits (vCPU-minutes)</p>
CPUCreditBalance	<p>The number of earned CPU credits that an instance has accrued since it was launched or started. For T2 Standard, the CPUCreditBalance also includes the number of launch credits that have been accrued.</p> <p>Credits are accrued in the credit balance after they are earned, and removed from the credit balance when they are spent. The credit balance has a maximum limit, determined by the instance size. After the limit is reached, any new credits that are earned are</p>

Metric	Description
	<p>discarded. For T2 Standard, launch credits do not count towards the limit.</p> <p>The credits in the <code>CPUCreditBalance</code> are available for the instance to spend to burst beyond its baseline CPU utilization.</p> <p>When an instance is running, credits in the <code>CPUCreditBalance</code> do not expire. When a T3 or T4g instance stops, the <code>CPUCreditBalance</code> value persists for seven days. Thereafter, all accrued credits are lost. When a T2 instance stops, the <code>CPUCreditBalance</code> value does not persist, and all accrued credits are lost.</p> <p>CPU credit metrics are available at a five-minute frequency only.</p> <p>Units: Credits (vCPU-minutes)</p>
<code>CPUSurplusCreditBalance</code>	<p>The number of surplus credits that have been spent by an <code>unlimited</code> instance when its <code>CPUCreditBalance</code> value is zero.</p> <p>The <code>CPUSurplusCreditBalance</code> value is paid down by earned CPU credits. If the number of surplus credits exceeds the maximum number of credits that the instance can earn in a 24-hour period, the spent surplus credits above the maximum incur an additional charge.</p> <p>Units: Credits (vCPU-minutes)</p>
<code>CPUSurplusCreditsCharged</code>	<p>The number of spent surplus credits that are not paid down by earned CPU credits, and which thus incur an additional charge.</p> <p>Spent surplus credits are charged when any of the following occurs:</p> <ul style="list-style-type: none"> • The spent surplus credits exceed the maximum number of credits that the instance can earn in a 24-hour period. Spent surplus credits above the maximum are charged at the end of the hour. • The instance is stopped or terminated. • The instance is switched from <code>unlimited</code> to <code>standard</code>. <p>Units: Credits (vCPU-minutes)</p>

Calculating CPU credit usage

The CPU credit usage of instances is calculated using the instance CloudWatch metrics described in the preceding table.

Amazon EC2 sends the metrics to CloudWatch every five minutes. A reference to the *prior* value of a metric at any point in time implies the previous value of the metric, sent *five minutes ago*.

Calculating CPU credit usage for Standard instances

- The CPU credit balance increases if CPU utilization is below the baseline, when the credits spent are less than the credits earned in the prior five-minute interval.
- The CPU credit balance decreases if CPU utilization is above the baseline, when the credits spent are more than the credits earned in the prior five-minute interval.

Mathematically, this is captured by the following equation:

Example

```
CPUCreditBalance = prior CPUCreditBalance + [Credits earned per hour * (5/60) -  
CPUCreditUsage]
```

The size of the instance determines the number of credits that the instance can earn per hour and the number of earned credits that it can accrue in the credit balance. For information about the number of credits earned per hour, and the credit balance limit for each instance size, see the [credit table \(p. 221\)](#).

Example

This example uses a t3.nano instance. To calculate the CPUCreditBalance value of the instance, use the preceding equation as follows:

- CPUCreditBalance – The current credit balance to calculate.
- prior CPUCreditBalance – The credit balance five minutes ago. In this example, the instance had accrued two credits.
- Credits earned per hour – A t3.nano instance earns six credits per hour.
- 5/60 – Represents the five-minute interval between CloudWatch metric publication. Multiply the credits earned per hour by 5/60 (five minutes) to get the number of credits that the instance earned in the past five minutes. A t3.nano instance earns 0.5 credits every five minutes.
- CPUCreditUsage – How many credits the instance spent in the past five minutes. In this example, the instance spent one credit in the past five minutes.

Using these values, you can calculate the CPUCreditBalance value:

Example

```
CPUCreditBalance = 2 + [0.5 - 1] = 1.5
```

[Calculating CPU credit usage for Unlimited instances](#)

When a burstable performance instance needs to burst above the baseline, it always spends accrued credits before spending surplus credits. When it depletes its accrued CPU credit balance, it can spend surplus credits to burst CPU for as long as it needs. When CPU utilization falls below the baseline, surplus credits are always paid down before the instance accrues earned credits.

We use the term `Adjusted balance` in the following equations to reflect the activity that occurs in this five-minute interval. We use this value to arrive at the values for the CPUCreditBalance and CPUSurplusCreditBalance CloudWatch metrics.

Example

```
Adjusted balance = [prior CPUCreditBalance - prior CPUSurplusCreditBalance] + [Credits  
earned per hour * (5/60) - CPUCreditUsage]
```

A value of 0 for `Adjusted balance` indicates that the instance spent all its earned credits for bursting, and no surplus credits were spent. As a result, both CPUCreditBalance and CPUSurplusCreditBalance are set to 0.

A positive `Adjusted balance` value indicates that the instance accrued earned credits, and previous surplus credits, if any, were paid down. As a result, the `Adjusted balance` value is assigned to CPUCreditBalance, and the CPUSurplusCreditBalance is set to 0. The instance size determines the [maximum number of credits \(p. 221\)](#) that it can accrue.

Example

```
CPUCreditBalance = min [max earned credit balance, Adjusted balance]  
CPUSurplusCreditBalance = 0
```

A negative `Adjusted balance` value indicates that the instance spent all its earned credits that it accrued and, in addition, also spent surplus credits for bursting. As a result, the `Adjusted balance` value is assigned to `CPUSurplusCreditBalance` and `CPUCreditBalance` is set to 0. Again, the instance size determines the [maximum number of credits \(p. 221\)](#) that it can accrue.

Example

```
CPUSurplusCreditBalance = min [max earned credit balance, -Adjusted balance]  
CPUCreditBalance = 0
```

If the surplus credits spent exceed the maximum credits that the instance can accrue, the surplus credit balance is set to the maximum, as shown in the preceding equation. The remaining surplus credits are charged as represented by the `CPUSurplusCreditsCharged` metric.

Example

```
CPUSurplusCreditsCharged = max [-Adjusted balance - max earned credit balance, 0]
```

Finally, when the instance terminates, any surplus credits tracked by the `CPUSurplusCreditBalance` are charged. If the instance is switched from `unlimited` to `standard`, any remaining `CPUSurplusCreditBalance` is also charged.

Compute optimized instances

Compute optimized instances are ideal for compute-bound applications that benefit from high-performance processors.

C5 and C5n instances

These instances are well suited for the following:

- Batch processing workloads
- Media transcoding
- High-performance web servers
- High-performance computing (HPC)
- Scientific modeling
- Dedicated gaming servers and ad serving engines
- Machine learning inference and other compute-intensive applications

Bare metal instances, such as `c5.metal`, provide your applications with direct access to physical resources of the host server, such as processors and memory.

For more information, see [Amazon EC2 C5 Instances](#).

C6g and C6gd instances

These instances are powered by AWS Graviton2 processors and are ideal for running advanced, compute-intensive workloads, such as the following:

- High-performance computing (HPC)
- Batch processing

- Ad serving
- Video encoding
- Gaming servers
- Scientific modeling
- Distributed analytics
- CPU-based machine learning inference

Bare metal instances, such as `c6g.metal`, provide your applications with direct access to physical resources of the host server, such as processors and memory.

For more information, see [Amazon EC2 C6g Instances](#).

Contents

- [Hardware specifications \(p. 254\)](#)
- [Instance performance \(p. 256\)](#)
- [Network performance \(p. 256\)](#)
- [SSD I/O performance \(p. 257\)](#)
- [Instance features \(p. 259\)](#)
- [Release notes \(p. 259\)](#)

Hardware specifications

The following is a summary of the hardware specifications for compute optimized instances.

Instance type	Default vCPUs	Memory (GiB)
<code>c4.large</code>	2	3.75
<code>c4.xlarge</code>	4	7.5
<code>c4.2xlarge</code>	8	15
<code>c4.4xlarge</code>	16	30
<code>c4.8xlarge</code>	36	60
<code>c5.large</code>	2	4
<code>c5.xlarge</code>	4	8
<code>c5.2xlarge</code>	8	16
<code>c5.4xlarge</code>	16	32
<code>c5.9xlarge</code>	36	72
<code>c5.12xlarge</code>	48	96
<code>c5.18xlarge</code>	72	144
<code>c5.24xlarge</code>	96	192
<code>c5.metal</code>	96	192
<code>c5a.large</code>	2	4

Instance type	Default vCPUs	Memory (GiB)
c5a.xlarge	4	8
c5a.2xlarge	8	16
c5a.4xlarge	16	32
c5a.8xlarge	32	64
c5a.12xlarge	48	96
c5a.16xlarge	64	128
c5a.24xlarge	96	192
c5ad.large	2	4
c5ad.xlarge	4	8
c5ad.2xlarge	8	16
c5ad.4xlarge	16	32
c5ad.8xlarge	32	64
c5ad.12xlarge	48	96
c5ad.16xlarge	64	128
c5ad.24xlarge	96	192
c5d.large	2	4
c5d.xlarge	4	8
c5d.2xlarge	8	16
c5d.4xlarge	16	32
c5d.9xlarge	36	72
c5d.12xlarge	48	96
c5d.18xlarge	72	144
c5d.24xlarge	96	192
c5d.metal	96	192
c5n.large	2	5.25
c5n.xlarge	4	10.5
c5n.2xlarge	8	21
c5n.4xlarge	16	42
c5n.9xlarge	36	96
c5n.18xlarge	72	192
c5n.metal	72	192

Instance type	Default vCPUs	Memory (GiB)
c6g.medium	1	2
c6g.large	2	4
c6g.xlarge	4	8
c6g.2xlarge	8	16
c6g.4xlarge	16	32
c6g.8xlarge	32	64
c6g.12xlarge	48	96
c6g.16xlarge	64	128
c6g.metal	64	128
c6gd.medium	1	2
c6gd.large	2	4
c6gd.xlarge	4	8
c6gd.2xlarge	8	16
c6gd.4xlarge	16	32
c6gd.8xlarge	32	64
c6gd.12xlarge	48	96
c6gd.16xlarge	64	128
c6gd.metal	64	128

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instance Types](#).

For more information about specifying CPU options, see [Optimizing CPU options \(p. 632\)](#).

Instance performance

EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your instance. Some compute optimized instances are EBS-optimized by default at no additional cost. For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#).

Some compute optimized instance types provide the ability to control processor C-states and P-states on Linux. C-states control the sleep levels that a core can enter when it is inactive, while P-states control the desired performance (in CPU frequency) from a core. For more information, see [Processor state control for your EC2 instance \(p. 622\)](#).

Network performance

You can enable enhanced networking on supported instance types to provide lower latencies, lower network jitter, and higher packet-per-second (PPS) performance. Most applications do not consistently

need a high level of network performance, but can benefit from access to increased bandwidth when they send or receive data. For more information, see [Enhanced networking on Linux \(p. 813\)](#).

The following is a summary of network performance for compute optimized instances that support enhanced networking.

Instance type	Network performance	Enhanced networking
c5.4xlarge and smaller c5d.4xlarge and smaller c6g.4xlarge and smaller c6gd.4xlarge and smaller	Up to 10 Gbps †	ENAs (p. 814)
c5.9xlarge c5d.9xlarge	10 Gbps	ENAs (p. 814)
c5.12xlarge c5d.12xlarge c6g.8xlarge c6gd.8xlarge	12 Gbps	ENAs (p. 814)
c6g.12xlarge c6gd.12xlarge	20 Gbps	ENAs (p. 814)
c5n.4xlarge and smaller	Up to 25 Gbps †	ENAs (p. 814)
c5.18xlarge c5.24xlarge c5.metal c5d.18xlarge c5d.24xlarge c5d.metal c6g.16xlarge c6g.metal c6gd.16xlarge c6gd.metal	25 Gbps	ENAs (p. 814)
c5n.9xlarge	50 Gbps	ENAs (p. 814)
c5n.18xlarge c5n.metal	100 Gbps	ENAs (p. 814)
c4.large	Moderate	Intel 82599 VF (p. 826)
c4.xlarge c4.2xlarge c4.4xlarge	High	Intel 82599 VF (p. 826)
c4.8xlarge	10 Gbps	Intel 82599 VF (p. 826)

† These instances use a network I/O credit mechanism to allocate network bandwidth to instances based on average bandwidth utilization. They accrue credits when their bandwidth is below their baseline bandwidth, and can use these credits when they perform network data transfers. For more information, open a support case and ask about baseline bandwidth for the specific instance types that you are interested in.

SSD I/O performance

If you use a Linux AMI with kernel version 4.4 or later and use all the SSD-based instance store volumes available to your instance, you get the IOPS (4,096 byte block size) performance listed in the following table (at queue depth saturation). Otherwise, you get lower IOPS performance.

Instance Size	100% Random Read IOPS	Write IOPS
c5ad.large	16,283	7,105
c5ad.xlarge	32,566	14,211

Instance Size	100% Random Read IOPS	Write IOPS
c5ad.2xlarge	65,132	28,421
c5ad.4xlarge	130,263	56,842
c5ad.8xlarge	260,526	113,684
c5ad.12xlarge	412,500	180,000
c5ad.16xlarge	521,053	227,368
c5ad.24xlarge	825,000	360,000
c5d.large *	20,000	9,000
c5d.xlarge *	40,000	18,000
c5d.2xlarge *	80,000	37,000
c5d.4xlarge *	175,000	75,000
c5d.9xlarge	350,000	170,000
c5d.12xlarge	700,000	340,000
c5d.18xlarge	700,000	340,000
c5d.24xlarge	1,400,000	680,000
c5d.metal	1,400,000	680,000
c6gd.medium	13,438	5,625
c6gd.large	26,875	11,250
c6gd.xlarge	53,750	22,500
c6gd.2xlarge	107,500	45,000
c6gd.4xlarge	215,000	90,000
c6gd.8xlarge	430,000	180,000
c6gd.12xlarge	645,000	270,000
c6gd.16xlarge	860,000	360,000
c6gd.metal	860,000	360,000

* For these instances, you can get up to the specified performance.

As you fill the SSD-based instance store volumes for your instance, the number of write IOPS that you can achieve decreases. This is due to the extra work the SSD controller must do to find available space, rewrite existing data, and erase unused space so that it can be rewritten. This process of garbage collection results in internal write amplification to the SSD, expressed as the ratio of SSD write operations to user write operations. This decrease in performance is even larger if the write operations are not in multiples of 4,096 bytes or not aligned to a 4,096-byte boundary. If you write a smaller amount of bytes or bytes that are not aligned, the SSD controller must read the surrounding data and store the result in a new location. This pattern results in significantly increased write amplification, increased latency, and dramatically reduced I/O performance.

SSD controllers can use several strategies to reduce the impact of write amplification. One such strategy is to reserve space in the SSD instance storage so that the controller can more efficiently manage the space available for write operations. This is called *over-provisioning*. The SSD-based instance store volumes provided to an instance don't have any space reserved for over-provisioning. To reduce write amplification, we recommend that you leave 10% of the volume unpartitioned so that the SSD controller can use it for over-provisioning. This decreases the storage that you can use, but increases performance even if the disk is close to full capacity.

For instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see [Instance store volume TRIM support \(p. 1201\)](#).

Instance features

The following is a summary of features for compute optimized instances:

	EBS only	NVMe EBS	Instance store	Placement group
C4	Yes	No	No	Yes
C5	Yes	Yes	No	Yes
C5a	Yes	Yes	No	Yes
C5ad	No	Yes	NVMe *	Yes
C5d	No	Yes	NVMe *	Yes
C5n	Yes	Yes	No	Yes
C6g	Yes	Yes	No	Yes
C6gd	No	Yes	NVMe *	Yes

* The root device volume must be an Amazon EBS volume.

For more information, see the following:

- [Amazon EBS and NVMe on Linux instances \(p. 1136\)](#)
- [Amazon EC2 instance store \(p. 1188\)](#)
- [Placement groups \(p. 869\)](#)

Release notes

- C5 and C5d instances feature a 3.1 GHz Intel Xeon Platinum 8000 series processor from either the first generation (Skylake-SP) or second generation (Cascade Lake).
- C5a and C5ad instances feature a second-generation AMD EPYC processor (Rome) running at frequencies as high as 3.3. GHz.
- C6g and C6gd instances feature an AWS Graviton2 processor based on 64-bit Arm architecture.
- C4 instances and instances based on the [Nitro System \(p. 205\)](#) require 64-bit EBS-backed HVM AMIs. They have high-memory and require a 64-bit operating system to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. In addition, you must use an HVM AMI to take advantage of enhanced networking.
- Instances built on the Nitro System have the following requirements:

- NVMe drivers (p. 1136) must be installed
- Elastic Network Adapter (ENA) drivers (p. 814) must be installed

The following Linux AMIs meet these requirements:

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (with `linux-aws` kernel) or later
- Red Hat Enterprise Linux 7.4 or later
- SUSE Linux Enterprise Server 12 SP2 or later
- CentOS 7.4.1708 or later
- FreeBSD 11.1 or later
- Debian GNU/Linux 9 or later
- Instances with an AWS Graviton processors have the following requirements:
 - Use an AMI for the 64-bit Arm architecture.
 - Support booting through UEFI with ACPI tables and support ACPI hot-plug of PCI devices.

The following AMIs meet these requirements:

- Amazon Linux 2 (64-bit Arm)
- Ubuntu 16.04 or later (64-bit Arm)
- Red Hat Enterprise Linux 8.0 or later (64-bit Arm)
- SUSE Linux Enterprise Server 15 or later (64-bit Arm)
- Debian 10 or later (64-bit Arm)
- Instances built on the Nitro System instances support a maximum of 28 attachments, including network interfaces, EBS volumes, and NVMe instance store volumes. For more information, see [Nitro System volume limits \(p. 1209\)](#).
- Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
- To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support. Amazon Linux 2 and the latest versions of the Amazon Linux AMI support PCIe native hotplug, but earlier versions do not. You must enable the following Linux kernel configuration options:

```
CONFIG_HOTPLUG_PCI_PCIE=y  
CONFIG_PCIEASPM=y
```

- Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
- Instances built on the Nitro System should have acpid installed to support clean shutdown through API requests.
- There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see [How many instances can I run in Amazon EC2?](#) in the Amazon EC2 FAQ.

Memory optimized instances

Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory.

R5, R5a, and R5n instances

These instances are well suited for the following:

- High-performance, relational (MySQL) and NoSQL (MongoDB, Cassandra) databases.
- Distributed web scale cache stores that provide in-memory caching of key-value type data (Memcached and Redis).
- In-memory databases using optimized data storage formats and analytics for business intelligence (for example, SAP HANA).
- Applications performing real-time processing of big unstructured data (financial services, Hadoop/Spark clusters).
- High-performance computing (HPC) and Electronic Design Automation (EDA) applications.

Bare metal instances, such as `r5.metal`, provide your applications with direct access to physical resources of the host server, such as processors and memory. These instances are well suited for the following:

- Workloads that require access to low-level hardware features (for example, Intel VT) that are not available or fully supported in virtualized environments
- Applications that require a non-virtualized environment for licensing or support

For more information, see [Amazon EC2 R5 Instances](#).

R6g and R6gd instances

These instances are powered by AWS Graviton2 processors and are ideal for running memory-intensive workloads, such as the following:

- Open-source databases (for example, MySQL, MariaDB, and PostgreSQL)
- In-memory caches (for example, Memcached, Redis, and KeyDB)

Bare metal instances, such as `r6g.metal`, provide your applications with direct access to physical resources of the host server, such as processors and memory.

For more information, see [Amazon EC2 R6g Instances](#).

High memory instances

High memory instances (`u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal`, and `u-24tb1.metal`) offer 6 TiB, 9 TiB, 12 TiB, 18 TiB, and 24 TiB of memory per instance. These instances are designed to run large in-memory databases, including production deployments of the SAP HANA in-memory database, in the cloud. They offer bare metal performance with direct access to host hardware.

For more information, see [Amazon EC2 High Memory Instances and Storage Configuration for SAP HANA](#).

X1 instances

These instances are well suited for the following:

- In-memory databases such as SAP HANA, including SAP-certified support for Business Suite S/4HANA, Business Suite on HANA (SoH), Business Warehouse on HANA (BW), and Data Mart Solutions on HANA. For more information, see [SAP HANA on the AWS Cloud](#).
- Big-data processing engines such as Apache Spark or Presto.
- High-performance computing (HPC) applications.

For more information, see [Amazon EC2 X1 Instances](#).

X1e instances

These instances are well suited for the following:

- High-performance databases.
- In-memory databases such as SAP HANA. For more information, see [SAP HANA on the AWS Cloud](#).
- Memory-intensive enterprise applications.

For more information, see [Amazon EC2 X1e Instances](#).

z1d instances

These instances deliver both high compute and high memory and are well-suited for the following:

- Electronic Design Automation (EDA)
- Relational database workloads

`z1d.metal` instances provide your applications with direct access to physical resources of the host server, such as processors and memory. These instances are well suited for the following:

- Workloads that require access to low-level hardware features (for example, Intel VT) that are not available or fully supported in virtualized environments
- Applications that require a non-virtualized environment for licensing or support

For more information, see [Amazon EC2 z1d Instances](#).

Contents

- [Hardware specifications \(p. 262\)](#)
- [Memory performance \(p. 266\)](#)
- [Instance performance \(p. 266\)](#)
- [Network performance \(p. 266\)](#)
- [SSD I/O performance \(p. 267\)](#)
- [Instance features \(p. 269\)](#)
- [Support for vCPUs \(p. 270\)](#)
- [Release notes \(p. 270\)](#)

Hardware specifications

The following is a summary of the hardware specifications for memory optimized instances.

Instance type	Default vCPUs	Memory (GiB)
r4.large	2	15.25
r4.xlarge	4	30.5
r4.2xlarge	8	61
r4.4xlarge	16	122
r4.8xlarge	32	244

Instance type	Default vCPUs	Memory (GiB)
r4.16xlarge	64	488
r5.large	2	16
r5.xlarge	4	32
r5.2xlarge	8	64
r5.4xlarge	16	128
r5.8xlarge	32	256
r5.12xlarge	48	384
r5.16xlarge	64	512
r5.24xlarge	96	768
r5.metal	96	768
r5a.large	2	16
r5a.xlarge	4	32
r5a.2xlarge	8	64
r5a.4xlarge	16	128
r5a.8xlarge	32	256
r5a.12xlarge	48	384
r5a.16xlarge	64	512
r5a.24xlarge	96	768
r5ad.large	2	16
r5ad.xlarge	4	32
r5ad.2xlarge	8	64
r5ad.4xlarge	16	128
r5ad.8xlarge	32	256
r5ad.12xlarge	48	384
r5ad.16xlarge	64	512
r5ad.24xlarge	96	768
r5d.large	2	16
r5d.xlarge	4	32
r5d.2xlarge	8	64
r5d.4xlarge	16	128
r5d.8xlarge	32	256

Instance type	Default vCPUs	Memory (GiB)
r5d.12xlarge	48	384
r5d.16xlarge	64	512
r5d.24xlarge	96	768
r5d.metal	96	768
r5dn.large	2	16
r5dn.xlarge	4	32
r5dn.2xlarge	8	64
r5dn.4xlarge	16	128
r5dn.8xlarge	32	256
r5dn.12xlarge	48	384
r5dn.16xlarge	64	512
r5dn.24xlarge	96	768
r5n.large	2	16
r5n.xlarge	4	32
r5n.2xlarge	8	64
r5n.4xlarge	16	128
r5n.8xlarge	32	256
r5n.12xlarge	48	384
r5n.16xlarge	64	512
r5n.24xlarge	96	768
r6g.medium	1	8
r6g.large	2	16
r6g.xlarge	4	32
r6g.2xlarge	8	64
r6g.4xlarge	16	128
r6g.8xlarge	32	256
r6g.12xlarge	48	384
r6g.16xlarge	64	512
r6gd.medium	1	8
r6gd.large	2	16
r6gd.xlarge	4	32

Instance type	Default vCPUs	Memory (GiB)
r6gd.2xlarge	8	64
r6gd.4xlarge	16	128
r6gd.8xlarge	32	256
r6gd.12xlarge	48	384
r6gd.16xlarge	64	512
u-6tb1.metal	448 *	6,144
u-9tb1.metal	448 *	9,216
u-12tb1.metal	448 *	12,288
u-18tb1.metal	448 *	18,432
u-24tb1.metal	448 *	24,576
x1.16xlarge	64	976
x1.32xlarge	128	1,952
xle.xlarge	4	122
xle.2xlarge	8	244
xle.4xlarge	16	488
xle.8xlarge	32	976
xle.16xlarge	64	1,952
xle.32xlarge	128	3,904
z1d.large	2	16
z1d.xlarge	4	32
z1d.2xlarge	8	64
z1d.3xlarge	12	96
z1d.6xlarge	24	192
z1d.12xlarge	48	384
z1d.metal	48	384

* Each logical processor is a hyperthread on 224 cores.

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instance Types](#).

For more information about specifying CPU options, see [Optimizing CPU options \(p. 632\)](#).

Memory performance

X1 instances include Intel Scalable Memory Buffers, providing 300 GiB/s of sustainable memory-read bandwidth and 140 GiB/s of sustainable memory-write bandwidth.

For more information about how much RAM can be enabled for memory optimized instances, see [Hardware specifications \(p. 262\)](#).

Memory optimized instances have high memory and require 64-bit HVM AMIs to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on memory optimized instances. For more information, see [Linux AMI virtualization types \(p. 102\)](#).

Instance performance

Memory optimized instances enable increased cryptographic performance through the latest Intel AES-NI feature, support Intel Transactional Synchronization Extensions (TSX) to boost the performance of in-memory transactional data processing, and support Advanced Vector Extensions 2 (Intel AVX2) processor instructions to expand most integer commands to 256 bits.

Some memory optimized instances provide the ability to control processor C-states and P-states on Linux. C-states control the sleep levels that a core can enter when it is inactive, while P-states control the desired performance (measured by CPU frequency) from a core. For more information, see [Processor state control for your EC2 instance \(p. 622\)](#).

Network performance

You can enable enhanced networking on supported instance types to provide lower latencies, lower network jitter, and higher packet-per-second (PPS) performance. Most applications do not consistently need a high level of network performance, but can benefit from access to increased bandwidth when they send or receive data. For more information, see [Enhanced networking on Linux \(p. 813\)](#).

The following is a summary of network performance for memory optimized instances that support enhanced networking.

Instance type	Network performance	Enhanced networking
r4.4xlarge and smaller r5.4xlarge and smaller r5a.8xlarge and smaller r5ad.8xlarge and smaller r5d.4xlarge and smaller r6g.4xlarge and smaller r6gd.4xlarge and smaller x1e.8large and smaller z1d.3xlarge and smaller	Up to 10 Gbps †	ENA (p. 814)
r4.8xlarge r5.8xlarge r5.12xlarge r5a.12xlarge r5ad.12xlarge r5d.8xlarge r5d.12xlarge x1.16xlarge x1e.16xlarge z1d.6xlarge	10 Gbps	ENA (p. 814)
r5a.16xlarge r5ad.16xlarge r6g.8xlarge r6gd.8xlarge	12 Gbps	ENA (p. 814)
r5.16xlarge r5a.24xlarge r5ad.24xlarge r5d.16xlarge r6g.12xlarge r6gd.12xlarge	20 Gbps	ENA (p. 814)
r5dn.4xlarge and smaller r5n.4xlarge and smaller	Up to 25 Gbps †	ENA (p. 814)

Instance type	Network performance	Enhanced networking
r4.16xlarge r5.24xlarge r5.metal r5d.24xlarge r5d.metal r5dn.8xlarge r5n.8xlarge r6g.16xlarge r6g.metal r6gd.16xlarge r6gd.metal x1.32xlarge x1e.32xlarge z1d.12xlarge z1d.metal	25 Gbps	ENAs (p. 814)
r5dn.12xlarge r5n.12xlarge	50 Gbps	ENAs (p. 814)
r5dn.16xlarge r5n.16xlarge	75 Gbps	ENAs (p. 814)
r5dn.24xlarge r5n.24xlarge u-6tb1.metal * u-9tb1.metal * u-12tb1.metal * u-18tb1.metal u-24tb1.metal	100 Gbps	ENAs (p. 814)

* Instances of this type launched after March 12, 2020 provide network performance of 100 Gbps. Instances of this type launched before March 12, 2020 might only provide network performance of 25 Gbps. To ensure that instances launched before March 12, 2020 have a network performance of 100 Gbps, contact your account team to upgrade your instance at no additional cost.

† These instances use a network I/O credit mechanism to allocate network bandwidth to instances based on average bandwidth utilization. They accrue credits when their bandwidth is below their baseline bandwidth, and can use these credits when they perform network data transfers. For more information, open a support case and ask about baseline bandwidth for the specific instance types that you are interested in.

SSD I/O performance

If you use a Linux AMI with kernel version 4.4 or later and use all the SSD-based instance store volumes available to your instance, you get the IOPS (4,096 byte block size) performance listed in the following table (at queue depth saturation). Otherwise, you get lower IOPS performance.

Instance Size	100% Random Read IOPS	Write IOPS
r5ad.large *	30,000	15,000
r5ad.xlarge *	59,000	29,000
r5ad.2xlarge *	117,000	57,000
r5ad.4xlarge *	234,000	114,000
r5ad.8xlarge	466,666	233,333
r5ad.12xlarge	700,000	340,000
r5ad.16xlarge	933,333	466,666
r5ad.24xlarge	1,400,000	680,000
r5d.large *	30,000	15,000
r5d.xlarge *	59,000	29,000
r5d.2xlarge *	117,000	57,000
r5d.4xlarge *	234,000	114,000

Instance Size	100% Random Read IOPS	Write IOPS
r5d.8xlarge	466,666	233,333
r5d.12xlarge	700,000	340,000
r5d.16xlarge	933,333	466,666
r5d.24xlarge	1,400,000	680,000
r5d.metal	1,400,000	680,000
r5dn.large *	30,000	15,000
r5dn.xlarge *	59,000	29,000
r5dn.2xlarge *	117,000	57,000
r5dn.4xlarge *	234,000	114,000
r5dn.8xlarge	466,666	233,333
r5dn.12xlarge	700,000	340,000
r5dn.16xlarge	933,333	466,666
r5dn.24xlarge	1,400,000	680,000
r6gd.medium	13,438	5,625
r6gd.large	26,875	11,250
r6gd.xlarge	53,750	22,500
r6gd.2xlarge	107,500	45,000
r6gd.4xlarge	215,000	90,000
r6gd.8xlarge	430,000	180,000
r6gd.12xlarge	645,000	270,000
r6gd.16xlarge	860,000	360,000
r6gd.metal	860,000	360,000
z1d.large *	30,000	15,000
z1d.xlarge *	59,000	29,000
z1d.2xlarge *	117,000	57,000
z1d.3xlarge *	175,000	75,000
z1d.6xlarge	350,000	170,000
z1d.12xlarge	700,000	340,000
z1d.metal	700,000	340,000

* For these instances, you can get up to the specified performance.

As you fill the SSD-based instance store volumes for your instance, the number of write IOPS that you can achieve decreases. This is due to the extra work the SSD controller must do to find available space, rewrite existing data, and erase unused space so that it can be rewritten. This process of garbage collection results in internal write amplification to the SSD, expressed as the ratio of SSD write operations to user write operations. This decrease in performance is even larger if the write operations are not in multiples of 4,096 bytes or not aligned to a 4,096-byte boundary. If you write a smaller amount of bytes or bytes that are not aligned, the SSD controller must read the surrounding data and store the result in a new location. This pattern results in significantly increased write amplification, increased latency, and dramatically reduced I/O performance.

SSD controllers can use several strategies to reduce the impact of write amplification. One such strategy is to reserve space in the SSD instance storage so that the controller can more efficiently manage the space available for write operations. This is called *over-provisioning*. The SSD-based instance store volumes provided to an instance don't have any space reserved for over-provisioning. To reduce write amplification, we recommend that you leave 10% of the volume unpartitioned so that the SSD controller can use it for over-provisioning. This decreases the storage that you can use, but increases performance even if the disk is close to full capacity.

For instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see [Instance store volume TRIM support \(p. 1201\)](#).

Instance features

The following is a summary of features for memory optimized instances.

	EBS only	NVMe EBS	Instance store	Placement group
R4	Yes	No	No	Yes
R5	Yes	Yes	No	Yes
R5a	Yes	Yes	No	Yes
R5ad	No	Yes	NVME *	Yes
R5d	No	Yes	NVME *	Yes
R5dn	No	Yes	NVME *	Yes
R5n	Yes	Yes	No	Yes
R6g	Yes	Yes	No	Yes
R6gd	No	Yes	NVMe *	Yes
u-6tb1.metal	Yes	Yes	No	No
u-9tb1.metal	Yes	Yes	No	No
u-12tb1.metal	Yes	Yes	No	No
u-18tb1.metal	Yes	Yes	No	No
u-24tb1.metal	Yes	Yes	No	No
X1	No	No	SSD	Yes
X1e	No	No	SSD *	Yes

	EBS only	NVMe EBS	Instance store	Placement group
z1d	No	Yes	NVME *	Yes

* The root device volume must be an Amazon EBS volume.

For more information, see the following:

- [Amazon EBS and NVMe on Linux instances \(p. 1136\)](#)
- [Amazon EC2 instance store \(p. 1188\)](#)
- [Placement groups \(p. 869\)](#)

Support for vCPUs

Memory optimized instances provide a high number of vCPUs, which can cause launch issues with operating systems that have a lower vCPU limit. We strongly recommend that you use the latest AMIs when you launch memory optimized instances.

The following AMIs support launching memory optimized instances:

- Amazon Linux 2 (HVM)
- Amazon Linux AMI 2016.03 (HVM) or later
- Ubuntu Server 14.04 LTS (HVM)
- Red Hat Enterprise Linux 7.1 (HVM)
- SUSE Linux Enterprise Server 12 SP1 (HVM)
- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2 64-bit
- Windows Server 2008 SP2 64-bit

Release notes

- R4 instances feature up to 64 vCPUs and are powered by two AWS-customized Intel XEON processors based on E5-2686v4 that feature high-memory bandwidth and larger L3 caches to boost the performance of in-memory applications.
- R5 and R5d instances feature a 3.1 GHz Intel Xeon Platinum 8000 series processor from either the first generation (Skylake-SP) or second generation (Cascade Lake).
- R5a and R5ad instances feature a 2.5 GHz AMD EPYC 7000 series processor.
- R6g and R6gd instances feature an AWS Graviton2 processor based on 64-bit Arm architecture.
- High memory instances (`u-6tb1.metal`, `u-9tb1.metal`, and `u-12tb1.metal`) are the first instances to be powered by an eight-socket platform with the latest generation Intel Xeon Platinum 8176M (Skylake) processors that are optimized for mission-critical enterprise workloads. High Memory instances with 18 TB and 24 TB of memory (`u-18tb1.metal` and `u-24tb1.metal`) are the first instances powered by an 8-socket platform with 2nd Generation Intel Xeon Scalable 8280L (Cascade Lake) processors.
- X1e and X1 instances feature up to 128 vCPUs and are powered by four Intel Xeon E7-8880 v3 processors that feature high-memory bandwidth and larger L3 caches to boost the performance of in-memory applications.

- Instances built on the Nitro System have the following requirements:
 - NVMe drivers (p. 1136) must be installed
 - Elastic Network Adapter (ENA) drivers (p. 814) must be installed

The following Linux AMIs meet these requirements:

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (with `linux-aws` kernel) or later
- Red Hat Enterprise Linux 7.4 or later
- SUSE Linux Enterprise Server 12 SP2 or later
- CentOS 7.4.1708 or later
- FreeBSD 11.1 or later
- Debian GNU/Linux 9 or later
- Instances with an AWS Graviton processors have the following requirements:
 - Use an AMI for the 64-bit Arm architecture.
 - Support booting through UEFI with ACPI tables and support ACPI hot-plug of PCI devices.

The following AMIs meet these requirements:

- Amazon Linux 2 (64-bit Arm)
- Ubuntu 16.04 or later (64-bit Arm)
- Red Hat Enterprise Linux 8.0 or later (64-bit Arm)
- SUSE Linux Enterprise Server 15 or later (64-bit Arm)
- Debian 10 or later (64-bit Arm)
- Instances built on the Nitro System instances support a maximum of 28 attachments, including network interfaces, EBS volumes, and NVMe instance store volumes. For more information, see [Nitro System volume limits \(p. 1209\)](#).
- Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
- To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support. Amazon Linux 2 and the latest versions of the Amazon Linux AMI support PCIe native hotplug, but earlier versions do not. You must enable the following Linux kernel configuration options:

```
CONFIG_HOTPLUG_PCI_PCIE=y  
CONFIG_PCIEASPM=y
```

- Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
- You can't launch X1 instances using a Windows Server 2008 SP2 64-bit AMI, except for `x1.16xlarge` instances.
- You can't launch X1e instances using a Windows Server 2008 SP2 64-bit AMI.
- With earlier versions of the Windows Server 2008 R2 64-bit AMI, you can't launch `r4.1.large` and `r4.4xlarge` instances. If you experience this issue, update to the latest version of this AMI.
- There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see [How many instances can I run in Amazon EC2?](#) in the Amazon EC2 FAQ.

Storage optimized instances

Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage. They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications.

D2 instances

These instances are well suited for the following:

- Massive parallel processing (MPP) data warehouse
- MapReduce and Hadoop distributed computing
- Log or data processing applications

H1 instances

These instances are well suited for the following:

- Data-intensive workloads such as MapReduce and distributed file systems
- Applications requiring sequential access to large amounts of data on direct-attached instance storage
- Applications that require high-throughput access to large quantities of data

I3 and I3en instances

These instances are well suited for the following:

- High frequency online transaction processing (OLTP) systems
- Relational databases
- NoSQL databases
- Cache for in-memory databases (for example, Redis)
- Data warehousing applications
- Distributed file systems

Bare metal instances provide your applications with direct access to physical resources of the host server, such as processors and memory. These instances are well suited for the following:

- Workloads that require access to low-level hardware features (for example, Intel VT) that are not available or fully supported in virtualized environments
- Applications that require a non-virtualized environment for licensing or support

For more information, see [Amazon EC2 I3 Instances](#).

Contents

- [Hardware specifications \(p. 273\)](#)
- [Instance performance \(p. 274\)](#)
- [Network performance \(p. 274\)](#)
- [SSD I/O performance \(p. 275\)](#)
- [Instance features \(p. 276\)](#)
- [Support for vCPUs \(p. 276\)](#)

- [Release notes \(p. 277\)](#)

Hardware specifications

The primary data storage for D2 instances is HDD instance store volumes. The primary data storage for I3 and I3en instances is non-volatile memory express (NVMe) SSD instance store volumes.

Instance store volumes persist only for the life of the instance. When you stop or terminate an instance, the applications and data in its instance store volumes are erased. We recommend that you regularly back up or replicate important data in your instance store volumes. For more information, see [Amazon EC2 instance store \(p. 1188\)](#) and [SSD instance store volumes \(p. 1199\)](#).

The following is a summary of the hardware specifications for storage optimized instances.

Instance type	Default vCPUs	Memory (GiB)
d2.xlarge	4	30.5
d2.2xlarge	8	61
d2.4xlarge	16	122
d2.8xlarge	36	244
h1.2xlarge	8	32
h1.4xlarge	16	64
h1.8xlarge	32	128
h1.16xlarge	64	256
i3.large	2	15.25
i3.xlarge	4	30.5
i3.2xlarge	8	61
i3.4xlarge	16	122
i3.8xlarge	32	244
i3.16xlarge	64	488
i3.metal	72	512
i3en.large	2	16
i3en.xlarge	4	32
i3en.2xlarge	8	64
i3en.3xlarge	12	96
i3en.6xlarge	24	192
i3en.12xlarge	48	384
i3en.24xlarge	96	768
i3en.metal	96	768

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instance Types](#).

For more information about specifying CPU options, see [Optimizing CPU options \(p. 632\)](#).

Instance performance

To ensure the best disk throughput performance from your instance on Linux, we recommend that you use the most recent version of Amazon Linux 2 or the Amazon Linux AMI.

For instances with NVMe instance store volumes, you must use a Linux AMI with kernel version 4.4 or later. Otherwise, your instance will not achieve the maximum IOPS performance available.

D2 instances provide the best disk performance when you use a Linux kernel that supports persistent grants, an extension to the Xen block ring protocol that significantly improves disk throughput and scalability. For more information about persistent grants, see [this article](#) in the Xen Project Blog.

EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your instance. Some storage optimized instances are EBS-optimized by default at no additional cost. For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#).

Some storage optimized instance types provide the ability to control processor C-states and P-states on Linux. C-states control the sleep levels that a core can enter when it is inactive, while P-states control the desired performance (in CPU frequency) from a core. For more information, see [Processor state control for your EC2 instance \(p. 622\)](#).

Network performance

You can enable enhanced networking on supported instance types to provide lower latencies, lower network jitter, and higher packet-per-second (PPS) performance. Most applications do not consistently need a high level of network performance, but can benefit from access to increased bandwidth when they send or receive data. For more information, see [Enhanced networking on Linux \(p. 813\)](#).

The following is a summary of network performance for storage optimized instances that support enhanced networking.

Instance type	Network performance	Enhanced networking
i3.4xlarge and smaller	Up to 10 Gbps †	ENAs (p. 814)
i3.8xlarge h1.8xlarge	10 Gbps	ENAs (p. 814)
i3en.3xlarge and smaller	Up to 25 Gbps †	ENAs (p. 814)
i3.16xlarge i3.metal i3en.6xlarge h1.16xlarge	25 Gbps	ENAs (p. 814)
i3en.12xlarge	50 Gbps	ENAs (p. 814)
i3en.24xlarge i3en.metal	100 Gbps	ENAs (p. 814)
d2.xlarge	Moderate	Intel 82599 VF (p. 826)
d2.2xlarge d2.4xlarge	High	Intel 82599 VF (p. 826)
d2.8xlarge	10 Gbps	Intel 82599 VF (p. 826)

† These instances use a network I/O credit mechanism to allocate network bandwidth to instances based on average bandwidth utilization. They accrue credits when their bandwidth is below their baseline bandwidth, and can use these credits when they perform network data transfers. For more information, open a support case and ask about baseline bandwidth for the specific instance types that you are interested in.

SSD I/O performance

If you use a Linux AMI with kernel version 4.4 or later and use all the SSD-based instance store volumes available to your instance, you get the IOPS (4,096 byte block size) performance listed in the following table (at queue depth saturation). Otherwise, you get lower IOPS performance.

Instance Size	100% Random Read IOPS	Write IOPS
i3.large *	100,125	35,000
i3.xlarge *	206,250	70,000
i3.2xlarge	412,500	180,000
i3.4xlarge	825,000	360,000
i3.8xlarge	1.65 million	720,000
i3.16xlarge	3.3 million	1.4 million
i3.metal	3.3 million	1.4 million
i3en.large *	42,500	32,500
i3en.xlarge *	85,000	65,000
i3en.2xlarge *	170,000	130,000
i3en.3xlarge	250,000	200,000
i3en.6xlarge	500,000	400,000
i3en.12xlarge	1 million	800,000
i3en.24xlarge	2 million	1.6 million
i3en.metal	2 million	1.6 million

* For these instances, you can get up to the specified performance.

As you fill your SSD-based instance store volumes, the I/O performance that you get decreases. This is due to the extra work that the SSD controller must do to find available space, rewrite existing data, and erase unused space so that it can be rewritten. This process of garbage collection results in internal write amplification to the SSD, expressed as the ratio of SSD write operations to user write operations. This decrease in performance is even larger if the write operations are not in multiples of 4,096 bytes or not aligned to a 4,096-byte boundary. If you write a smaller amount of bytes or bytes that are not aligned, the SSD controller must read the surrounding data and store the result in a new location. This pattern results in significantly increased write amplification, increased latency, and dramatically reduced I/O performance.

SSD controllers can use several strategies to reduce the impact of write amplification. One such strategy is to reserve space in the SSD instance storage so that the controller can more efficiently manage the space available for write operations. This is called *over-provisioning*. The SSD-based instance store

volumes provided to an instance don't have any space reserved for over-provisioning. To reduce write amplification, we recommend that you leave 10% of the volume unpartitioned so that the SSD controller can use it for over-provisioning. This decreases the storage that you can use, but increases performance even if the disk is close to full capacity.

For instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see [Instance store volume TRIM support \(p. 1201\)](#).

Instance features

The following is a summary of features for storage optimized instances:

	EBS only	Instance store	Placement group
D2	No	HDD	Yes
H1	No	HDD *	Yes
I3	No	NVMe *	Yes
I3en	No	NVMe *	Yes

* The root device volume must be an Amazon EBS volume.

For more information, see the following:

- [Amazon EBS and NVMe on Linux instances \(p. 1136\)](#)
- [Amazon EC2 instance store \(p. 1188\)](#)
- [Placement groups \(p. 869\)](#)

Support for vCPUs

The d2.8xlarge instance type provides 36 vCPUs, which might cause launch issues in some Linux operating systems that have a vCPU limit of 32. We strongly recommend that you use the latest AMIs when you launch d2.8xlarge instances.

The following Linux AMIs support launching d2.8xlarge instances with 36 vCPUs:

- [Amazon Linux 2 \(HVM\)](#)
- [Amazon Linux AMI 2018.03 \(HVM\)](#)
- [Ubuntu Server 14.04 LTS \(HVM\) or later](#)
- [Red Hat Enterprise Linux 7.1 \(HVM\)](#)
- [SUSE Linux Enterprise Server 12 \(HVM\)](#)

If you must use a different AMI for your application, and your d2.8xlarge instance launch does not complete successfully (for example, if your instance status changes to stopped during launch with a `Client.InstanceInitiatedShutdown` state transition reason), modify your instance as described in the following procedure to support more than 32 vCPUs so that you can use the d2.8xlarge instance type.

To update an instance to support more than 32 vCPUs

1. Launch a D2 instance using your AMI, choosing any D2 instance type other than d2.8xlarge.

2. Update the kernel to the latest version by following your operating system-specific instructions. For example, for RHEL 6, use the following command:

```
sudo yum update -y kernel
```

3. Stop the instance.
4. (Optional) Create an AMI from the instance that you can use to launch any additional d2.8xlarge instances that you need in the future.
5. Change the instance type of your stopped instance to d2.8xlarge (choose **Actions, Instance Settings, Change Instance Type**, and then follow the directions).
6. Start the instance. If the instance launches properly, you are done. If the instance still does not boot properly, proceed to the next step.
7. (Optional) If the instance still does not boot properly, the kernel on your instance may not support more than 32 vCPUs. However, you may be able to boot the instance if you limit the vCPUs.
 - a. Change the instance type of your stopped instance to any D2 instance type other than d2.8xlarge (choose **Actions, Instance Settings, Change Instance Type**, and then follow the directions).
 - b. Add the `maxcpus=32` option to your boot kernel parameters by following your operating system-specific instructions. For example, for RHEL 6, edit the `/boot/grub/menu.lst` file and add the following option to the most recent and active kernel entry:

```
default=0
timeout=1
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.32-504.3.3.el6.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-2.6.32-504.3.3.el6.x86_64 maxcpus=32 console=ttyS0 ro
    root=UUID=9996863e-b964-47d3-a33b-3920974fdbd9 rd_NO_LUKS KEYBOARDTYPE=pc
    KEYTABLE=us LANG=en_US.UTF-8 xen_blkfront.sda_is_xvda=1 console=ttyS0,115200n8
    console=tty0 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=auto rd_NO_LVM
    rd_NO_DM
initrd /boot/initramfs-2.6.32-504.3.3.el6.x86_64.img
```

- c. Stop the instance.
- d. (Optional) Create an AMI from the instance that you can use to launch any additional d2.8xlarge instances that you need in the future.
- e. Change the instance type of your stopped instance to d2.8xlarge (choose **Actions, Instance Settings, Change Instance Type**, and then follow the directions).
- f. Start the instance.

Release notes

- You must launch storage optimized instances using an HVM AMI. For more information, see [Linux AMI virtualization types \(p. 102\)](#).
- Instances built on the [Nitro System \(p. 205\)](#) have the following requirements:
 - [NVMe drivers \(p. 1136\)](#) must be installed
 - [Elastic Network Adapter \(ENA\) drivers \(p. 814\)](#) must be installed

The following Linux AMIs meet these requirements:

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (with `linux-aws` kernel) or later

- Red Hat Enterprise Linux 7.4 or later
- SUSE Linux Enterprise Server 12 SP2 or later
- CentOS 7.4.1708 or later
- FreeBSD 11.1 or later
- Debian GNU/Linux 9 or later
- Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
- To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support. Amazon Linux 2 and the latest versions of the Amazon Linux AMI support PCIe native hotplug, but earlier versions do not. You must enable the following Linux kernel configuration options:

```
CONFIG_HOTPLUG_PCI_PCIE=y  
CONFIG_PCIEASPM=y
```

- Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
- With FreeBSD AMIs, bare metal instances take nearly an hour to boot and I/O to the local NVMe storage does not complete. As a workaround, add the following line to `/boot/loader.conf` and reboot:

```
hw.nvme.per_cpu_io_queues="0"
```

- The d2.8xlarge instance type has 36 vCPUs, which might cause launch issues in some Linux operating systems that have a vCPU limit of 32. For more information, see [Support for vCPUs \(p. 276\)](#).
- There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see [How many instances can I run in Amazon EC2?](#) in the Amazon EC2 FAQ.

Linux accelerated computing instances

Accelerated computing instances use hardware accelerators, or co-processors, to perform some functions, such as floating point number calculations, graphics processing, or data pattern matching, more efficiently than is possible in software running on CPUs. These instances enable more parallelism for higher throughput on compute-intensive workloads.

If you require high processing capability, you'll benefit from using accelerated computing instances, which provide access to hardware-based compute accelerators such as Graphics Processing Units (GPUs), Field Programmable Gate Arrays (FPGAs), or AWS Inferentia.

Contents

- [GPU instances \(p. 279\)](#)
- [Instances with AWS Inferentia \(p. 280\)](#)
- [FPGA instances \(p. 280\)](#)
- [Hardware specifications \(p. 281\)](#)
- [Instance performance \(p. 282\)](#)
- [Network performance \(p. 282\)](#)
- [Instance features \(p. 283\)](#)

- [Release notes \(p. 284\)](#)
- [Installing NVIDIA drivers on Linux instances \(p. 284\)](#)
- [Activate NVIDIA GRID Virtual Applications \(p. 292\)](#)
- [Optimizing GPU settings \(p. 292\)](#)

GPU instances

GPU-based instances provide access to NVIDIA GPUs with thousands of compute cores. You can use these instances to accelerate scientific, engineering, and rendering applications by leveraging the CUDA or Open Computing Language (OpenCL) parallel computing frameworks. You can also use them for graphics applications, including game streaming, 3-D application streaming, and other graphics workloads.

G4 instances

G4 instances use NVIDIA Tesla GPUs and provide a cost-effective, high-performance platform for general purpose GPU computing using the CUDA or machine learning frameworks along with graphics applications using DirectX or OpenGL. G4 instances provide high-bandwidth networking, powerful half and single-precision floating-point capabilities, along with INT8 and INT4 precisions. Each GPU has 16 GiB of GDDR6 memory, making G4 instances well-suited for machine learning inference, video transcoding, and graphics applications like remote graphics workstations and game streaming in the cloud.

For more information, see [Amazon EC2 G4 Instances](#).

G4 instances support NVIDIA GRID Virtual Workstation. For more information, see [NVIDIA Marketplace offerings](#).

G3 instances

G3 instances use NVIDIA Tesla M60 GPUs and provide a cost-effective, high-performance platform for graphics applications using DirectX or OpenGL. G3 instances also provide NVIDIA GRID Virtual Workstation features, such as support for four monitors with resolutions up to 4096x2160, and NVIDIA GRID Virtual Applications. G3 instances are well-suited for applications such as 3D visualizations, graphics-intensive remote workstations, 3D rendering, video encoding, virtual reality, and other server-side graphics workloads requiring massively parallel processing power.

For more information, see [Amazon EC2 G3 Instances](#).

G3 instances support NVIDIA GRID Virtual Workstation and NVIDIA GRID Virtual Applications. To activate either of these features, see [Activate NVIDIA GRID Virtual Applications \(p. 292\)](#).

G2 instances

G2 instances use NVIDIA GRID K520 GPUs and provide a cost-effective, high-performance platform for graphics applications using DirectX or OpenGL. NVIDIA GRID GPUs also support NVIDIA's fast capture and encode API operations. Example applications include video creation services, 3D visualizations, streaming graphics-intensive applications, and other server-side graphics workloads.

P3 instances

P3 instances use NVIDIA Tesla V100 GPUs and are designed for general purpose GPU computing using the CUDA or OpenCL programming models or through a machine learning framework. P3 instances provide high-bandwidth networking, powerful half, single, and double-precision floating-point capabilities, and up to 32 GiB of memory per GPU, which makes them ideal for deep learning, computational fluid dynamics, computational finance, seismic analysis, molecular modeling, genomics,

rendering, and other server-side GPU compute workloads. Tesla V100 GPUs do not support graphics mode.

For more information, see [Amazon EC2 P3 Instances](#).

P3 instances support NVIDIA NVLink peer to peer transfers. For more information, see [NVIDIA NVLink](#).

P2 instances

P2 instances use NVIDIA Tesla K80 GPUs and are designed for general purpose GPU computing using the CUDA or OpenCL programming models. P2 instances provide high-bandwidth networking, powerful single and double precision floating-point capabilities, and 12 GiB of memory per GPU, which makes them ideal for deep learning, graph databases, high-performance databases, computational fluid dynamics, computational finance, seismic analysis, molecular modeling, genomics, rendering, and other server-side GPU compute workloads.

P2 instances support NVIDIA GPUDirect peer to peer transfers. For more information, see [NVIDIA GPUDirect](#).

Instances with AWS Inferentia

These instances are designed to accelerate machine learning using [AWS Inferentia](#), a custom AI/ML chip from Amazon that provides high performance and low latency machine learning inference. These instances are optimized for deploying Deep Learning (DL) models for applications, such as natural language processing, object detection and classification, content personalization and filtering, and speech recognition.

There are a variety of ways that you can get started:

- Use SageMaker, a fully-managed service that is the easiest way to get started with machine learning models. For more information, see [Compile and deploy a TensorFlow model on Inf1 instances](#) on github.
- Launch an Inf1 instance using the Deep Learning AMI. For more information, see [AWS Inferentia with DLAMI](#) in the [AWS Deep Learning AMI Developer Guide](#).
- Launch an Inf1 instance using your own AMI and install the [AWS Neuron SDK](#), which enables you to compile, run, and profile deep learning models for AWS Inferentia.
- Launch a container instance using an Inf1 instance and an Amazon ECS-optimized AMI. For more information, see [Amazon Linux 2 \(Inferentia\) AMIs](#) in the [Amazon Elastic Container Service Developer Guide](#).
- Create an Amazon EKS cluster with nodes running Inf1 instances. For more information, see [Inferentia support](#) in the [Amazon EKS User Guide](#).

For more information, see [Machine Learning on AWS](#).

Inf1 instances

Inf1 instances use AWS Inferentia machine learning inference chips. Inferentia was developed to enable highly cost-effective low latency inference performance at any scale.

For more information, see [Amazon EC2 Inf1 Instances](#).

FPGA instances

FPGA-based instances provide access to large FPGAs with millions of parallel system logic cells. You can use FPGA-based accelerated computing instances to accelerate workloads such as genomics, financial

analysis, real-time video processing, big data analysis, and security workloads by leveraging custom hardware accelerations. You can develop these accelerations using hardware description languages such as Verilog or VHDL, or by using higher-level languages such as OpenCL parallel computing frameworks. You can either develop your own hardware acceleration code or purchase hardware accelerations through the [AWS Marketplace](#).

The [FPGA Developer AMI](#) provides the tools for developing, testing, and building AFIs. You can use the FPGA Developer AMI on any EC2 instance with at least 32 GB of system memory (for example, C5, M4, and R4 instances).

For more information, see the documentation for the [AWS FPGA Hardware Development Kit](#).

F1 instances

F1 instances use Xilinx UltraScale+ VU9P FPGAs and are designed to accelerate computationally intensive algorithms, such as data-flow or highly parallel operations not suited to general purpose CPUs. Each FPGA in an F1 instance contains approximately 2.5 million logic elements and approximately 6,800 Digital Signal Processing (DSP) engines, along with 64 GiB of local DDR ECC protected memory, connected to the instance by a dedicated PCIe Gen3 x16 connection. F1 instances provide local NVMe SSD volumes.

Developers can use the FPGA Developer AMI and AWS Hardware Developer Kit to create custom hardware accelerations for use on F1 instances. The FPGA Developer AMI includes development tools for full-cycle FPGA development in the cloud. Using these tools, developers can create and share Amazon FPGA Images (AFIs) that can be loaded onto the FPGA of an F1 instance.

For more information, see [Amazon EC2 F1 Instances](#).

Hardware specifications

The following is a summary of the hardware specifications for accelerated computing instances.

Instance type	Default vCPUs	Memory (GiB)	Accelerators
p2.xlarge	4	61	1
p2.8xlarge	32	488	8
p2.16xlarge	64	732	16
p3.2xlarge	8	61	1
p3.8xlarge	32	244	4
p3.16xlarge	64	488	8
p3dn.24xlarge	96	768	8
g2.2xlarge	8	15	1
g2.8xlarge	32	60	4
g3s.xlarge	4	30.5	1
g3.4xlarge	16	122	1
g3.8xlarge	32	244	2
g3.16xlarge	64	488	4

Instance type	Default vCPUs	Memory (GiB)	Accelerators
g4dn.xlarge	4	16	1
g4dn.2xlarge	8	32	1
g4dn.4xlarge	16	64	1
g4dn.8xlarge	32	128	1
g4dn.12xlarge	48	192	4
g4dn.16xlarge	64	256	1
g4dn.metal	96	384	8
f1.2xlarge	8	122	1
f1.4xlarge	16	244	2
f1.16xlarge	64	976	8
inf1.xlarge	4	8	1
inf1.2xlarge	8	16	1
inf1.6xlarge	24	48	4
inf1.24xlarge	96	192	16

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instance Types](#).

For more information about specifying CPU options, see [Optimizing CPU options \(p. 632\)](#).

Instance performance

There are several GPU setting optimizations that you can perform to achieve the best performance on your instances. For more information, see [Optimizing GPU settings \(p. 292\)](#).

EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your instance. Some accelerated computing instances are EBS-optimized by default at no additional cost. For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#).

Some accelerated computing instance types provide the ability to control processor C-states and P-states on Linux. C-states control the sleep levels that a core can enter when it is inactive, while P-states control the desired performance (in CPU frequency) from a core. For more information, see [Processor state control for your EC2 instance \(p. 622\)](#).

Network performance

You can enable enhanced networking on supported instance types to provide lower latencies, lower network jitter, and higher packet-per-second (PPS) performance. Most applications do not consistently need a high level of network performance, but can benefit from access to increased bandwidth when they send or receive data. For more information, see [Enhanced networking on Linux \(p. 813\)](#).

The following is a summary of network performance for accelerated computing instances that support enhanced networking.

Instance type	Network performance	Enhanced networking
f1.2xlarge f1.4xlarge g3.4xlarge p3.2xlarge	Up to 10 Gbps †	ENAv (p. 814)
g3s.xlarge g3.8xlarge p2.8xlarge p3.8xlarge	10 Gbps	ENAv (p. 814)
g4dn.xlarge g4dn.2xlarge g4dn.4xlarge inf1.xlarge inf1.2xlarge	Up to 25 Gbps †	ENAv (p. 814)
f1.16xlarge g3.16xlarge inf1.6xlarge p2.16xlarge p3.16xlarge	25 Gbps	ENAv (p. 814)
g4dn.8xlarge g4dn.12xlarge g4dn.16xlarge	50 Gbps	ENAv (p. 814)
g4dn.metal inf1.24xlarge p3dn.24xlarge	100 Gbps	ENAv (p. 814)

† These instances use a network I/O credit mechanism to allocate network bandwidth to instances based on average bandwidth utilization. They accrue credits when their bandwidth is below their baseline bandwidth, and can use these credits when they perform network data transfers. For more information, open a support case and ask about baseline bandwidth for the specific instance types that you are interested in.

Instance features

The following is a summary of features for accelerated computing instances.

	EBS only	NVMe EBS	Instance store	Placement group
G2	No	No	SSD	Yes
G3	Yes	No	No	Yes
G4	No	Yes	NVMe *	Yes
Inf1	Yes	No	No	Yes
P2	Yes	No	No	Yes
P3	24xlarge: No All other sizes: Yes	24xlarge: Yes All other sizes: No	24xlarge: NVMe *	Yes
F1	No	No	NVMe *	Yes

* The root device volume must be an Amazon EBS volume.

For more information, see the following:

- [Amazon EBS and NVMe on Linux instances \(p. 1136\)](#)
- [Amazon EC2 instance store \(p. 1188\)](#)

- [Placement groups \(p. 869\)](#)

Release notes

- You must launch the instance using an HVM AMI.
- Instances built on the [Nitro System \(p. 205\)](#) have the following requirements:
 - [NVMe drivers \(p. 1136\)](#) must be installed
 - [Elastic Network Adapter \(ENA\) drivers \(p. 814\)](#) must be installed

The following Linux AMIs meet these requirements:

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (with `linux-aws` kernel) or later
- Red Hat Enterprise Linux 7.4 or later
- SUSE Linux Enterprise Server 12 SP2 or later
- CentOS 7.4.1708 or later
- FreeBSD 11.1 or later
- Debian GNU/Linux 9 or later
- GPU-based instances can't access the GPU unless the NVIDIA drivers are installed. For more information, see [Installing NVIDIA drivers on Linux instances \(p. 284\)](#).
- Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
- To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support. Amazon Linux 2 and the latest versions of the Amazon Linux AMI support PCIe native hotplug, but earlier versions do not. You must enable the following Linux kernel configuration options:

```
CONFIG_HOTPLUG_PCI_PCIE=y  
CONFIG_PCIEASPM=y
```

- Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
- There is a limit of 100 APIs per Region.
- There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see [How many instances can I run in Amazon EC2?](#) in the Amazon EC2 FAQ.

Installing NVIDIA drivers on Linux instances

An instance with an attached GPU, such as a P3 or G4 instance, must have the appropriate NVIDIA driver installed. Depending on the instance type, you can either download a public NVIDIA driver, download a driver from Amazon S3 that is available only to AWS customers, or use an AMI with the driver pre-installed.

Contents

- [Types of NVIDIA drivers \(p. 285\)](#)
- [Available drivers by instance type \(p. 285\)](#)

- [Installation options \(p. 286\)](#)
 - [Option 1: AMIs with the NVIDIA drivers installed \(p. 286\)](#)
 - [Option 2: Public NVIDIA drivers \(p. 286\)](#)
 - [Option 3: GRID drivers \(G3 and G4 instances\) \(p. 287\)](#)
 - [Option 4: NVIDIA gaming drivers \(G4 instances\) \(p. 289\)](#)
- [Installing an additional version of CUDA \(p. 291\)](#)

Types of NVIDIA drivers

The following are the main types of NVIDIA drivers that can be used with GPU-based instances.

Tesla drivers

These drivers are intended primarily for compute workloads, which use GPUs for computational tasks such as parallelized floating-point calculations for machine learning and fast Fourier transforms for high performance computing applications.

GRID drivers

These drivers are certified to provide optimal performance for professional visualization applications that render content such as 3D models or high-resolution videos. You can configure GRID drivers to support two modes. Quadro Virtual Workstations provide access to four 4K displays per GPU. GRID vApps provide RDSH App hosting capabilities.

Gaming drivers

These drivers contain optimizations for gaming and are updated frequently to provide performance enhancements. They support a single 4K display per GPU.

NVIDIA control panel

The NVIDIA control panel is supported with GRID and Gaming drivers. It is not supported with Tesla drivers.

Supported APIs for Tesla, GRID, and gaming drivers

- OpenCL, OpenGL, and Vulkan
- NVIDIA CUDA and related libraries (for example, cuDNN, TensorRT, nvJPEG, and cuBLAS)
- NVENC for video encoding and NVDEC for video decoding

Available drivers by instance type

The following table summarizes the supported NVIDIA drivers for each GPU instance type.

Instance type	Tesla driver	GRID driver	Gaming driver
G2	No	Yes	No
G3	Yes	Yes	No
G4	Yes	Yes	Yes
P2	Yes	No	No
P3	Yes	Yes †	No

† Using Marketplace AMIs only

Installation options

Use one of the following options to get the NVIDIA drivers required for your GPU instance.

Options

- [Option 1: AMIs with the NVIDIA drivers installed \(p. 286\)](#)
- [Option 2: Public NVIDIA drivers \(p. 286\)](#)
- [Option 3: GRID drivers \(G3 and G4 instances\) \(p. 287\)](#)
- [Option 4: NVIDIA gaming drivers \(G4 instances\) \(p. 289\)](#)

Option 1: AMIs with the NVIDIA drivers installed

AWS and NVIDIA offer different Amazon Machine Images (AMI) that come with the NVIDIA drivers installed.

- [Marketplace offerings with the Tesla driver](#)
- [Marketplace offerings with the GRID driver](#)
- [Marketplace offerings with the Gaming driver](#)

To update the driver version installed using one of these AMIs, you must uninstall the NVIDIA packages from your instance to avoid version conflicts. Use this command to uninstall the NVIDIA packages:

```
[ec2-user ~]$ sudo yum erase nvidia cuda
```

The CUDA toolkit package has dependencies on the NVIDIA drivers. Uninstalling the NVIDIA packages erases the CUDA toolkit. You must reinstall the CUDA toolkit after installing the NVIDIA driver.

Option 2: Public NVIDIA drivers

The options offered by AWS come with the necessary license for the driver. Alternatively, you can install the public drivers and bring your own license. To install a public driver, download it from the NVIDIA site as described here.

Alternatively, you can use the options offered by AWS instead of the public drivers. To use a GRID driver on a P3 instance, use the AWS Marketplace AMIs as described in [Option 1 \(p. 286\)](#). To use a GRID driver on a G3 or G4 instance, use the AWS Marketplace AMIs, as described in Option 1 or install the NVIDIA drivers provided by AWS as described in [Option 3 \(p. 287\)](#).

To download a public NVIDIA driver

Log on to your Linux instance and download the 64-bit NVIDIA driver appropriate for the instance type from <http://www.nvidia.com/Download/Find.aspx>. For **Product Type**, **Product Series**, and **Product**, use the options in the following table.

Instance	Product Type	Product Series	Product
G2	GRID	GRID Series	GRID K520
G3	Tesla	M-Class	M60
G4 †	Tesla	T-Series	T4

Instance	Product Type	Product Series	Product
P2	Tesla	K-Series	K80
P3	Tesla	V-Series	V100

† G4 instances require driver version 418.87 or later.

To install the NVIDIA driver on Linux

For more information about installing and configuring the driver, see the [NVIDIA Driver Installation Quickstart Guide](#).

Option 3: GRID drivers (G3 and G4 instances)

These downloads are available to AWS customers only. By downloading, you agree to use the downloaded software only to develop AMIs for use with the NVIDIA Tesla T4 or NVIDIA Tesla M60 hardware. Upon installation of the software, you are bound by the terms of the [NVIDIA GRID Cloud End User License Agreement](#).

Prerequisites

- Install the AWS CLI on your Linux instance and configure default credentials. For more information, see [Installing the AWS CLI](#) in the *AWS Command Line Interface User Guide*.
- IAM users must have the permissions granted by the [AmazonS3ReadOnlyAccess](#) policy.

To install the NVIDIA GRID driver on your Linux instance

1. Connect to your Linux instance. Install **gcc** and **make**, if they are not already installed.
2. Update your package cache and get necessary package updates for your instance.
 - For Amazon Linux, CentOS, and Red Hat Enterprise Linux:

```
[ec2-user ~]$ sudo yum update -y
```

- For Ubuntu and Debian:

```
$ sudo apt-get update -y
```

3. (Ubuntu 16.04 and later, with the `linux-aws` package) Upgrade the `linux-aws` package to receive the latest version.

```
$ sudo apt-get upgrade -y linux-aws
```

4. Reboot your instance to load the latest kernel version.

```
[ec2-user ~]$ sudo reboot
```

5. Reconnect to your instance after it has rebooted.
6. Install the **gcc** compiler and the kernel headers package for the version of the kernel you are currently running.
 - For Amazon Linux, CentOS, and Red Hat Enterprise Linux:

```
[ec2-user ~]$ sudo yum install -y gcc kernel-devel-$(uname -r)
```

- For Ubuntu and Debian:

```
$ sudo apt-get install -y gcc make linux-headers-$(uname -r)
```

7. [CentOS, Red Hat Enterprise Linux, Ubuntu, Debian] Disable the nouveau open source driver for NVIDIA graphics cards.

- a. Add nouveau to the /etc/modprobe.d/blacklist.conf blacklist file. Copy the following code block and paste it into a terminal.

```
[ec2-user ~]$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

- b. Edit the /etc/default/grub file and add the following line:

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

- c. Rebuild the Grub configuration.

- For CentOS and Red Hat Enterprise Linux:

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

- For Ubuntu and Debian:

```
$ sudo update-grub
```

8. Download the GRID driver installation utility using the following command:

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linu-x-nvidia-drivers/latest/ .
```

Multiple versions of the GRID driver are stored in this bucket. You can see all of the available versions using the following command.

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linu-x-nvidia-drivers/
```

Starting with GRID version 11.0, you can use the driver packages under latest for both G3 and G4 instances. We will not add versions later than 11.0 to g4/latest, but will keep version 11.0 and the earlier versions specific to G4 under g4/latest.

9. Add permissions to run the driver installation utility using the following command.

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

10. Run the self-install script as follows to install the GRID driver that you downloaded. For example:

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

When prompted, accept the license agreement and specify the installation options as required (you can accept the default options).

11. Reboot the instance.

```
[ec2-user ~]$ sudo reboot
```

12. Confirm that the driver is functional. The response for the following command lists the installed version of the NVIDIA driver and details about the GPUs.

```
[ec2-user ~]$ nvidia-smi -q | head
```

13. (Optional) Depending on your use case, you might complete the following optional steps. If you do not require this functionality, do not complete these steps.
 - a. To help take advantage of the four displays of up to 4K resolution, set up the high-performance display protocol [NICE DCV](#).
 - b. NVIDIA Quadro Virtual Workstation mode is enabled by default. To activate GRID Virtual Applications for RDSH Application hosting capabilities, complete the GRID Virtual Application activation steps in [Activate NVIDIA GRID Virtual Applications \(p. 292\)](#).

Option 4: NVIDIA gaming drivers (G4 instances)

These drivers are available to AWS customers only. By downloading them, you agree to use the downloaded software only to develop AMIs for use with the NVIDIA Tesla T4 hardware. Upon installation of the software, you are bound by the terms of the [NVIDIA GRID Cloud End User License Agreement](#).

Prerequisites

- Install the AWS CLI on your Linux instance and configure default credentials. For more information, see [Installing the AWS CLI](#) in the *AWS Command Line Interface User Guide*.
- IAM users must have the permissions granted by the [AmazonS3ReadOnlyAccess](#) policy.

To install the NVIDIA gaming driver on your Linux instance

1. Connect to your Linux instance. Install **gcc** and **make**, if they are not already installed.
2. Update your package cache and get necessary package updates for your instance.
 - For Amazon Linux, CentOS, and Red Hat Enterprise Linux:

```
[ec2-user ~]$ sudo yum update -y
```

- For Ubuntu and Debian:

```
$ sudo apt-get update -y
```

3. (Ubuntu 16.04 and later, with the `linux-aws` package) Upgrade the `linux-aws` package to receive the latest version.

```
$ sudo apt-get upgrade -y linux-aws
```

4. Reboot your instance to load the latest kernel version.

```
[ec2-user ~]$ sudo reboot
```

5. Reconnect to your instance after it has rebooted.
6. Install the **gcc** compiler and the kernel headers package for the version of the kernel you are currently running.

- For Amazon Linux, CentOS, and Red Hat Enterprise Linux:

```
[ec2-user ~]$ sudo yum install -y gcc kernel-devel-$(uname -r)
```

- For Ubuntu and Debian:

```
$ sudo apt-get install -y gcc make linux-headers-$(uname -r)
```

7. [CentOS, Red Hat Enterprise Linux, Ubuntu, Debian] Disable the nouveau open source driver for NVIDIA graphics cards.

- a. Add nouveau to the /etc/modprobe.d/blacklist.conf blacklist file. Copy the following code block and paste it into a terminal.

```
[ec2-user ~]$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

- b. Edit the /etc/default/grub file and add the following line:

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

- c. Rebuild the Grub configuration.

- For CentOS and Red Hat Enterprise Linux:

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

- For Ubuntu and Debian:

```
$ sudo update-grub
```

8. Download the gaming driver installation utility using the following command:

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
```

Multiple versions of the gaming driver are stored in this bucket. You can see all of the available versions using the following command:

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

9. Add permissions to run the driver installation utility using the following command.

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

10. Run the installer using the following command:

```
[ec2-user ~]$ sudo ./NVIDIA-Linux-x86_64*.run
```

When prompted, accept the license agreement and specify the installation options as required (you can accept the default options).

11. Use the following command to create the required configuration file.

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

12. Use the following command to download and rename the certification file.

- For version 440.68 or later:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2020_04.cert"
```

- For earlier versions:

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://nvidia-
gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCert-Linux_2019_09.cert"
```

13. Reboot the instance.

```
[ec2-user ~]$ sudo reboot
```

14. (Optional) To help take advantage of a single display of up to 4K resolution, set up the high-performance display protocol [NICE DCV](#). If you do not require this functionality, do not complete this step.

Installing an additional version of CUDA

After you install an NVIDIA graphics driver on your instance, you can install a version of CUDA other than the version that is bundled with the graphics driver. The following procedure demonstrates how to configure multiple versions of CUDA on the instance.

To install the CUDA toolkit

1. Connect to your Linux instance.
2. Open the [NVIDIA website](#) and select the version of CUDA that you need.
3. Select the architecture, distribution, and version for the operating system on your instance. For **Installer Type**, select **runfile (local)**.
4. Follow the instructions to download the install script.
5. Add run permissions to the install script that you downloaded using the following command.

```
[ec2-user ~]$ chmod +x downloaded_installer_file
```

6. Run the install script as follows to install the CUDA toolkit and add the CUDA version number to the toolkit path.

```
[ec2-user ~]$ sudo downloaded_installer_file --silent --override --toolkit --samples --
toolkitpath=/usr/local/cuda-<version> --samplespath=/usr/local/cuda --no-opengl-lib
```

7. (Optional) Set the default CUDA version as follows.

```
[ec2-user ~]$ ln -s /usr/local/cuda-<version> /usr/local/cuda
```

Activate NVIDIA GRID Virtual Applications

To activate the GRID Virtual Applications on G3 and G4 instances (NVIDIA GRID Virtual Workstation is enabled by default), you must define the product type for the driver in the `/etc/nvidia/gridd.conf` file.

To activate GRID Virtual Applications on Linux instances

1. Create the `/etc/nvidia/gridd.conf` file from the provided template file.

```
[ec2-user ~]$ sudo cp /etc/nvidia/gridd.conf.template /etc/nvidia/gridd.conf
```

2. Open the `/etc/nvidia/gridd.conf` file in your favorite text editor.
3. Find the `FeatureType` line, and set it equal to 0. Then add a line with `IgnoreSP=TRUE`.

```
FeatureType=0
IgnoreSP=TRUE
```

4. Save the file and exit.
5. Reboot the instance to pick up the new configuration.

```
[ec2-user ~]$ sudo reboot
```

Optimizing GPU settings

There are several GPU setting optimizations that you can perform to achieve the best performance on G3, G4, P2, P3, and P3dn instances. By default, the NVIDIA driver uses an autoboot feature, which varies the GPU clock speeds. By disabling the autoboot feature and setting the GPU clock speeds to their maximum frequency, you can consistently achieve the maximum performance with your GPU instances. The following procedure helps you to configure the GPU settings to be persistent, disable the autoboot feature, and set the GPU clock speeds to their maximum frequency.

To optimize GPU settings

1. Configure the GPU settings to be persistent. This command can take several minutes to run.

```
[ec2-user ~]$ sudo nvidia-persistenced
```

2. Disable the autoboot feature for all GPUs on the instance.

```
[ec2-user ~]$ sudo nvidia-smi --auto-boost-default=0
```

Note

GPUs on P3, P3dn, and G4 instances do not support autoboot.

3. Set all GPU clock speeds to their maximum frequency. Use the memory and graphics clock speeds specified in the following commands.

Note

Some versions of the NVIDIA driver do not allow setting application clock speed and throw a "Setting applications clocks is not supported for GPU ..." error, which you can ignore.

- G3 instances:

```
[ec2-user ~]$ sudo nvidia-smi -ac 2505,1177
```

- G4 instances:

```
[ec2-user ~]$ sudo nvidia-smi -ac 5001,1590
```

- P2 instances:

```
[ec2-user ~]$ sudo nvidia-smi -ac 2505,875
```

- P3 and P3dn instances:

```
[ec2-user ~]$ sudo nvidia-smi -ac 877,1530
```

Finding an Amazon EC2 instance type

Before you can launch an instance, you must select an instance type to use. The instance type that you choose might depend on your requirements for the instances that you'll launch. For example, you might choose an instance type based on the following requirements:

- Availability Zone or Region
- Compute
- Memory
- Networking
- Pricing
- Storage

Finding an instance type using the console

You can find an instance type that meets your needs using the Amazon EC2 console.

To find an instance type using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region in which to launch your instances. You can select any Region that's available to you, regardless of your location.
3. In the navigation pane, choose **Instance Types**.
4. (Optional) Choose the preferences (gear) icon to select which instance type attributes to display, such as **On-Demand Linux pricing**, and then choose **Confirm**. Alternatively, select an instance type and view all attributes using the **Details** pane.
5. Use the instance type attributes to filter the list of displayed instance types to only the instance types that meet your needs. For example, you can list all instance types that have more than eight vCPUs and also support hibernation.
6. (Optional) Select multiple instance types to see a side-by-side comparison across all attributes in the **Details** pane.
7. (Optional) To save the list of instance types to a comma-separated values (.csv) file for further review, choose **Download list CSV**. The file includes all instance types that match the filters you set.
8. After locating instance types that meet your needs, you can use them to launch instances. For more information, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#).

Finding an instance type using the AWS CLI

You can use AWS CLI commands for Amazon EC2 to find an instance type that meet your needs.

To find an instance type using the AWS CLI

1. If you have not done so already, install the AWS CLI. For more information, see the [AWS Command Line Interface User Guide](#).
2. Use the [describe-instance-types](#) command to filter instance types based on instance attributes. For example, you can use the following command to display only instance types with 48 vCPUs.

```
aws ec2 describe-instance-types --filters "Name=vcpu-info.default-vcpus,Values=48"
```

3. Use the [describe-instance-type-offerings](#) command to filter instance types offered by location (Region or Availability Zone). For example, you can use the following command to display the instance types offered in the specified Availability Zone.

```
aws ec2 describe-instance-type-offerings --location-type "availability-zone" --filters Name=location,Values=us-east-2a --region us-east-2
```

4. After locating instance types that meet your needs, make note of them so that you can use these instance types when you launch instances. For more information, see [Launching an Instance Using the AWS CLI](#) in the *AWS Command Line Interface User Guide*.

Changing the instance type

As your needs change, you might find that your instance is over-utilized (the instance type is too small) or under-utilized (the instance type is too large). If this is the case, you can change the size of your instance. For example, if your `t2.micro` instance is too small for its workload, you can change it to another instance type that is appropriate for the workload.

You might also want to migrate from a previous generation instance type to a current generation instance type to take advantage of some features; for example, support for IPv6.

If the root device for your instance is an EBS volume, you can change the size of the instance simply by changing its instance type, which is known as *resizing* it. If the root device for your instance is an instance store volume, you must migrate your application to a new instance with the instance type that you need. For more information about root device volumes, see [Storage for the root device \(p. 100\)](#).

When you resize an instance, you must select an instance type that is compatible with the configuration of the instance. If the instance type that you want is not compatible with the instance configuration you have, then you must migrate your application to a new instance with the instance type that you need.

Important

When you resize an instance, the resized instance usually has the same number of instance store volumes that you specified when you launched the original instance. With instance types that support NVMe instance store volumes (which are available by default), the resized instance might have additional instance store volumes, depending on the AMI. Otherwise, you can migrate your application to an instance with a new instance type manually, specifying the number of instance store volumes that you need when you launch the new instance.

Contents

- [Compatibility for resizing instances \(p. 295\)](#)
- [Resizing an Amazon EBS-backed instance \(p. 295\)](#)
- [Migrating an instance store-backed instance \(p. 297\)](#)
- [Migrating to a new instance configuration \(p. 298\)](#)

Compatibility for resizing instances

You can resize an instance only if its current instance type and the new instance type that you want are compatible in the following ways:

- **Virtualization type:** Linux AMIs use one of two types of virtualization: paravirtual (PV) or hardware virtual machine (HVM). You can't resize an instance that was launched from a PV AMI to an instance type that is HVM only. For more information, see [Linux AMI virtualization types \(p. 102\)](#). To check the virtualization type of your instance, see the **Virtualization** field on the details pane of the **Instances** screen in the Amazon EC2 console.
- **Architecture:** AMIs are specific to the architecture of the processor, so you must select an instance type with the same processor architecture as the current instance type. For example:
 - If you are resizing an instance type with a processor based on the Arm architecture, you are limited to the instance types that support a processor based on the Arm architecture, such as A1 and M6g.
 - The following instance types are the only instance types that support 32-bit AMIs: t2.nano, t2.micro, t2.small, t2.medium, c3.large, t1.micro, m1.small, m1.medium, and c1.medium. If you are resizing a 32-bit instance, you are limited to these instance types.
- **Network:** Newer instance types must be launched in a VPC. Therefore, you can't resize an instance in the EC2-Classic platform to a instance type that is available only in a VPC unless you have a nondefault VPC. To check whether your instance is in a VPC, check the **VPC ID** value on the details pane of the **Instances** screen in the Amazon EC2 console. For more information, see [Migrating from EC2-Classic to a VPC \(p. 904\)](#).
- **Enhanced networking:** Instance types that support [enhanced networking \(p. 813\)](#) require the necessary drivers installed. For example, instances based on the [Nitro System \(p. 205\)](#) require EBS-backed AMIs with the Elastic Network Adapter (ENA) drivers installed. To resize an instance from a type that does not support enhanced networking to a type that supports enhanced networking, you must install the [ENA drivers \(p. 814\)](#) or [ixgbevf drivers \(p. 826\)](#) on the instance, as appropriate.
- **NVMe:** EBS volumes are exposed as NVMe block devices on instances built on the [Nitro System \(p. 205\)](#). If you resize an instance from an instance type that does not support NVMe to an instance type that supports NVMe, you must first install the [NVMe drivers \(p. 1136\)](#) on your instance. Also, the device names for devices that you specify in the block device mapping are renamed using NVMe device names (`/dev/nvme[0-26]n1`). Therefore, to mount file systems at boot time using `/etc/fstab`, you must use UUID/Label instead of device names.
- **AMI:** For information about the AMIs required by instance types that support enhanced networking and NVMe, see the Release Notes in the following documentation:
 - [General purpose instances \(p. 208\)](#)
 - [Compute optimized instances \(p. 253\)](#)
 - [Memory optimized instances \(p. 260\)](#)
 - [Storage optimized instances \(p. 272\)](#)

Resizing an Amazon EBS-backed instance

You must stop your Amazon EBS-backed instance before you can change its instance type. When you stop and start an instance, be aware of the following:

- We move the instance to new hardware; however, the instance ID does not change.
- If your instance has a public IPv4 address, we release the address and give it a new public IPv4 address. The instance retains its private IPv4 addresses, any Elastic IP addresses, and any IPv6 addresses.
- If your instance is in an Auto Scaling group, the Amazon EC2 Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. To prevent this, you can suspend the scaling processes for the group while you're resizing your instance. For more information, see [Suspending and Resuming Scaling Processes](#) in the *Amazon EC2 Auto Scaling User Guide*.

- If your instance is in a [cluster placement group \(p. 870\)](#) and, after changing the instance type, the instance start fails, try the following: stop all the instances in the cluster placement group, change the instance type for the affected instance, and then restart all the instances in the cluster placement group.
- Ensure that you plan for downtime while your instance is stopped. Stopping and resizing an instance may take a few minutes, and restarting your instance may take a variable amount of time depending on your application's startup scripts.

For more information, see [Stop and start your instance \(p. 587\)](#).

Use the following procedure to resize an Amazon EBS-backed instance using the AWS Management Console.

New console

To resize an Amazon EBS-backed instance

1. (Optional) If the new instance type requires drivers that are not installed on the existing instance, you must connect to your instance and install the drivers first. For more information, see [Compatibility for resizing instances \(p. 295\)](#).
2. Open the Amazon EC2 console.
3. In the navigation pane, choose **Instances**.
4. Select the instance and choose **Actions, Instance state, Stop instance**.
5. In the confirmation dialog box, choose **Stop**. It can take a few minutes for the instance to stop.
6. With the instance still selected, choose **Actions, Instance settings, Change instance type**. This action is disabled if the instance state is not stopped.
7. In the **Change instance type** dialog box, do the following:
 - a. From **Instance type**, select the instance type that you want. If the instance type that you want does not appear in the list, then it is not compatible with the configuration of your instance (for example, because of virtualization type). For more information, see [Compatibility for resizing instances \(p. 295\)](#).
 - b. (Optional) If the instance type that you selected supports EBS-optimization, select **EBS-optimized** to enable EBS-optimization or deselect **EBS-optimized** to disable EBS-optimization. If the instance type that you selected is EBS-optimized by default, **EBS-optimized** is selected and you can't deselect it.
 - c. Choose **Apply** to accept the new settings.
8. To restart the stopped instance, select the instance and choose **Actions, Instance state, Start instance**. It can take a few minutes for the instance to enter the `running` state.
9. (Troubleshooting) If your instance won't boot, it is possible that one of the requirements for the new instance type was not met. For more information, see [Why is my Linux instance not booting after I changed its type?](#)

Old console

To resize an Amazon EBS-backed instance

1. (Optional) If the new instance type requires drivers that are not installed on the existing instance, you must connect to your instance and install the drivers first. For more information, see [Compatibility for resizing instances \(p. 295\)](#).
2. Open the Amazon EC2 console.
3. In the navigation pane, choose **Instances**.
4. Select the instance and choose **Actions, Instance State, Stop**.

5. In the confirmation dialog box, choose **Yes, Stop**. It can take a few minutes for the instance to stop.
6. With the instance still selected, choose **Actions, Instance Settings, Change Instance Type**. This action is disabled if the instance state is not stopped.
7. In the **Change Instance Type** dialog box, do the following:
 - a. From **Instance Type**, select the instance type that you want. If the instance type that you want does not appear in the list, then it is not compatible with the configuration of your instance (for example, because of virtualization type). For more information, see [Compatibility for resizing instances \(p. 295\)](#).
 - b. (Optional) If the instance type that you selected supports EBS-optimization, select **EBS-optimized** to enable EBS-optimization or deselect **EBS-optimized** to disable EBS-optimization. If the instance type that you selected is EBS-optimized by default, **EBS-optimized** is selected and you can't deselect it.
 - c. Choose **Apply** to accept the new settings.
8. To restart the stopped instance, select the instance and choose **Actions, Instance State, Start**.
9. In the confirmation dialog box, choose **Yes, Start**. It can take a few minutes for the instance to enter the `running` state.
10. (Troubleshooting) If your instance won't boot, it is possible that one of the requirements for the new instance type was not met. For more information, see [Why is my Linux instance not booting after I changed its type?](#)

Migrating an instance store-backed instance

When you want to move your application from one instance store-backed instance to an instance store-backed instance with a different instance type, you must migrate it by creating an image from your instance, and then launching a new instance from this image with the instance type that you need. To ensure that your users can continue to use the applications that you're hosting on your instance uninterrupted, you must take any Elastic IP address that you've associated with your original instance and associate it with the new instance. Then you can terminate the original instance.

New console

To migrate an instance store-backed instance

1. Back up any data on your instance store volumes that you need to keep to persistent storage. To migrate data on your EBS volumes that you need to keep, take a snapshot of the volumes (see [Creating Amazon EBS snapshots \(p. 1063\)](#)) or detach the volume from the instance so that you can attach it to the new instance later (see [Detaching an Amazon EBS volume from a Linux instance \(p. 1058\)](#)).
2. Create an AMI from your instance store-backed instance by satisfying the prerequisites and following the procedures in [Creating an instance store-backed Linux AMI \(p. 127\)](#). When you are finished creating an AMI from your instance, return to this procedure.
3. Open the Amazon EC2 console and in the navigation pane, choose **AMIs**. From the filter lists, choose **Owned by me**, and select the image that you created in the previous step. Notice that **AMI Name** is the name that you specified when you registered the image and **Source** is your Amazon S3 bucket.

Note

If you do not see the AMI that you created in the previous step, make sure that you have selected the Region in which you created your AMI.

4. Choose **Launch**. When you specify options for the instance, be sure to select the new instance type that you want. If the instance type that you want can't be selected, then it is not compatible with configuration of the AMI that you created (for example, because of

virtualization type). You can also specify any EBS volumes that you detached from the original instance.

It can take a few minutes for the instance to enter the `running` state.

5. (Optional) You can terminate the instance that you started with, if it's no longer needed. Select the instance and verify that you are about to terminate the original instance, not the new instance (for example, check the name or launch time). Choose **Actions**, **Instance state**, **Terminate instance**.

Old console

To migrate an instance store-backed instance

1. Back up any data on your instance store volumes that you need to keep to persistent storage. To migrate data on your EBS volumes that you need to keep, take a snapshot of the volumes (see [Creating Amazon EBS snapshots \(p. 1063\)](#)) or detach the volume from the instance so that you can attach it to the new instance later (see [Detaching an Amazon EBS volume from a Linux instance \(p. 1058\)](#)).
2. Create an AMI from your instance store-backed instance by satisfying the prerequisites and following the procedures in [Creating an instance store-backed Linux AMI \(p. 127\)](#). When you are finished creating an AMI from your instance, return to this procedure.
3. Open the Amazon EC2 console and in the navigation pane, choose **AMIs**. From the filter lists, choose **Owned by me**, and choose the image that you created in the previous step. Notice that **AMI Name** is the name that you specified when you registered the image and **Source** is your Amazon S3 bucket.

Note

If you do not see the AMI that you created in the previous step, make sure that you have selected the Region in which you created your AMI.

4. Choose **Launch**. When you specify options for the instance, be sure to select the new instance type that you want. If the instance type that you want can't be selected, then it is not compatible with configuration of the AMI that you created (for example, because of virtualization type). You can also specify any EBS volumes that you detached from the original instance.

It can take a few minutes for the instance to enter the `running` state.

5. (Optional) You can terminate the instance that you started with, if it's no longer needed. Select the instance and verify that you are about to terminate the original instance, not the new instance (for example, check the name or launch time). Choose **Actions**, **Instance State**, **Terminate**.

Migrating to a new instance configuration

If the current configuration of your instance is incompatible with the new instance type that you want, then you can't resize the instance to that instance type. Instead, you can migrate your application to a new instance with a configuration that is compatible with the new instance type that you want.

If you want to move from an instance launched from a PV AMI to an instance type that is HVM only, the general process is as follows:

New console

To migrate your application to a compatible instance

1. Back up any data on your instance store volumes that you need to keep to persistent storage. To migrate data on your EBS volumes that you need to keep, create a snapshot of the volumes

(see [Creating Amazon EBS snapshots \(p. 1063\)](#)) or detach the volume from the instance so that you can attach it to the new instance later (see [Detaching an Amazon EBS volume from a Linux instance \(p. 1058\)](#)).

2. Launch a new instance, selecting the following:
 - An HVM AMI.
 - The HVM only instance type.
 - If you are using an Elastic IP address, select the VPC that the original instance is currently running in.
 - Any EBS volumes that you detached from the original instance and want to attach to the new instance, or new EBS volumes based on the snapshots that you created.
 - If you want to allow the same traffic to reach the new instance, select the security group that is associated with the original instance.
3. Install your application and any required software on the instance.
4. Restore any data that you backed up from the instance store volumes of the original instance.
5. If you are using an Elastic IP address, assign it to the newly launched instance as follows:
 - a. In the navigation pane, choose **Elastic IPs**.
 - b. Select the Elastic IP address that is associated with the original instance and choose **Actions, Disassociate Elastic IP address**. When prompted for confirmation, choose **Disassociate**.
 - c. With the Elastic IP address still selected, choose **Actions, Associate Elastic IP address**.
 - d. For **Resource type**, choose **Instance**.
 - e. For **Instance**, choose the instance with which to associate the Elastic IP address. You can also enter text to search for a specific instance.
 - f. (Optional) For **Private IP address**, specify a private IP address with which to associate the Elastic IP address.
 - g. Choose **Associate**.
6. (Optional) You can terminate the original instance if it's no longer needed. Select the instance and verify that you are about to terminate the original instance, not the new instance (for example, check the name or launch time). Choose **Actions, Instance state, Terminate instance**.

Old console

To migrate your application to a compatible instance

1. Back up any data on your instance store volumes that you need to keep to persistent storage. To migrate data on your EBS volumes that you need to keep, create a snapshot of the volumes (see [Creating Amazon EBS snapshots \(p. 1063\)](#)) or detach the volume from the instance so that you can attach it to the new instance later (see [Detaching an Amazon EBS volume from a Linux instance \(p. 1058\)](#)).
2. Launch a new instance, selecting the following:
 - An HVM AMI.
 - The HVM only instance type.
 - If you are using an Elastic IP address, select the VPC that the original instance is currently running in.
 - Any EBS volumes that you detached from the original instance and want to attach to the new instance, or new EBS volumes based on the snapshots that you created.
 - If you want to allow the same traffic to reach the new instance, select the security group that is associated with the original instance.
3. Install your application and any required software on the instance.

4. Restore any data that you backed up from the instance store volumes of the original instance.
5. If you are using an Elastic IP address, assign it to the newly launched instance as follows:
 - a. In the navigation pane, choose **Elastic IPs**.
 - b. Select the Elastic IP address that is associated with the original instance and choose **Actions, Disassociate address**. When prompted for confirmation, choose **Disassociate address**.
 - c. With the Elastic IP address still selected, choose **Actions, Associate address**.
 - d. From **Instance**, select the new instance, and then choose **Associate**.
6. (Optional) You can terminate the original instance if it's no longer needed. Select the instance and verify that you are about to terminate the original instance, not the new instance (for example, check the name or launch time). Choose **Actions, Instance State, Terminate**.

Getting recommendations for an instance type

AWS Compute Optimizer provides Amazon EC2 instance recommendations to help you improve performance, save money, or both. You can use these recommendations to decide whether to move to a new instance type.

To make recommendations, Compute Optimizer analyzes your existing instance specifications and utilization metrics. The compiled data is then used to recommend which Amazon EC2 instance types are best able to handle the existing workload. Recommendations are returned along with per-hour instance pricing.

This topic outlines how to view recommendations through the Amazon EC2 console. For more information, see the [AWS Compute Optimizer User Guide](#).

Note

To get recommendations from Compute Optimizer, you must first opt in to Compute Optimizer. For more information, see [Getting Started with AWS Compute Optimizer](#) in the *AWS Compute Optimizer User Guide*.

Contents

- [Limitations \(p. 300\)](#)
- [Findings \(p. 300\)](#)
- [Viewing recommendations \(p. 301\)](#)
- [Considerations for evaluating recommendations \(p. 302\)](#)

Limitations

Compute Optimizer currently generates recommendations for M, C, R, T, and X instance types. Other instance types are not considered by Compute Optimizer. If you're using other instance types, they will not be listed in the Compute Optimizer recommendations view. For information about these and other instance types, see [Instance types \(p. 200\)](#).

Findings

Compute Optimizer classifies its findings for EC2 instances as follows:

- **Under-provisioned** – An EC2 instance is considered under-provisioned when at least one specification of your instance, such as CPU, memory, or network, does not meet the performance requirements of your workload. Under-provisioned EC2 instances might lead to poor application performance.

- **Over-provisioned** – An EC2 instance is considered over-provisioned when at least one specification of your instance, such as CPU, memory, or network, can be sized down while still meeting the performance requirements of your workload, and when no specification is under-provisioned. Over-provisioned EC2 instances might lead to unnecessary infrastructure cost.
- **Optimized** – An EC2 instance is considered optimized when all specifications of your instance, such as CPU, memory, and network, meet the performance requirements of your workload, and the instance is not over-provisioned. An optimized EC2 instance runs your workloads with optimal performance and infrastructure cost. For optimized instances, Compute Optimizer might sometimes recommend a new generation instance type.
- **None** – There are no recommendations for this instance. This might occur if you've been opted in to Compute Optimizer for less than 12 hours, or when the instance has been running for less than 30 hours, or when the instance type is not supported by Compute Optimizer. For more information, see [Limitations \(p. 300\)](#) in the previous section.

Viewing recommendations

After you opt in to Compute Optimizer, you can view the findings that Compute Optimizer generates for your EC2 instances in the EC2 console. You can then access the Compute Optimizer console to view the recommendations. If you recently opted in, findings might not be reflected in the EC2 console for up to 12 hours.

New console

To view a recommendation for an EC2 instance through the EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and then choose the instance ID.
3. On the instance summary page, find the **AWS Compute Optimizer** banner, and choose **View detail**.

The instance opens in Compute Optimizer, where it is labeled as the **Current** instance. Up to three different instance type recommendations, labeled **Option 1**, **Option 2**, and **Option 3**, are provided. The bottom half of the window shows recent CloudWatch metric data for the current instance: **CPU utilization**, **Memory utilization**, **Network in**, and **Network out**.

4. (Optional) In the Compute Optimizer console, choose the settings () icon to change the visible columns in the table, or to view the public pricing information for a different purchasing option for the current and recommended instance types.

Note

If you've purchased a Reserved Instance, your On-Demand Instance might be billed as a Reserved Instance. Before you change your current instance type, first evaluate the impact on Reserved Instance utilization and coverage.

Old console

To view a recommendation for an EC2 instance through the EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select an instance, and on the **Description** tab, inspect the **Finding** field. Choose **View detail**.

The instance opens in Compute Optimizer, where it is labeled as the **Current** instance. Up to three different instance type recommendations, labeled **Option 1**, **Option 2**, and **Option 3**, are

provided. The bottom half of the window shows recent CloudWatch metric data for the current instance: **CPU utilization**, **Memory utilization**, **Network in**, and **Network out**.

4.

(Optional) In the Compute Optimizer console, choose the settings () icon to change the visible columns in the table, or to view the public pricing information for a different purchasing option for the current and recommended instance types.

Note

If you've purchased a Reserved Instance, your On-Demand Instance might be billed as a Reserved Instance. Before you change your current instance type, first evaluate the impact on Reserved Instance utilization and coverage.

Determine whether you want to use one of the recommendations. Decide whether to optimize for performance improvement, for cost reduction, or for a combination of the two. For more information, see [Viewing Resource Recommendations](#) in the *AWS Compute Optimizer User Guide*.

To view recommendations for all EC2 instances across all Regions through the Compute Optimizer console

1. Open the Compute Optimizer console at <https://console.aws.amazon.com/compute-optimizer/>.
2. Choose **View recommendations for all EC2 instances**.
3. You can perform the following actions on the recommendations page:
 - a. To filter recommendations to one or more AWS Regions, enter the name of the Region in the **Filter by one or more Regions** text box, or choose one or more Regions in the drop-down list that appears.
 - b. To view recommendations for resources in another account, choose **Account**, and then select a different account ID.

This option is available only if you are signed in to a management account of an organization, and you opted in all member accounts within the organization.
 - c. To clear the selected filters, choose **Clear filters**.
 - d. To change the purchasing option that is displayed for the current and recommended instance types, choose the settings () icon , and then choose **On-Demand Instances**, **Reserved Instances, standard 1-year no upfront**, or **Reserved Instances, standard 3-year no upfront**.
 - e. To view details, such as additional recommendations and a comparison of utilization metrics, choose the finding (**Under-provisioned**, **Over-provisioned**, or **Optimized**) listed next to the desired instance. For more information, see [Viewing Resource Details](#) in the *AWS Compute Optimizer User Guide*.

Considerations for evaluating recommendations

Before changing an instance type, consider the following:

- The recommendations don't forecast your usage. Recommendations are based on your historical usage over the most recent 14-day time period. Be sure to choose an instance type that is expected to meet your future resource needs.
- Focus on the graphed metrics to determine whether actual usage is lower than instance capacity. You can also view metric data (average, peak, percentile) in CloudWatch to further evaluate your EC2 instance recommendations. For example, notice how CPU percentage metrics change during the day and whether there are peaks that need to be accommodated. For more information, see [Viewing Available Metrics](#) in the *Amazon CloudWatch User Guide*.
- Compute Optimizer might supply recommendations for burstable performance instances, which are T3, T3a, and T2 instances. If you periodically burst above the baseline, make sure that you can continue

to do so based on the vCPUs of the new instance type. For more information, see [CPU credits and baseline utilization for burstable performance instances \(p. 220\)](#).

- If you've purchased a Reserved Instance, your On-Demand Instance might be billed as a Reserved Instance. Before you change your current instance type, first evaluate the impact on Reserved Instance utilization and coverage.
- Consider conversions to newer generation instances, where possible.
- When migrating to a different instance family, make sure the current instance type and the new instance type are compatible, for example, in terms of virtualization, architecture, or network type. For more information, see [Compatibility for resizing instances \(p. 295\)](#).
- Finally, consider the performance risk rating that's provided for each recommendation. Performance risk indicates the amount of effort you might need to spend in order to validate whether the recommended instance type meets the performance requirements of your workload. We also recommend rigorous load and performance testing before and after making any changes.

There are other considerations when resizing an EC2 instance. For more information, see [Changing the instance type \(p. 294\)](#).

Additional resources

- [Instance types \(p. 200\)](#)
- [AWS Compute Optimizer User Guide](#)

Instance purchasing options

Amazon EC2 provides the following purchasing options to enable you to optimize your costs based on your needs:

- **On-Demand Instances** – Pay, by the second, for the instances that you launch.
- **Savings Plans** – Reduce your Amazon EC2 costs by making a commitment to a consistent amount of usage, in USD per hour, for a term of 1 or 3 years.
- **Reserved Instances** – Reduce your Amazon EC2 costs by making a commitment to a consistent instance configuration, including instance type and Region, for a term of 1 or 3 years.
- **Scheduled Instances** – Purchase instances that are always available on the specified recurring schedule, for a one-year term.
- **Spot Instances** – Request unused EC2 instances, which can reduce your Amazon EC2 costs significantly.
- **Dedicated Hosts** – Pay for a physical host that is fully dedicated to running your instances, and bring your existing per-socket, per-core, or per-VM software licenses to reduce costs.
- **Dedicated Instances** – Pay, by the hour, for instances that run on single-tenant hardware.
- **Capacity Reservations** – Reserve capacity for your EC2 instances in a specific Availability Zone for any duration.

If you require a capacity reservation, purchase Reserved Instances or Capacity Reservations for a specific Availability Zone, or purchase Scheduled Instances. Spot Instances are a cost-effective choice if you can be flexible about when your applications run and if they can be interrupted. Dedicated Hosts or Dedicated Instances can help you address compliance requirements and reduce costs by using your existing server-bound software licenses. For more information, see [Amazon EC2 Pricing](#).

For more information about Savings Plans, see the [AWS Savings Plans User Guide](#).

Contents

- [Determining the instance lifecycle \(p. 304\)](#)
- [On-Demand Instances \(p. 305\)](#)
- [Reserved Instances \(p. 308\)](#)
- [Scheduled Reserved Instances \(p. 347\)](#)
- [Spot Instances \(p. 351\)](#)
- [Dedicated Hosts \(p. 438\)](#)
- [Dedicated Instances \(p. 469\)](#)
- [On-Demand Capacity Reservations \(p. 474\)](#)

Determining the instance lifecycle

The lifecycle of an instance starts when it is launched and ends when it is terminated. The purchasing option that you choose affects the lifecycle of the instance. For example, an On-Demand Instance runs when you launch it and ends when you terminate it. A Spot Instance runs as long as capacity is available and your maximum price is higher than the Spot price. You can launch a Scheduled Instance during its scheduled time period; Amazon EC2 launches the instances and then terminates them three minutes before the time period ends.

Use the following procedure to determine the lifecycle of an instance.

New console

To determine the instance lifecycle using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. On the **Details** tab, under **Instance details**, find **Lifecycle**. If the value is **spot**, the instance is a Spot Instance. If the value is **scheduled**, the instance is a Scheduled Reserved Instance. If the value is **normal**, the instance is either an On-Demand Instance or a Reserved Instance.
5. On the **Details** tab, under **Host and placement group**, find **Tenancy**. If the value is **host**, the instance is running on a Dedicated Host. If the value is **dedicated**, the instance is a Dedicated Instance.
6. (Optional) If you have purchased a Reserved Instance and want to verify that it is being applied, you can check the usage reports for Amazon EC2. For more information, see [Amazon EC2 usage reports \(p. 1243\)](#).

Old console

To determine the instance lifecycle using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. On the **Description** tab, find **Tenancy**. If the value is **host**, the instance is running on a Dedicated Host. If the value is **dedicated**, the instance is a Dedicated Instance.
5. On the **Description** tab, find **Lifecycle**. If the value is **spot**, the instance is a Spot Instance. If the value is **scheduled**, the instance is a Scheduled Instance. If the value is **normal**, the instance is either an On-Demand Instance or a Reserved Instance.

6. (Optional) If you have purchased a Reserved Instance and want to verify that it is being applied, you can check the usage reports for Amazon EC2. For more information, see [Amazon EC2 usage reports \(p. 1243\)](#).

To determine the instance lifecycle using the AWS CLI

Use the following [describe-instances](#) command:

```
aws ec2 describe-instances --instance-ids i-1234567890abcdef0
```

If the instance is running on a Dedicated Host, the output contains the following information:

```
"Tenancy": "host"
```

If the instance is a Dedicated Instance, the output contains the following information:

```
"Tenancy": "dedicated"
```

If the instance is a Spot Instance, the output contains the following information:

```
"InstanceLifecycle": "spot"
```

If the instance is a Scheduled Instance, the output contains the following information:

```
"InstanceLifecycle": "scheduled"
```

Otherwise, the output does not contain `InstanceLifecycle`.

On-Demand Instances

With On-Demand Instances, you pay for compute capacity by the second with no long-term commitments. You have full control over its lifecycle—you decide when to launch, stop, hibernate, start, reboot, or terminate it.

There is no long-term commitment required when you purchase On-Demand Instances. You pay only for the seconds that your On-Demand Instances are in the `running` state. The price per second for a running On-Demand Instance is fixed, and is listed on the [Amazon EC2 Pricing, On-Demand Pricing page](#).

We recommend that you use On-Demand Instances for applications with short-term, irregular workloads that cannot be interrupted.

For significant savings over On-Demand Instances, use [AWS Savings Plans, Spot Instances \(p. 351\)](#), or [Reserved Instances \(p. 308\)](#).

Contents

- [Working with On-Demand Instances \(p. 306\)](#)
- [On-Demand Instance limits \(p. 306\)](#)
 - [Calculating how many vCPUs you need \(p. 307\)](#)
 - [Requesting a limit increase \(p. 308\)](#)
 - [Monitoring On-Demand Instance limits and usage \(p. 308\)](#)

- [Querying the prices of AWS services \(p. 308\)](#)

Working with On-Demand Instances

You can work with On-Demand Instances in the following ways:

- [Launch your instance \(p. 497\)](#)
- [Connect to your Linux instance \(p. 563\)](#)
- [Stop and start your instance \(p. 587\)](#)
- [Hibernate your Linux instance \(p. 591\)](#)
- [Reboot your instance \(p. 603\)](#)
- [Instance retirement \(p. 604\)](#)
- [Terminate your instance \(p. 607\)](#)
- [Recover your instance \(p. 612\)](#)
- [Configuring your Amazon Linux instance \(p. 613\)](#)
- [Identify EC2 Linux instances \(p. 692\)](#)

If you're new to Amazon EC2, see [How to get started with Amazon EC2 \(p. 1\)](#).

On-Demand Instance limits

There is a limit on the number of running On-Demand Instances per AWS account per Region. On-Demand Instance limits are managed in terms of the *number of virtual central processing units (vCPUs)* that your running On-Demand Instances are using, regardless of the instance type.

There are six On-Demand Instance limits, listed in the following table. Each limit specifies the vCPU limit for one or more instance families. For information about the different instance families, generations, and sizes, see [Amazon EC2 Instance Types](#).

On-Demand Instance limit name	Default vCPU limit
Running On-Demand All Standard (A, C, D, H, I, M, R, T, Z) instances	1152 vCPUs
Running On-Demand All F instances	128 vCPUs
Running On-Demand All G instances	128 vCPUs
Running On-Demand All Inf instances	128 vCPUs
Running On-Demand All P instances	128 vCPUs
Running On-Demand All X instances	128 vCPUs

Note

New AWS accounts might start with limits that are lower than the limits described here.

With vCPU limits, you can use your limit in terms of the number of vCPUs required to launch any combination of instance types that meet your changing application needs. For example, with a Standard instance limit of 256 vCPUs, you could launch 32 `m5.2xlarge` instances (32 x 8 vCPUs) or 16 `c5.4xlarge` instances (16 x 16 vCPUs), or a combination of any Standard instance types and sizes that total 256 vCPUs. For more information, see [EC2 On-Demand Instance limits](#).

Calculating how many vCPUs you need

You can use the vCPU limits calculator to determine the number of vCPUs that you require for your application needs.

When using the calculator, keep the following in mind: The calculator assumes that you have reached your current limit. The value that you enter for **Instance count** is the number of instances that you need to launch *in addition* to what is permitted by your current limit. The calculator adds your current limit to the **Instance count** to arrive at a new limit.

The following screenshot shows the vCPU limits calculator.

The screenshot shows a modal window titled "Limits Calculator". It has a header bar with the title and a "Close" button. Below the header is a descriptive text: "Select the instance type and the number of instances you require. The calculator will display the number of vCPUs assigned to the selected instances. Use the New Limit value as a guide for requesting a limit increase." A table lists three instance types with their counts and assigned vCPUs:

Instance type	Instance count	vCPU count	Current limit	New limit
m5.2xlarge	32	256 vCPUs	2,016 vCPUs	2,272 vCPUs
c5.4xlarge	16	256 vCPUs	2,016 vCPUs	2,272 vCPUs
f1.16xlarge	2	128 vCPUs	176 vCPUs	304 vCPUs

A "Add instance type" button is located below the table. Below the table is a section titled "Limits calculation" which displays the current limit, vCPUs needed, and new limit for two limit types:

Instance limit name	Current limit	vCPUs needed	New limit	Options
All Standard (A, C, D, H, I, M, R, T, Z) instances	2,016 vCPUs	512 vCPUs	2,528 vCPUs	Request limit increase
All F instances	176 vCPUs	128 vCPUs	304 vCPUs	Request limit increase

You can view and use the following controls and information:

- **Instance type** – The instance types that you add to the vCPU limits calculator.
- **Instance count** – The number of instances that you require for the selected instance type.
- **vCPU count** – The number of vCPUs that corresponds to the **Instance count**.
- **Current limit** – Your current limit for the limit type to which the instance type belongs. The limit applies to all instance types of the same limit type. For example, in the preceding screenshot, the current limit for `m5.2xlarge` and `c5.4xlarge` is 1,920 vCPUs, which is the limit for all the instance types that belong to the All Standard instances limit.
- **New limit** – The new limit, in number of vCPUs, which is calculated by adding **vCPU count** and **Current limit**.
- **X** – Choose the **X** to remove the row.
- **Add instance type** – Choose **Add instance type** to add another instance type to the calculator.
- **Limits calculation** – Displays the current limit, vCPUs needed, and new limit for the limit types.
 - **Instance limit name** – The limit type for the instance types that you selected.
 - **Current limit** – The current limit for the limit type.
 - **vCPUs needed** – The number of vCPUs that corresponds to the number of instances that you specified in **Instance count**. For the All Standard instances limit type, the vCPUs needed is calculated by adding the values for **vCPU count** for all the instance types of this limit type.
 - **New limit** – The new limit is calculated by adding **Current limit** and **vCPUs needed**.
- **Options** – Choose **Request limit increase** to request a limit increase for the corresponding limit type.

To calculate the number of required vCPUs

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select a Region.
3. From the left navigator, choose **Limits**.
4. Choose **Calculate vCPU limit**.
5. Choose **Add instance type**, choose the required instance type, and specify the required number of instances. To add more instance types, choose **Add instance type** again.
6. View **Limits calculation** for the required new limit.
7. When you've finished using the calculator, choose **Close**.

Requesting a limit increase

You can request a limit increase for each On-Demand Instance limit type from the [Limits page](#) or the vCPU limits calculator in the Amazon EC2 console. Complete the required fields on the AWS Support Center [limit increase form](#) with your use case. For **Primary Instance Type**, select the limit type that corresponds to the **Instance limit name** in the vCPU limits calculator. For the new limit value, use the value that appears in the **New limit** column in the vCPU limits calculator. For more information about requesting a limit increase, see [Amazon EC2 service quotas \(p. 1241\)](#).

Monitoring On-Demand Instance limits and usage

You can view and manage your On-Demand Instance limits using the following:

- The [Limits page](#) in the Amazon EC2 console
- The Amazon EC2 [Services quotas page](#) in the Service Quotas console
- The [get-service-quota](#) AWS CLI
- The [Service Limits page](#) in the AWS Trusted Advisor console

For more information, see [Amazon EC2 service quotas \(p. 1241\)](#) in the *Amazon EC2 User Guide for Linux Instances*, [Viewing a Service Quota](#) in the *Service Quotas User Guide*, and [AWS Trusted Advisor](#).

With Amazon CloudWatch metrics integration, you can monitor EC2 usage against limits. You can also configure alarms to warn about approaching limits. For more information, see [Using Amazon CloudWatch Alarms](#) in the *Service Quotas User Guide*.

Querying the prices of AWS services

You can use the Price List Service API or the AWS Price List API to query the prices of On-Demand Instances. For more information, see [Using the AWS Price List API](#) in the *AWS Billing and Cost Management User Guide*.

Reserved Instances

Reserved Instances provide you with significant savings on your Amazon EC2 costs compared to On-Demand Instance pricing. Reserved Instances are not physical instances, but rather a billing discount applied to the use of On-Demand Instances in your account. These On-Demand Instances must match certain attributes, such as instance type and Region, in order to benefit from the billing discount.

Savings Plans also offer significant savings on your Amazon EC2 costs compared to On-Demand Instance pricing. With Savings Plans, you make a commitment to a consistent usage amount, measured in USD per hour. This provides you with the flexibility to use the instance configurations that best meet your needs

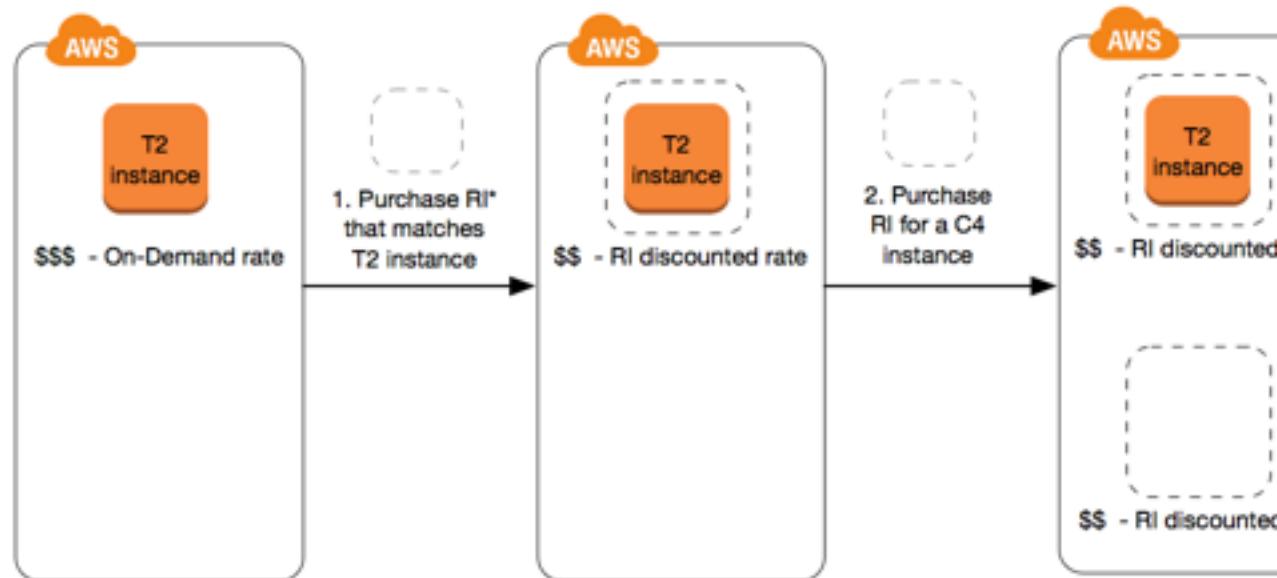
and continue to save money, instead of making a commitment to a specific instance configuration. For more information, see the [AWS Savings Plans User Guide](#).

Reserved Instances topics

- Reserved Instance overview (p. 309)
- Key variables that determine Reserved Instance pricing (p. 309)
- Reserved Instance limits (p. 311)
- Regional and zonal Reserved Instances (scope) (p. 311)
- Types of Reserved Instances (offering classes) (p. 312)
- How Reserved Instances are applied (p. 313)
- How you are billed (p. 318)
- Buying Reserved Instances (p. 322)
- Reserved Instance Marketplace (p. 329)
- Modifying Reserved Instances (p. 335)
- Exchanging Convertible Reserved Instances (p. 342)

Reserved Instance overview

The following diagram shows a basic overview of purchasing and using Reserved Instances.



*RI = Reserved Instance

In this scenario, you have a running On-Demand Instance (T2) in your account, for which you're currently paying On-Demand rates. You purchase a Reserved Instance that matches the attributes of your running instance, and the billing benefit is immediately applied. Next, you purchase a Reserved Instance for a C4 instance. You do not have any running instances in your account that match the attributes of this Reserved Instance. In the final step, you launch an instance that matches the attributes of the C4 Reserved Instance, and the billing benefit is immediately applied.

Key variables that determine Reserved Instance pricing

The Reserved Instance pricing is determined by the following key variables.

Instance attributes

A Reserved Instance has four instance attributes that determine its price.

- **Instance type:** For example, `m4.large`. This is composed of the instance family (for example, `m4`) and the instance size (for example, `large`).
- **Region:** The Region in which the Reserved Instance is purchased.
- **Tenancy:** Whether your instance runs on shared (default) or single-tenant (dedicated) hardware. For more information, see [Dedicated Instances \(p. 469\)](#).
- **Platform:** The operating system; for example, Windows or Linux/Unix. For more information, see [Choosing a platform \(p. 323\)](#).

Term commitment

You can purchase a Reserved Instance for a one-year or three-year commitment, with the three-year commitment offering a bigger discount.

- **One-year:** A year is defined as 31536000 seconds (365 days).
- **Three-year:** Three years is defined as 94608000 seconds (1095 days).

Reserved Instances do not renew automatically; when they expire, you can continue using the EC2 instance without interruption, but you are charged On-Demand rates. In the above example, when the Reserved Instances that cover the T2 and C4 instances expire, you go back to paying the On-Demand rates until you terminate the instances or purchase new Reserved Instances that match the instance attributes.

Payment options

The following payment options are available for Reserved Instances:

- **All Upfront:** Full payment is made at the start of the term, with no other costs or additional hourly charges incurred for the remainder of the term, regardless of hours used.
- **Partial Upfront:** A portion of the cost must be paid upfront and the remaining hours in the term are billed at a discounted hourly rate, regardless of whether the Reserved Instance is being used.
- **No Upfront:** You are billed a discounted hourly rate for every hour within the term, regardless of whether the Reserved Instance is being used. No upfront payment is required.

Note

No Upfront Reserved Instances are based on a contractual obligation to pay monthly for the entire term of the reservation. For this reason, a successful billing history is required before you can purchase No Upfront Reserved Instances.

Generally speaking, you can save more money making a higher upfront payment for Reserved Instances. You can also find Reserved Instances offered by third-party sellers at lower prices and shorter term lengths on the Reserved Instance Marketplace. For more information, see [Reserved Instance Marketplace \(p. 329\)](#).

Offering class

If your computing needs change, you may be able to modify or exchange your Reserved Instance, depending on the offering class.

- **Standard:** These provide the most significant discount, but can only be modified.

- **Convertible:** These provide a lower discount than Standard Reserved Instances, but can be exchanged for another Convertible Reserved Instance with different instance attributes. Convertible Reserved Instances can also be modified.

For more information, see [Types of Reserved Instances \(offering classes\) \(p. 312\)](#).

After you purchase a Reserved Instance, you cannot cancel your purchase. However, you may be able to [modify \(p. 335\)](#), [exchange \(p. 342\)](#), or [sell \(p. 329\)](#) your Reserved Instance if your needs change.

For more information, see the [Amazon EC2 Reserved Instances Pricing page](#).

Reserved Instance limits

There is a limit to the number of Reserved Instances that you can purchase per month. For each Region you can purchase 20 [regional \(p. 313\)](#) Reserved Instances per month plus an additional 20 [zonal \(p. 313\)](#) Reserved Instances per month for each Availability Zone.

For example, in a Region with three Availability Zones, the limit is 80 Reserved Instances per month: 20 regional Reserved Instances for the Region plus 20 zonal Reserved Instances for each of the three Availability Zones ($20 \times 3 = 60$).

A regional Reserved Instance applies a discount to a running On-Demand Instance. The default On-Demand Instance limit is 20. You cannot exceed your running On-Demand Instance limit by purchasing regional Reserved Instances. For example, if you already have 20 running On-Demand Instances, and you purchase 20 regional Reserved Instances, the 20 regional Reserved Instances are used to apply a discount to the 20 running On-Demand Instances. If you purchase more regional Reserved Instances, you will not be able to launch more instances because you have reached your On-Demand Instance limit.

Before purchasing regional Reserved Instances, make sure your On-Demand Instance limit matches or exceeds the number of regional Reserved Instances you intend to own. If required, make sure you request an increase to your On-Demand Instance limit *before* purchasing more regional Reserved Instances.

A zonal Reserved Instance—a Reserved Instance that is purchased for a specific Availability Zone—provides capacity reservation as well as a discount. You *can exceed* your running On-Demand Instance limit by purchasing zonal Reserved Instances. For example, if you already have 20 running On-Demand Instances, and you purchase 20 zonal Reserved Instances, you can launch a further 20 On-Demand Instances that match the specifications of your zonal Reserved Instances, giving you a total of 40 running instances.

The Amazon EC2 console provides limit information. For more information, see [Viewing your current limits \(p. 1241\)](#).

Regional and zonal Reserved Instances (scope)

When you purchase a Reserved Instance, you determine the scope of the Reserved Instance. The scope is either regional or zonal.

- **Regional:** When you purchase a Reserved Instance for a Region, it's referred to as a *regional* Reserved Instance.
- **Zonal:** When you purchase a Reserved Instance for a specific Availability Zone, it's referred to as a *zonal* Reserved Instance.

Differences between regional and zonal Reserved Instances

The following table highlights some key differences between regional Reserved Instances and zonal Reserved Instances:

	Regional Reserved Instances	Zonal Reserved Instances
Availability Zone flexibility	The Reserved Instance discount applies to instance usage in any Availability Zone in the specified Region.	No Availability Zone flexibility—the Reserved Instance discount applies to instance usage in the specified Availability Zone only.
Capacity reservation	No capacity reservation—a regional Reserved Instance does <i>not</i> provide a capacity reservation.	A zonal Reserved Instance provides a capacity reservation in the specified Availability Zone.
Instance size flexibility	The Reserved Instance discount applies to instance usage within the instance family, regardless of size. Only supported on Amazon Linux/Unix Reserved Instances with default tenancy. For more information, see Instance size flexibility determined by normalization factor (p. 313) .	No instance size flexibility—the Reserved Instance discount applies to instance usage for the specified instance type and size only.

For more information and examples, see [How Reserved Instances are applied \(p. 313\)](#).

Types of Reserved Instances (offering classes)

When you purchase a Reserved Instance, you can choose between a Standard or Convertible offering class. The Reserved Instance applies to a single instance type, platform, scope, and tenancy over a term. If your computing needs change, you may be able to modify or exchange your Reserved Instance, depending on the offering class. Offering classes may also have additional restrictions or limitations.

The following are the differences between Standard and Convertible offering classes.

Standard Reserved Instance	Convertible Reserved Instance
Some attributes, such as instance size, can be modified during the term; however, the instance family cannot be modified. You cannot exchange a Standard Reserved Instance, only modify it. For more information, see Modifying Reserved Instances (p. 335) .	Can be exchanged during the term for another Convertible Reserved Instance with new attributes including instance family, instance type, platform, scope, or tenancy. For more information, see Exchanging Convertible Reserved Instances (p. 342) . You can also modify some attributes of a Convertible Reserved Instance. For more information, see Modifying Reserved Instances (p. 335) .
Can be sold in the Reserved Instance Marketplace.	Cannot be sold in the Reserved Instance Marketplace.

Standard and Convertible Reserved Instances can be purchased to apply to instances in a specific Availability Zone (zonal Reserved Instances), or to instances in a Region (regional Reserved Instances). For more information and examples, see [How Reserved Instances are applied \(p. 313\)](#).

If you want to purchase capacity reservations that recur on a daily, weekly, or monthly basis, a Scheduled Reserved Instance may meet your needs. For more information, see [Scheduled Reserved Instances \(p. 347\)](#).

How Reserved Instances are applied

If you purchase a Reserved Instance and you already have a running instance that matches the specifications of the Reserved Instance, the billing benefit is immediately applied. You do not have to restart your instances. If you do not have an eligible running instance, launch an instance and ensure that you match the same criteria that you specified for your Reserved Instance. For more information, see [Using your Reserved Instances \(p. 328\)](#).

Reserved Instances apply to usage in the same manner, irrespective of the offering type (Standard or Convertible), and are automatically applied to running On-Demand Instances with matching attributes.

How zonal Reserved Instances are applied

Reserved Instances assigned to a specific Availability Zone provide the Reserved Instance discount to matching instance usage in that Availability Zone. For example, if you purchase two `c4.xlarge` default tenancy Linux/Unix Standard Reserved Instances in Availability Zone us-east-1a, then up to two `c4.xlarge` default tenancy Linux/Unix instances running in the Availability Zone us-east-1a can benefit from the Reserved Instance discount. The attributes (tenancy, platform, Availability Zone, instance type, and instance size) of the running instances must match that of the Reserved Instances.

How regional Reserved Instances are applied

Regional Reserved Instances are purchased for a Region and provide Availability Zone flexibility. The Reserved Instance discount applies to instance usage in any Availability Zone in that Region.

Regional Reserved Instances also provide instance size flexibility where the Reserved Instance discount applies to instance usage within the instance family, regardless of size.

Limitations for instance size flexibility

Instance size flexibility does not apply to the following Reserved Instances:

- Reserved Instances that are purchased for a specific Availability Zone (zonal Reserved Instances)
- Reserved Instances with dedicated tenancy
- Reserved Instances for Windows Server, Windows Server with SQL Standard, Windows Server with SQL Server Enterprise, Windows Server with SQL Server Web, RHEL, and SUSE Linux Enterprise Server
- Reserved Instances for G4 instances

Instance size flexibility determined by normalization factor

Instance size flexibility is determined by the normalization factor of the instance size. The discount applies either fully or partially to running instances of the same instance family, depending on the instance size of the reservation, in any Availability Zone in the Region. The only attributes that must be matched are the instance family, tenancy, and platform.

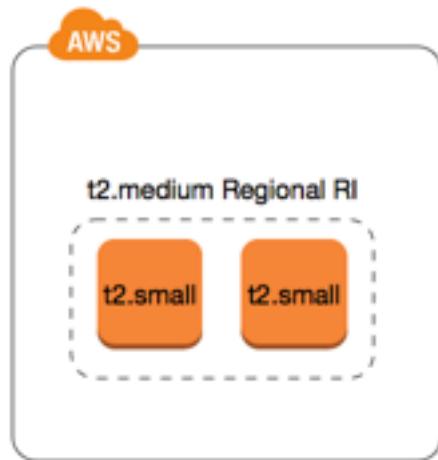
Instance size flexibility is applied from the smallest to the largest instance size within the instance family based on the normalization factor.

The following table lists the different sizes within an instance family, and the corresponding normalization factor per hour. This scale is used to apply the discounted rate of Reserved Instances to the normalized usage of the instance family.

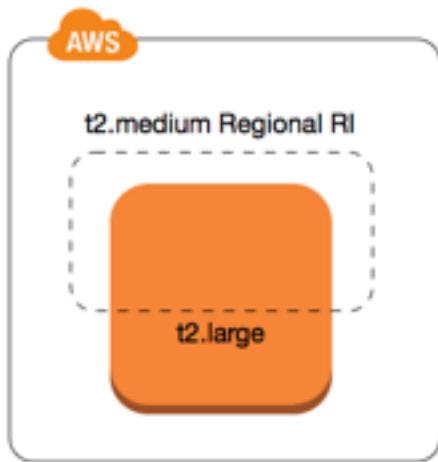
Instance size	Normalization factor
nano	0.25
micro	0.5

Instance size	Normalization factor
small	1
medium	2
large	4
xlarge	8
2xlarge	16
3xlarge	24
4xlarge	32
6xlarge	48
8xlarge	64
9xlarge	72
10xlarge	80
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256

For example, a `t2.medium` instance has a normalization factor of 2. If you purchase a `t2.medium` default tenancy Amazon Linux/Unix Reserved Instance in the US East (N. Virginia) and you have two running `t2.small` instances in your account in that Region, the billing benefit is applied in full to both instances.



Or, if you have one `t2.large` instance running in your account in the US East (N. Virginia) Region, the billing benefit is applied to 50% of the usage of the instance.



The normalization factor is also applied when modifying Reserved Instances. For more information, see [Modifying Reserved Instances \(p. 335\)](#).

Normalization factor for bare metal instances

Instance size flexibility also applies to bare metal instances within the instance family. If you have regional Amazon Linux/Unix Reserved Instances with shared tenancy on bare metal instances, you can benefit from the Reserved Instance savings within the same instance family. The opposite is also true: if you have regional Amazon Linux/Unix Reserved Instances with shared tenancy on instances in the same family as a bare metal instance, you can benefit from the Reserved Instance savings on the bare metal instance.

A bare metal instance is the same size as the largest instance within the same instance family. For example, an `i3.metal` is the same size as an `i3.16xlarge`, so they have the same normalization factor.

Note

The `.metal` instance sizes do not have a single normalization factor. They vary based on the specific instance family.

Bare metal instance size	Normalization factor
<code>a1.metal</code>	32
<code>c5.metal</code>	192
<code>c5d.metal</code>	192
<code>c5n.metal</code>	144
<code>c6g.metal</code>	128
<code>c6gd.metal</code>	128
<code>g4dn.metal</code>	128
<code>i3.metal</code>	128
<code>i3en.metal</code>	192
<code>m5.metal</code>	192
<code>m5d.metal</code>	192

Bare metal instance size	Normalization factor
m6g.metal	128
m6gd.metal	128
r5.metal	192
r5d.metal	192
r6g.metal	128
r6gd.metal	128
z1d.metal	96

For example, an `i3.metal` instance has a normalization factor of 128. If you purchase an `i3.metal` default tenancy Amazon Linux/Unix Reserved Instance in the US East (N. Virginia), the billing benefit can apply as follows:

- If you have one running `i3.16xlarge` in your account in that Region, the billing benefit is applied in full to the `i3.16xlarge` instance (`i3.16xlarge` normalization factor = 128).
- Or, if you have two running `i3.8xlarge` instances in your account in that Region, the billing benefit is applied in full to both `i3.8xlarge` instances (`i3.8xlarge` normalization factor = 64).
- Or, if you have four running `i3.4xlarge` instances in your account in that Region, the billing benefit is applied in full to all four `i3.4xlarge` instances (`i3.4xlarge` normalization factor = 32).

The opposite is also true. For example, if you purchase two `i3.8xlarge` default tenancy Amazon Linux/Unix Reserved Instances in the US East (N. Virginia), and you have one running `i3.metal` instance in that Region, the billing benefit is applied in full to the `i3.metal` instance.

Examples of applying Reserved Instances

The following scenarios cover the ways in which Reserved Instances are applied.

Example Scenario 1: Reserved Instances in a single account

You are running the following On-Demand Instances in account A:

- 4 x `m3.large` Linux, default tenancy instances in Availability Zone us-east-1a
- 2 x `m4.xlarge` Amazon Linux, default tenancy instances in Availability Zone us-east-1b
- 1 x `c4.xlarge` Amazon Linux, default tenancy instances in Availability Zone us-east-1c

You purchase the following Reserved Instances in account A:

- 4 x `m3.large` Linux, default tenancy Reserved Instances in Availability Zone us-east-1a (capacity is reserved)
- 4 x `m4.large` Amazon Linux, default tenancy Reserved Instances in Region us-east-1
- 1 x `c4.large` Amazon Linux, default tenancy Reserved Instances in Region us-east-1

The Reserved Instance benefits are applied in the following way:

- The discount and capacity reservation of the four `m3.large` zonal Reserved Instances is used by the four `m3.large` instances because the attributes (instance size, Region, platform, tenancy) between them match.

- The **m4.large** regional Reserved Instances provide Availability Zone and instance size flexibility, because they are regional Amazon Linux Reserved Instances with default tenancy.

An **m4.large** is equivalent to 4 normalized units/hour.

You've purchased four **m4.large** regional Reserved Instances, and in total, they are equal to 16 normalized units/hour (4x4). Account A has two **m4.xlarge** instances running, which is equivalent to 16 normalized units/hour (2x8). In this case, the four **m4.large** regional Reserved Instances provide the billing benefit to an entire hour of usage of the two **m4.xlarge** instances.

- The **c4.large** regional Reserved Instance in us-east-1 provides Availability Zone and instance size flexibility, because it is a regional Amazon Linux Reserved Instance with default tenancy, and applies to the **c4.xlarge** instance. A **c4.large** instance is equivalent to 4 normalized units/hour and a **c4.xlarge** is equivalent to 8 normalized units/hour.

In this case, the **c4.large** regional Reserved Instance provides partial benefit to **c4.xlarge** usage. This is because the **c4.large** Reserved Instance is equivalent to 4 normalized units/hour of usage, but the **c4.xlarge** instance requires 8 normalized units/hour. Therefore, the **c4.large** Reserved Instance billing discount applies to 50% of **c4.xlarge** usage. The remaining **c4.xlarge** usage is charged at the On-Demand rate.

Example Scenario 2: Regional Reserved Instances in linked accounts

Reserved Instances are first applied to usage within the purchasing account, followed by qualifying usage in any other account in the organization. For more information, see [Reserved Instances and consolidated billing \(p. 320\)](#). For regional Reserved Instances that offer instance size flexibility, the benefit is applied from the smallest to the largest instance size within the instance family.

You're running the following On-Demand Instances in account A (the purchasing account):

- 2 x **m4.xlarge** Linux, default tenancy instances in Availability Zone us-east-1a
- 1 x **m4.2xlarge** Linux, default tenancy instances in Availability Zone us-east-1b
- 2 x **c4.xlarge** Linux, default tenancy instances in Availability Zone us-east-1a
- 1 x **c4.2xlarge** Linux, default tenancy instances in Availability Zone us-east-1b

Another customer is running the following On-Demand Instances in account B—a linked account:

- 2 x **m4.xlarge** Linux, default tenancy instances in Availability Zone us-east-1a

You purchase the following regional Reserved Instances in account A:

- 4 x **m4.xlarge** Linux, default tenancy Reserved Instances in Region us-east-1
- 2 x **c4.xlarge** Linux, default tenancy Reserved Instances in Region us-east-1

The regional Reserved Instance benefits are applied in the following way:

- The discount of the four **m4.xlarge** Reserved Instances is used by the two **m4.xlarge** instances and the single **m4.2xlarge** instance in account A (purchasing account). All three instances match the attributes (instance family, Region, platform, tenancy). The discount is applied to instances in the purchasing account (account A) first, even though account B (linked account) has two **m4.xlarge** that also match the Reserved Instances. There is no capacity reservation because the Reserved Instances are regional Reserved Instances.
- The discount of the two **c4.xlarge** Reserved Instances applies to the two **c4.xlarge** instances, because they are a smaller instance size than the **c4.2xlarge** instance. There is no capacity reservation because the Reserved Instances are regional Reserved Instances.

Example Scenario 3: Zonal Reserved Instances in a linked account

In general, Reserved Instances that are owned by an account are applied first to usage in that account. However, if there are qualifying, unused Reserved Instances for a specific Availability Zone (zonal Reserved Instances) in other accounts in the organization, they are applied to the account before regional Reserved Instances owned by the account. This is done to ensure maximum Reserved Instance utilization and a lower bill. For billing purposes, all the accounts in the organization are treated as one account. The following example may help explain this.

You're running the following On-Demand Instance in account A (the purchasing account):

- 1 x m4.xlarge Linux, default tenancy instance in Availability Zone us-east-1a

A customer is running the following On-Demand Instance in linked account B:

- 1 x m4.xlarge Linux, default tenancy instance in Availability Zone us-east-1b

You purchase the following regional Reserved Instances in account A:

- 1 x m4.xlarge Linux, default tenancy Reserved Instance in Region us-east-1

A customer also purchases the following zonal Reserved Instances in linked account C:

- 1 x m4.xlarge Linux, default tenancy Reserved Instances in Availability Zone us-east-1a

The Reserved Instance benefits are applied in the following way:

- The discount of the m4.xlarge zonal Reserved Instance owned by account C is applied to the m4.xlarge usage in account A.
- The discount of the m4.xlarge regional Reserved Instance owned by account A is applied to the m4.xlarge usage in account B.
- If the regional Reserved Instance owned by account A was first applied to the usage in account A, the zonal Reserved Instance owned by account C remains unused and usage in account B is charged at On-Demand rates.

For more information, see [Reserved Instances in the Billing and Cost Management Report](#).

How you are billed

All Reserved Instances provide you with a discount compared to On-Demand pricing. With Reserved Instances, you pay for the entire term regardless of actual use. You can choose to pay for your Reserved Instance upfront, partially upfront, or monthly, depending on the [payment option \(p. 310\)](#) specified for the Reserved Instance.

When Reserved Instances expire, you are charged On-Demand rates for EC2 instance usage. You can queue a Reserved Instance for purchase up to three years in advance. This can help you ensure that you have uninterrupted coverage. For more information, see [Queueing your purchase \(p. 324\)](#).

The AWS Free Tier is available for new AWS accounts. If you are using the AWS Free Tier to run Amazon EC2 instances, and you purchase a Reserved Instance, you are charged under standard pricing guidelines. For information, see [AWS Free Tier](#).

Contents

- [Usage billing \(p. 319\)](#)

- [Viewing your bill \(p. 320\)](#)
- [Reserved Instances and consolidated billing \(p. 320\)](#)
- [Reserved Instance discount pricing tiers \(p. 321\)](#)

Usage billing

Reserved Instances are billed for every clock-hour during the term that you select, regardless of whether an instance is running. Each clock-hour starts on the hour (zero minutes and zero seconds past the hour) of a standard 24-hour clock. For example, 1:00:00 to 1:59:59 is one clock-hour. For more information about instance states, see [Instance lifecycle \(p. 493\)](#).

A Reserved Instance billing benefit can be applied to a running instance on a per-second basis. Per-second billing is available for instances using an open-source Linux distribution, such as Amazon Linux and Ubuntu. Per-hour billing is used for commercial Linux distributions, such as Red Hat Enterprise Linux and SUSE Linux Enterprise Server.

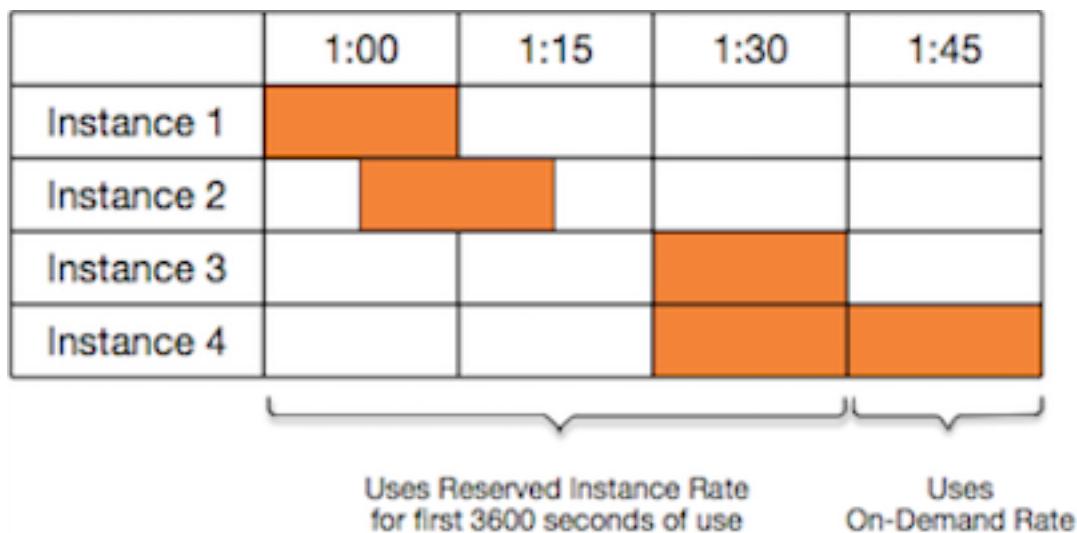
A Reserved Instance billing benefit can apply to a maximum of 3600 seconds (one hour) of instance usage per clock-hour. You can run multiple instances concurrently, but can only receive the benefit of the Reserved Instance discount for a total of 3600 seconds per clock-hour; instance usage that exceeds 3600 seconds in a clock-hour is billed at the On-Demand rate.

For example, if you purchase one m4.xlarge Reserved Instance and run four m4.xlarge instances concurrently for one hour, one instance is charged at one hour of Reserved Instance usage and the other three instances are charged at three hours of On-Demand usage.

However, if you purchase one m4.xlarge Reserved Instance and run four m4.xlarge instances for 15 minutes (900 seconds) each within the same hour, the total running time for the instances is one hour, which results in one hour of Reserved Instance usage and 0 hours of On-Demand usage.

	1:00	1:15	1:30	1:45
Instance 1	Orange			
Instance 2		Orange		
Instance 3			Orange	
Instance 4				Orange

If multiple eligible instances are running concurrently, the Reserved Instance billing benefit is applied to all the instances at the same time up to a maximum of 3600 seconds in a clock-hour; thereafter, On-Demand rates apply.



Cost Explorer on the [Billing and Cost Management](#) console enables you to analyze the savings against running On-Demand Instances. The [Reserved Instances FAQ](#) includes an example of a list value calculation.

If you close your AWS account, On-Demand billing for your resources stops. However, if you have any Reserved Instances in your account, you continue to receive a bill for these until they expire.

Viewing your bill

You can find out about the charges and fees to your account by viewing the [AWS Billing and Cost Management](#) console.

- The **Dashboard** displays a spend summary for your account.
- On the **Bills** page, under **Details** expand the **Elastic Compute Cloud** section and the Region to get billing information about your Reserved Instances.

You can view the charges online, or you can download a CSV file.

You can also track your Reserved Instance utilization using the AWS Cost and Usage Report. For more information, see [Reserved Instances](#) under Cost and Usage Report in the [AWS Billing and Cost Management User Guide](#).

Reserved Instances and consolidated billing

The pricing benefits of Reserved Instances are shared when the purchasing account is part of a set of accounts billed under one consolidated billing payer account. The instance usage across all member accounts is aggregated in the payer account every month. This is typically useful for companies in which there are different functional teams or groups; then, the normal Reserved Instance logic is applied to calculate the bill. For more information, see [Consolidated Billing and AWS Organizations](#) in the [AWS Organizations User Guide](#).

If you close the account that purchased the Reserved Instance, the payer account will continue being charged for the Reserved Instance until either the Reserved Instance expires or the closed account is permanently deleted. The closed account is permanently deleted after 90 days. After it is deleted, the member accounts will stop benefitting from the Reserved Instance billing discount. For more information about closing an account, see [Closing an AWS Account](#) in the [AWS Organizations User Guide](#).

Reserved Instance discount pricing tiers

If your account qualifies for a discount pricing tier, it automatically receives discounts on upfront and instance usage fees for Reserved Instance purchases that you make within that tier level from that point on. To qualify for a discount, the list value of your Reserved Instances in the Region must be \$500,000 USD or more.

The following rules apply:

- Pricing tiers and related discounts apply only to purchases of Amazon EC2 Standard Reserved Instances.
- Pricing tiers do not apply to Reserved Instances for Windows with SQL Server Standard, SQL Server Web, and SQL Server Enterprise.
- Pricing tiers do not apply to Reserved Instances for Linux with SQL Server Standard, SQL Server Web, and SQL Server Enterprise.
- Pricing tier discounts only apply to purchases made from AWS. They do not apply to purchases of third-party Reserved Instances.
- Discount pricing tiers are currently not applicable to Convertible Reserved Instance purchases.

Topics

- [Calculating Reserved Instance pricing discounts \(p. 321\)](#)
- [Buying with a discount tier \(p. 322\)](#)
- [Crossing pricing tiers \(p. 322\)](#)
- [Consolidated billing for pricing tiers \(p. 322\)](#)

Calculating Reserved Instance pricing discounts

You can determine the pricing tier for your account by calculating the list value for all of your Reserved Instances in a Region. Multiply the hourly recurring price for each reservation by the total number of hours for the term and add the undiscounted upfront price (also known as the fixed price) at the time of purchase. Because the list value is based on undiscounted (public) pricing, it is not affected if you qualify for a volume discount or if the price drops after you buy your Reserved Instances.

```
List value = fixed price + (undiscounted recurring hourly price * hours in term)
```

For example, for a 1-year Partial Upfront t2.small Reserved Instance, assume the upfront price is \$60.00 and the hourly rate is \$0.007. This provides a list value of \$121.32.

```
121.32 = 60.00 + (0.007 * 8760)
```

To view the fixed price values for Reserved Instances using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. Display the **Upfront Price** column by choosing **Show/Hide Columns** (the gear-shaped icon) in the top right corner.

To view the fixed price values for Reserved Instances using the command line

- [describe-reserved-instances \(AWS CLI\)](#)
- [Get-EC2ReservedInstance \(AWS Tools for Windows PowerShell\)](#)

- [DescribeReservedInstances](#) (Amazon EC2 API)

Buying with a discount tier

When you buy Reserved Instances, Amazon EC2 automatically applies any discounts to the part of your purchase that falls within a discount pricing tier. You don't need to do anything differently, and you can buy Reserved Instances using any of the Amazon EC2 tools. For more information, see [Buying Reserved Instances \(p. 322\)](#).

After the list value of your active Reserved Instances in a Region crosses into a discount pricing tier, any future purchase of Reserved Instances in that Region are charged at a discounted rate. If a single purchase of Reserved Instances in a Region takes you over the threshold of a discount tier, then the portion of the purchase that is above the price threshold is charged at the discounted rate. For more information about the temporary Reserved Instance IDs that are created during the purchase process, see [Crossing pricing tiers \(p. 322\)](#).

If your list value falls below the price point for that discount pricing tier—for example, if some of your Reserved Instances expire—future purchases of Reserved Instances in the Region are not discounted. However, you continue to get the discount applied against any Reserved Instances that were originally purchased within the discount pricing tier.

When you buy Reserved Instances, one of four possible scenarios occurs:

- **No discount**—Your purchase within a Region is still below the discount threshold.
- **Partial discount**—Your purchase within a Region crosses the threshold of the first discount tier. No discount is applied to one or more reservations and the discounted rate is applied to the remaining reservations.
- **Full discount**—Your entire purchase within a Region falls within one discount tier and is discounted appropriately.
- **Two discount rates**—Your purchase within a Region crosses from a lower discount tier to a higher discount tier. You are charged two different rates: one or more reservations at the lower discounted rate, and the remaining reservations at the higher discounted rate.

Crossing pricing tiers

If your purchase crosses into a discounted pricing tier, you see multiple entries for that purchase: one for that part of the purchase charged at the regular price, and another for that part of the purchase charged at the applicable discounted rate.

The Reserved Instance service generates several Reserved Instance IDs because your purchase crossed from an undiscounted tier, or from one discounted tier to another. There is an ID for each set of reservations in a tier. Consequently, the ID returned by your purchase CLI command or API action is different from the actual ID of the new Reserved Instances.

Consolidated billing for pricing tiers

A consolidated billing account aggregates the list value of member accounts within a Region. When the list value of all active Reserved Instances for the consolidated billing account reaches a discount pricing tier, any Reserved Instances purchased after this point by any member of the consolidated billing account are charged at the discounted rate (as long as the list value for that consolidated account stays above the discount pricing tier threshold). For more information, see [Reserved Instances and consolidated billing \(p. 320\)](#).

Buying Reserved Instances

To purchase a Reserved Instance, search for *Reserved Instance offerings* from AWS and third-party sellers, adjusting your search parameters until you find the exact match that you're looking for.

When you search for Reserved Instances to buy, you receive a quote on the cost of the returned offerings. When you proceed with the purchase, AWS automatically places a limit price on the purchase price. The total cost of your Reserved Instances won't exceed the amount that you were quoted.

If the price rises or changes for any reason, the purchase is not completed. If, at the time of purchase, there are offerings similar to your choice but at a lower price, AWS sells you the offerings at the lower price.

Before you confirm your purchase, review the details of the Reserved Instance that you plan to buy, and make sure that all the parameters are accurate. After you purchase a Reserved Instance (either from a third-party seller in the Reserved Instance Marketplace or from AWS), you cannot cancel your purchase.

Note

To purchase and modify Reserved Instances, ensure that your IAM user account has the appropriate permissions, such as the ability to describe Availability Zones. For information, see [Example Policies for Working With the AWS CLI or an AWS SDK](#) and [Example Policies for Working in the Amazon EC2 Console](#).

Tasks

- [Choosing a platform \(p. 323\)](#)
- [Queuing your purchase \(p. 324\)](#)
- [Buying Standard Reserved Instances \(p. 324\)](#)
- [Buying Convertible Reserved Instances \(p. 326\)](#)
- [Viewing your Reserved Instances \(p. 327\)](#)
- [Canceling a queued purchase \(p. 328\)](#)
- [Renewing a Reserved Instance \(p. 328\)](#)
- [Using your Reserved Instances \(p. 328\)](#)

Choosing a platform

Amazon EC2 supports the following Linux platforms for Reserved Instances:

- Linux/UNIX
- Linux with SQL Server Standard
- Linux with SQL Server Web
- Linux with SQL Server Enterprise
- SUSE Linux
- Red Hat Enterprise Linux

When you purchase a Reserved Instance, you must choose an offering for a *platform* that represents the operating system for your instance.

- For SUSE Linux and RHEL distributions, you must choose offerings for those specific platforms, i.e., for the **SUSE Linux** or **Red Hat Enterprise Linux** platforms.
- For all other Linux distributions (including Ubuntu), choose an offering for the **Linux/UNIX** platform.
- If you bring your existing RHEL subscription, you must choose an offering for the **Linux/UNIX** platform, not an offering for the **Red Hat Enterprise Linux** platform.

Important

- If you purchase a Reserved Instance to apply to an On-Demand Instance that was launched from an AMI with a billing product code, make sure that the Reserved Instance has the

matching billing product code. If you purchase a Reserved Instance without the matching billing product code, the Reserved Instance will not be applied to the On-Demand Instance. For more information about how to obtain the AMI billing code, see [Obtaining billing information \(p. 169\)](#).

- If you plan to purchase a Reserved Instance to apply to an On-Demand Instance that was launched from an AWS Marketplace AMI, first check the `PlatformDetails` field of the AMI. The `PlatformDetails` field indicates which Reserved Instance to purchase. The platform details of the AMI must match the platform of the Reserved Instance, otherwise the Reserved Instance will not be applied to the On-Demand Instance. For information about how to view the platform details of the AMI, see [Obtaining billing information \(p. 169\)](#).

For information about the supported platforms for Windows, see [Choosing a platform](#) in the *Amazon EC2 User Guide for Windows Instances*.

Queuing your purchase

By default, when you purchase a Reserved Instance, it is executed immediately. Alternatively, you can queue your purchases for a future date and time. For example, you can queue a purchase for around the time that an existing Reserved Instance expires. This can help you ensure that you have uninterrupted coverage.

You can queue purchases for regional Reserved Instances, but not zonal Reserved Instances or Reserved Instances from other sellers. You can queue a purchase up to three years in advance. On the scheduled date and time, the purchase is executed using the default payment method. After the payment is successful, the billing benefit is applied.

You can view your queued purchases in the Amazon EC2 console. The status of a queued purchase is **queued**. You can cancel a queued purchase any time before its scheduled time. For details, see [Canceling a queued purchase \(p. 328\)](#).

Buying Standard Reserved Instances

You can buy Standard Reserved Instances in a specific Availability Zone and get a capacity reservation. Alternatively, you can forego the capacity reservation and purchase a regional Standard Reserved Instance.

To buy Standard Reserved Instances using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**, and then choose **Purchase Reserved Instances**.
3. For **Offering Class**, choose **Standard** to display Standard Reserved Instances.
4. To purchase a capacity reservation, choose **Only show offerings that reserve capacity** in the top-right corner of the purchase screen. To purchase a regional Reserved Instance, leave the check box unselected.
5. Select other configurations as needed and choose **Search**.

To purchase a Standard Reserved Instance from the Reserved Instance Marketplace, look for **3rd Party** in the **Seller** column in the search results. The **Term** column displays non-standard terms.

6. Select the Reserved Instances to purchase, enter the quantity, and choose **Add to Cart**.
7. To see a summary of the Reserved Instances that you selected, choose **View Cart**.
8. If **Order On** is **Now**, the purchase is completed immediately. To queue a purchase, choose **Now** and select a date. You can select a different date for each eligible offering in the cart. The purchase is queued until 00:00, in the time zone of your browser, on the selected date.
9. To complete the order, choose **Order**.

If, at the time of placing the order, there are offerings similar to your choice but with a lower price, AWS sells you the offerings at the lower price.

10. The status of your order is listed in the **State** column. When your order is complete, the **State** value changes from `payment-pending` to `active`. When the Reserved Instance is `active`, it is ready to use.

Note

If the status goes to `retired`, AWS may not have received your payment.

To buy a Standard Reserved Instance using the AWS CLI

1. Find available Reserved Instances using the `describe-reserved-instances-offerings` command. Specify `standard` for the `--offering-class` parameter to return only Standard Reserved Instances. You can apply additional parameters to narrow your results. For example, if you want to purchase a regional `t2.large` Reserved Instance with a default tenancy for Linux/UNIX for a 1-year term only:

```
aws ec2 describe-reserved-instances-offerings \
--instance-type t2.large \
--offering-class standard \
--product-description "Linux/UNIX" \
--instance-tenancy default \
--filters Name=duration,Values=31536000 Name=scope,Values=Region
```

To find Reserved Instances on the Reserved Instance Marketplace only, use the `marketplace` filter and do not specify a duration in the request, as the term may be shorter than a 1- or 3-year term.

```
aws ec2 describe-reserved-instances-offerings \
--instance-type t2.large \
--offering-class standard \
--product-description "Linux/UNIX" \
--instance-tenancy default \
--filters Name=marketplace,Values=true
```

When you find a Reserved Instance that meets your needs, take note of the offering ID. For example:

```
"ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2"
```

2. Use the `purchase-reserved-instances-offering` command to buy your Reserved Instance. You must specify the Reserved Instance offering ID you obtained the previous step and you must specify the number of instances for the reservation.

```
aws ec2 purchase-reserved-instances-offering \
--reserved-instances-offering-id bec624df-a8cc-4aad-a72f-4f8abc34caf2 \
--instance-count 1
```

By default, the purchase is completed immediately. Alternatively, to queue the purchase, add the following parameter to the previous call.

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. Use the `describe-reserved-instances` command to get the status of your Reserved Instance.

```
aws ec2 describe-reserved-instances
```

Alternatively, use the following AWS Tools for Windows PowerShell commands:

- [Get-EC2ReservedInstancesOffering](#)
- [New-EC2ReservedInstance](#)
- [Get-EC2ReservedInstance](#)

After the purchase is complete, if you already have a running instance that matches the specifications of the Reserved Instance, the billing benefit is immediately applied. You do not have to restart your instances. If you do not have a suitable running instance, launch an instance and ensure that you match the same criteria that you specified for your Reserved Instance. For more information, see [Using your Reserved Instances \(p. 328\)](#).

For examples of how Reserved Instances are applied to your running instances, see [How Reserved Instances are applied \(p. 313\)](#).

Buying Convertible Reserved Instances

You can buy Convertible Reserved Instances in a specific Availability Zone and get a capacity reservation. Alternatively, you can forego the capacity reservation and purchase a regional Convertible Reserved Instance.

To buy Convertible Reserved Instances using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**, and then choose **Purchase Reserved Instances**.
3. For **Offering Class**, choose **Convertible** to display Convertible Reserved Instances.
4. To purchase a capacity reservation, choose **Only show offerings that reserve capacity** in the top-right corner of the purchase screen. To purchase a regional Reserved Instance, leave the check box unselected.
5. Select other configurations as needed and choose **Search**.
6. Select the Convertible Reserved Instances to purchase, enter the quantity, and choose **Add to Cart**.
7. To see a summary of your selection, choose **View Cart**.
8. If **Order On** is **Now**, the purchase is completed immediately. To queue a purchase, choose **Now** and select a date. You can select a different date for each eligible offering in the cart. The purchase is queued until 00:00, in the time zone of your browser, on the selected date.
9. To complete the order, choose **Order**.

If, at the time of placing the order, there are offerings similar to your choice but with a lower price, AWS sells you the offerings at the lower price.

10. The status of your order is listed in the **State** column. When your order is complete, the **State** value changes from **payment-pending** to **active**. When the Reserved Instance is **active**, it is ready to use.

Note

If the status goes to **retired**, AWS may not have received your payment.

To buy a Convertible Reserved Instance using the AWS CLI

1. Find available Reserved Instances using the [describe-reserved-instances-offerings](#) command. Specify **convertible** for the **--offering-class** parameter to return only Convertible Reserved Instances. You can apply additional parameters to narrow your results; for example, if you want to purchase a regional **t2.large** Reserved Instance with a default tenancy for **Linux/UNIX**:

```
aws ec2 describe-reserved-instances-offerings \
```

```
--instance-type t2.large \
--offering-class convertible \
--product-description "Linux/UNIX" \
--instance-tenancy default \
--filters Name=scope,Values=Region
```

When you find a Reserved Instance that meets your needs, take note of the offering ID. For example:

```
"ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2"
```

2. Use the [purchase-reserved-instances-offering](#) command to buy your Reserved Instance. You must specify the Reserved Instance offering ID you obtained the previous step and you must specify the number of instances for the reservation.

```
aws ec2 purchase-reserved-instances-offering \
--reserved-instances-offering-id bec624df-a8cc-4aad-a72f-4f8abc34caf2 \
--instance-count 1
```

By default, the purchase is completed immediately. Alternatively, to queue the purchase, add the following parameter to the previous call.

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. Use the [describe-reserved-instances](#) command to get the status of your Reserved Instance.

```
aws ec2 describe-reserved-instances
```

Alternatively, use the following AWS Tools for Windows PowerShell commands:

- [Get-EC2ReservedInstancesOffering](#)
- [New-EC2ReservedInstance](#)
- [Get-EC2ReservedInstance](#)

If you already have a running instance that matches the specifications of the Reserved Instance, the billing benefit is immediately applied. You do not have to restart your instances. If you do not have a suitable running instance, launch an instance and ensure that you match the same criteria that you specified for your Reserved Instance. For more information, see [Using your Reserved Instances \(p. 328\)](#).

For examples of how Reserved Instances are applied to your running instances, see [How Reserved Instances are applied \(p. 313\)](#).

Viewing your Reserved Instances

You can view the Reserved Instances you've purchased using the Amazon EC2 console, or a command line tool.

To view your Reserved Instances in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. Your active and retired Reserved Instances are listed. The **State** column displays the state.
4. If you are a seller in the Reserved Instance Marketplace the **My Listings** tab displays the status of a reservation that's listed in the [Reserved Instance Marketplace \(p. 329\)](#). For more information, see [Reserved Instance listing states \(p. 333\)](#).

To view your Reserved Instances using the command line

- [describe-reserved-instances](#) (AWS CLI)
- [Get-EC2ReservedInstance](#) (Tools for Windows PowerShell)

Canceling a queued purchase

You can queue a purchase up to three years in advance. You can cancel a queued purchase any time before its scheduled time.

To cancel a queued purchase

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. Select one or more Reserved Instances.
4. Choose **Actions, Delete Queued Reserved Instances**.
5. When prompted for confirmation, choose **Yes, Delete**.

To cancel a queued purchase using the command line

- [delete-queued-reserved-instances](#) (AWS CLI)
- [Remove-EC2QueuedReservedInstance](#) (Tools for Windows PowerShell)

Renewing a Reserved Instance

You can renew a Reserved Instance before it is scheduled to expire. Renewing a Reserved Instance queues the purchase of a Reserved Instance with the same configuration until the current Reserved Instance expires.

To renew an Reserved Instance using a queued purchase

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. Select one or more Reserved Instances.
4. Choose **Actions, Renew Reserved Instances**.
5. To complete the order, choose **Order**.

Using your Reserved Instances

Reserved Instances are automatically applied to running On-Demand Instances provided that the specifications match. If you have no running On-Demand Instances that match the specifications of your Reserved Instance, the Reserved Instance is unused until you launch an instance with the required specifications.

If you're launching an instance to take advantage of the billing benefit of a Reserved Instance, ensure that you specify the following information during launch:

- Platform: You must choose an Amazon Machine Image (AMI) that matches the platform (product description) of your Reserved Instance. For example, if you specified `Linux/UNIX`, you can launch an instance from an Amazon Linux AMI or an Ubuntu AMI.
- Instance type: Specify the same instance type as your Reserved Instance; for example, `t2.large`.

- **Availability Zone:** If you purchased a Reserved Instance for a specific Availability Zone, you must launch the instance into the same Availability Zone. If you purchased a regional Reserved Instance, you can launch your instance into any Availability Zone.
- **Tenancy:** The tenancy of your instance must match the tenancy of the Reserved Instance; for example, dedicated or shared. For more information, see [Dedicated Instances \(p. 469\)](#).

For more information, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#). For examples of how Reserved Instances are applied to your running instances, see [How Reserved Instances are applied \(p. 313\)](#).

You can use Amazon EC2 Auto Scaling or other AWS services to launch the On-Demand Instances that use your Reserved Instance benefits. For more information, see the [Amazon EC2 Auto Scaling User Guide](#).

Reserved Instance Marketplace

The Reserved Instance Marketplace is a platform that supports the sale of third-party and AWS customers' unused Standard Reserved Instances, which vary in term lengths and pricing options. For example, you may want to sell Reserved Instances after moving instances to a new AWS Region, changing to a new instance type, ending projects before the term expiration, when your business needs change, or if you have unneeded capacity.

If you want to sell your unused Reserved Instances on the Reserved Instance Marketplace, you must meet certain eligibility criteria.

Contents

- [Selling on the Reserved Instance Marketplace \(p. 329\)](#)
- [Buying from the Reserved Instance Marketplace \(p. 335\)](#)

Selling on the Reserved Instance Marketplace

As soon as you list your Reserved Instances in the Reserved Instance Marketplace, they are available for potential buyers to find. All Reserved Instances are grouped according to the duration of the term remaining and the hourly price.

To fulfill a buyer's request, AWS first sells the Reserved Instance with the lowest upfront price in the specified grouping. Then, we sell the Reserved Instance with the next lowest price, until the buyer's entire order is fulfilled. AWS then processes the transactions and transfers ownership of the Reserved Instances to the buyer.

You own your Reserved Instance until it's sold. After the sale, you've given up the capacity reservation and the discounted recurring fees. If you continue to use your instance, AWS charges you the On-Demand price starting from the time that your Reserved Instance was sold.

Contents

- [Restrictions and limitations \(p. 330\)](#)
- [Registering as a seller \(p. 330\)](#)
- [Bank account for disbursement \(p. 330\)](#)
- [Tax information \(p. 331\)](#)
- [Pricing your Reserved Instances \(p. 332\)](#)
- [Listing your Reserved Instances \(p. 332\)](#)
- [Reserved Instance listing states \(p. 333\)](#)
- [Lifecycle of a listing \(p. 333\)](#)
- [After your Reserved Instance is sold \(p. 334\)](#)
- [Getting paid \(p. 334\)](#)

- [Information shared with the buyer \(p. 334\)](#)

Restrictions and limitations

Before you can sell your unused reservations, you must register as a seller in the Reserved Instance Marketplace. For information, see [Registering as a seller \(p. 330\)](#).

The following limitations and restrictions apply when selling Reserved Instances:

- Only Amazon EC2 Standard Reserved Instances can be sold in the Reserved Instance Marketplace. Amazon EC2 Convertible Reserved Instances cannot be sold. Reserved Instances for other AWS services, such as Amazon RDS and Amazon ElastiCache, cannot be sold.
- There must be at least one month remaining in the term of the Standard Reserved Instance.
- You cannot sell a Standard Reserved Instance in a Region that is [disabled by default \(p. 8\)](#).
- The minimum price allowed in the Reserved Instance Marketplace is \$0.00.
- You can sell No Upfront, Partial Upfront, or All Upfront Reserved Instances in the Reserved Instance Marketplace. If there is an upfront payment on a Reserved Instance, it can be sold only after AWS has received the upfront payment and the reservation has been active (you've owned it) for at least 30 days.
- You cannot modify your listing in the Reserved Instance Marketplace directly. However, you can change your listing by first canceling it and then creating another listing with new parameters. For information, see [Pricing your Reserved Instances \(p. 332\)](#). You can also modify your Reserved Instances before listing them. For information, see [Modifying Reserved Instances \(p. 335\)](#).
- AWS charges a service fee of 12 percent of the total upfront price of each Standard Reserved Instance you sell in the Reserved Instance Marketplace. The upfront price is the price the seller is charging for the Standard Reserved Instance.
- When you register as a seller, the bank you specify must have a US address. For more information, see [Additional seller requirements for paid products](#) in the *AWS Marketplace Seller Guide*.
- Amazon Internet Services Private Limited (AISPL) customers can't sell Reserved Instances in the Reserved Instance Marketplace even if they have a US bank account. For more information, see [What are the differences between AWS accounts and AISPL accounts?](#)

Registering as a seller

Note

Only the AWS account root user can register an account as a seller.

To sell in the Reserved Instance Marketplace, you must first register as a seller. During registration, you provide the following information:

- **Bank information**—AWS must have your bank information in order to disburse funds collected when you sell your reservations. The bank you specify must have a US address. For more information, see [Bank account for disbursement \(p. 330\)](#).
- **Tax information**—All sellers are required to complete a tax information interview to determine any necessary tax reporting obligations. For more information, see [Tax information \(p. 331\)](#).

After AWS receives your completed seller registration, you receive an email confirming your registration and informing you that you can get started selling in the Reserved Instance Marketplace.

Bank account for disbursement

AWS must have your bank information in order to disburse funds collected when you sell your Reserved Instance. The bank you specify must have a US address. For more information, see [Additional seller requirements for paid products](#) in the *AWS Marketplace Seller Guide*.

To register a default bank account for disbursements

1. Open the [Reserved Instance Marketplace Seller Registration](#) page and sign in using your AWS credentials.
2. On the **Manage Bank Account** page, provide the following information about the bank through to receive payment:
 - Bank account holder name
 - Routing number
 - Account number
 - Bank account type

Note

If you are using a corporate bank account, you are prompted to send the information about the bank account via fax (1-206-765-3424).

After registration, the bank account provided is set as the default, pending verification with the bank. It can take up to two weeks to verify a new bank account, during which time you can't receive disbursements. For an established account, it usually takes about two days for disbursements to complete.

To change the default bank account for disbursement

1. On the [Reserved Instance Marketplace Seller Registration](#) page, sign in with the account that you used when you registered.
2. On the **Manage Bank Account** page, add a new bank account or modify the default bank account as needed.

Tax information

Your sale of Reserved Instances might be subject to a transaction-based tax, such as sales tax or value-added tax. You should check with your business's tax, legal, finance, or accounting department to determine if transaction-based taxes are applicable. You are responsible for collecting and sending the transaction-based taxes to the appropriate tax authority.

As part of the seller registration process, you must complete a tax interview in the [Seller Registration Portal](#). The interview collects your tax information and populates an IRS form W-9, W-8BEN, or W-8BEN-E, which is used to determine any necessary tax reporting obligations.

The tax information you enter as part of the tax interview might differ depending on whether you operate as an individual or business, and whether you or your business are a US or non-US person or entity. As you fill out the tax interview, keep in mind the following:

- Information provided by AWS, including the information in this topic, does not constitute tax, legal, or other professional advice. To find out how the IRS reporting requirements might affect your business, or if you have other questions, contact your tax, legal, or other professional advisor.
- To fulfill the IRS reporting requirements as efficiently as possible, answer all questions and enter all information requested during the interview.
- Check your answers. Avoid misspellings or entering incorrect tax identification numbers. They can result in an invalidated tax form.

Based on your tax interview responses and IRS reporting thresholds, Amazon might file Form 1099-K. Amazon mails a copy of your Form 1099-K on or before January 31 in the year following the year that

your tax account reaches the threshold levels. For example, if your account reaches the threshold in 2018, your Form 1099-K is mailed on or before January 31, 2019.

For more information about IRS requirements and Form 1099-K, see the [IRS website](#).

Pricing your Reserved Instances

The upfront fee is the only fee that you can specify for the Reserved Instance that you're selling. The upfront fee is the one-time fee that the buyer pays when they purchase a Reserved Instance.

The following are important limits to note:

- **You can sell up to \$50,000 in Reserved Instances.** To increase this limit, complete the [EC2 Reserved Instance Sales form](#).
- **You can sell up to 5,000 Reserved Instances.** To increase this limit, complete the [EC2 Reserved Instance Sales form](#).
- **The minimum price is \$0.** The minimum allowed price in the Reserved Instance Marketplace is \$0.00.

You cannot modify your listing directly. However, you can change your listing by first canceling it and then creating another listing with new parameters.

You can cancel your listing at any time, as long as it's in the active state. You cannot cancel the listing if it's already matched or being processed for a sale. If some of the instances in your listing are matched and you cancel the listing, only the remaining unmatched instances are removed from the listing.

Because the value of Reserved Instances decreases over time, by default, AWS can set prices to decrease in equal increments month over month. However, you can set different upfront prices based on when your reservation sells.

For example, if your Reserved Instance has nine months of its term remaining, you can specify the amount that you would accept if a customer were to purchase that Reserved Instance with nine months remaining. You could set another price with five months remaining, and yet another price with one month remaining.

[Listing your Reserved Instances](#)

As a registered seller, you can choose to sell one or more of your Reserved Instances. You can choose to sell all of them in one listing or in portions. In addition, you can list Reserved Instances with any configuration of instance type, platform, and scope.

The console determines a suggested price. It checks for offerings that match your Reserved Instance and matches the one with the lowest price. Otherwise, it calculates a suggested price based on the cost of the Reserved Instance for its remaining time. If the calculated value is less than \$1.01, the suggested price is \$1.01.

If you cancel your listing and a portion of that listing has already been sold, the cancellation is not effective on the portion that has been sold. Only the unsold portion of the listing is no longer available in the Reserved Instance Marketplace.

To list a Reserved Instance in the Reserved Instance Marketplace using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. Select the Reserved Instances to list, and choose **Sell Reserved Instances**.
4. On the **Configure Your Reserved Instance Listing** page, set the number of instances to sell and the upfront price for the remaining term in the relevant columns. See how the value of your reservation changes over the remainder of the term by selecting the arrow next to the **Months Remaining** column.

5. If you are an advanced user and you want to customize the pricing, you can enter different values for the subsequent months. To return to the default linear price drop, choose **Reset**.
6. Choose **Continue** when you are finished configuring your listing.
7. Confirm the details of your listing, on the **Confirm Your Reserved Instance Listing** page and if you're satisfied, choose **List Reserved Instance**.

To view your listings in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances**.
3. Select the Reserved Instance that you've listed and choose **My Listings**.

To manage Reserved Instances in the Reserved Instance Marketplace using the AWS CLI

1. Get a list of your Reserved Instances by using the [describe-reserved-instances](#) command.
2. Note the ID of the Reserved Instance you want to list and call [create-reserved-instances-listing](#). You must specify the ID of the Reserved Instance, the number of instances, and the pricing schedule.
3. To view your listing, use the [describe-reserved-instances-listings](#) command.
4. To cancel your listing, use the [cancel-reserved-instances-listings](#) command.

Reserved Instance listing states

Listing State on the **My Listings** tab of the Reserved Instances page displays the current status of your listings:

The information displayed by **Listing State** is about the status of your listing in the Reserved Instance Marketplace. It is different from the status information that is displayed by the **State** column in the **Reserved Instances** page. This **Status** information is about your reservation.

- **active**—The listing is available for purchase.
- **canceled**—The listing is canceled and isn't available for purchase in the Reserved Instance Marketplace.
- **closed**—The Reserved Instance is not listed. A Reserved Instance might be closed because the sale of the listing was completed.

Lifecycle of a listing

When all the instances in your listing are matched and sold, the **My Listings** tab shows that the **Total instance count** matches the count listed under **Sold**. Also, there are no **Available** instances left for your listing, and its **Status** is **closed**.

When only a portion of your listing is sold, AWS retires the Reserved Instances in the listing and creates the number of Reserved Instances equal to the Reserved Instances remaining in the count. So, the listing ID and the listing that it represents, which now has fewer reservations for sale, is still active.

Any future sales of Reserved Instances in this listing are processed this way. When all the Reserved Instances in the listing are sold, AWS marks the listing as **closed**.

For example, you create a listing *Reserved Instances listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample* with a listing count of 5.

The **My Listings** tab in the **Reserved Instance** console page displays the listing this way:

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- Total reservation count = 5
- Sold = 0
- Available = 5
- Status = active

A buyer purchases two of the reservations, which leaves a count of three reservations still available for sale. Because of this partial sale, AWS creates a new reservation with a count of three to represent the remaining reservations that are still for sale.

This is how your listing looks in the **My Listings** tab:

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- Total reservation count = 5
- Sold = 2
- Available = 3
- Status = active

If you cancel your listing and a portion of that listing has already sold, the cancellation is not effective on the portion that has been sold. Only the unsold portion of the listing is no longer available in the Reserved Instance Marketplace.

After your Reserved Instance is sold

When your Reserved Instance is sold, AWS sends you an email notification. Each day that there is any kind of activity, you receive one email notification capturing all the activities of the day. For example, you create or sell a listing, or AWS sends funds to your account.

To track the status of a Reserved Instance listing in the console, choose **Reserved Instance, My Listings**. The **My Listings** tab contains the **Listing State** value. It also contains information about the term, listing price, and a breakdown of how many instances in the listing are available, pending, sold, and canceled. You can also use the [describe-reserved-instances-listings](#) command with the appropriate filter to obtain information about your listings.

Getting paid

As soon as AWS receives funds from the buyer, a message is sent to the registered owner account email for the sold Reserved Instance.

AWS sends an Automated Clearing House (ACH) wire transfer to your specified bank account. Typically, this transfer occurs between one to three days after your Reserved Instance has been sold. Disbursements take place once a day. You will receive an email with a disbursement report after the funds are released. Keep in mind that you can't receive disbursements until AWS receives verification from your bank. This can take up to two weeks.

The Reserved Instance that you sold continues to appear when you describe your Reserved Instances.

You receive a cash disbursement for your Reserved Instances through a wire transfer directly into your bank account. AWS charges a service fee of 12 percent of the total upfront price of each Reserved Instance you sell in the Reserved Instance Marketplace.

Information shared with the buyer

When you sell in the Reserved Instance Marketplace, AWS shares your company's legal name on the buyer's statement in accordance with US regulations. In addition, if the buyer calls AWS Support because the buyer needs to contact you for an invoice or for some other tax-related reason, AWS may need to provide the buyer with your email address so that the buyer can contact you directly.

For similar reasons, the buyer's ZIP code and country information are provided to the seller in the disbursement report. As a seller, you might need this information to accompany any necessary transaction taxes that you remit to the government (such as sales tax and value-added tax).

AWS cannot offer tax advice, but if your tax specialist determines that you need specific additional information, [contact AWS Support](#).

Buying from the Reserved Instance Marketplace

You can purchase Reserved Instances from third-party sellers who own Reserved Instances that they no longer need from the Reserved Instance Marketplace. You can do this using the Amazon EC2 console or a command line tool. The process is similar to purchasing Reserved Instances from AWS. For more information, see [Buying Reserved Instances \(p. 322\)](#).

There are a few differences between Reserved Instances purchased in the Reserved Instance Marketplace and Reserved Instances purchased directly from AWS:

- **Term**—Reserved Instances that you purchase from third-party sellers have less than a full standard term remaining. Full standard terms from AWS run for one year or three years.
- **Upfront price**—Third-party Reserved Instances can be sold at different upfront prices. The usage or recurring fees remain the same as the fees set when the Reserved Instances were originally purchased from AWS.
- **Types of Reserved Instances**—Only Amazon EC2 Standard Reserved Instances can be purchased from the Reserved Instance Marketplace. Convertible Reserved Instances, Amazon RDS and Amazon ElastiCache Reserved Instances are not available for purchase on the Reserved Instance Marketplace.

Basic information about you is shared with the seller, for example, your ZIP code and country information.

This information enables sellers to calculate any necessary transaction taxes that they have to remit to the government (such as sales tax or value-added tax) and is provided as a disbursement report. In rare circumstances, AWS might have to provide the seller with your email address, so that they can contact you regarding questions related to the sale (for example, tax questions).

For similar reasons, AWS shares the legal entity name of the seller on the buyer's purchase invoice. If you need additional information about the seller for tax or related reasons, contact [AWS Support](#).

Modifying Reserved Instances

When your needs change, you can modify your Standard or Convertible Reserved Instances and continue to benefit from the billing benefit. You can modify attributes such as the Availability Zone, instance size (within the same instance family), and scope of your Reserved Instance.

Note

You can also exchange a Convertible Reserved Instance for another Convertible Reserved Instance with a different configuration. For more information, see [Exchanging Convertible Reserved Instances \(p. 342\)](#).

You can modify all or a subset of your Reserved Instances. You can separate your original Reserved Instances into two or more new Reserved Instances. For example, if you have a reservation for 10 instances in us-east-1a and decide to move 5 instances to us-east-1b, the modification request results in two new reservations: one for 5 instances in us-east-1a and the other for 5 instances in us-east-1b.

You can also *merge* two or more Reserved Instances into a single Reserved Instance. For example, if you have four t2.small Reserved Instances of one instance each, you can merge them to create one t2.large Reserved Instance. For more information, see [Support for modifying instance sizes \(p. 337\)](#).

After modification, the benefit of the Reserved Instances is applied only to instances that match the new parameters. For example, if you change the Availability Zone of a reservation, the capacity reservation and pricing benefits are automatically applied to instance usage in the new Availability Zone. Instances that no longer match the new parameters are charged at the On-Demand rate, unless your account has other applicable reservations.

If your modification request succeeds:

- The modified reservation becomes effective immediately and the pricing benefit is applied to the new instances beginning at the hour of the modification request. For example, if you successfully modify your reservations at 9:15PM, the pricing benefit transfers to your new instance at 9:00PM. You can get the effective date of the modified Reserved Instances by using the [describe-reserved-instances](#) command.
- The original reservation is retired. Its end date is the start date of the new reservation, and the end date of the new reservation is the same as the end date of the original Reserved Instance. If you modify a three-year reservation that had 16 months left in its term, the resulting modified reservation is a 16-month reservation with the same end date as the original one.
- The modified reservation lists a \$0 fixed price and not the fixed price of the original reservation.
- The fixed price of the modified reservation does not affect the discount pricing tier calculations applied to your account, which are based on the fixed price of the original reservation.

If your modification request fails, your Reserved Instances maintain their original configuration, and are immediately available for another modification request.

There is no fee for modification, and you do not receive any new bills or invoices.

You can modify your reservations as frequently as you like, but you cannot change or cancel a pending modification request after you submit it. After the modification has completed successfully, you can submit another modification request to roll back any changes you made, if needed.

Contents

- [Requirements and restrictions for modification \(p. 336\)](#)
- [Support for modifying instance sizes \(p. 337\)](#)
- [Submitting modification requests \(p. 341\)](#)
- [Troubleshooting modification requests \(p. 342\)](#)

Requirements and restrictions for modification

You can modify these attributes as follows.

Modifiable attribute	Supported platforms	Limitations
Change Availability Zones within the same Region	Linux and Windows	-
Change the scope from Availability Zone to Region and vice versa	Linux and Windows	If you change the scope from Availability Zone to Region, you lose the capacity reservation benefit. If you change the scope from Region to Availability Zone, you lose Availability Zone flexibility and instance size flexibility (if applicable). For more

Modifiable attribute	Supported platforms	Limitations
		information, see How Reserved Instances are applied (p. 313) .
Change the instance size within the same instance family	Linux/UNIX only Instance size flexibility is not available for Reserved Instances on the other platforms, which include Linux with SQL Server Standard, Linux with SQL Server Web, Linux with SQL Server Enterprise, Red Hat Enterprise Linux, SUSE Linux, Windows, Windows with SQL Standard, Windows with SQL Server Enterprise, and Windows with SQL Server Web.	The reservation must use default tenancy. Some instance families are not supported, because there are no other sizes available. For more information, see Support for modifying instance sizes (p. 337) .
Change the network from EC2-Classic to Amazon VPC and vice versa	Linux and Windows	The network platform must be available in your AWS account. If you created your AWS account after 2013-12-04, it does not support EC2-Classic.

Requirements

Amazon EC2 processes your modification request if there is sufficient capacity for your target configuration (if applicable), and if the following conditions are met:

- The Reserved Instance cannot be modified before or at the same time that you purchase it
- The Reserved Instance must be active
- There cannot be a pending modification request
- The Reserved Instance is not listed in the Reserved Instance Marketplace
- There must be a match between the instance size footprint of the active reservation and the target configuration. For more information, see [Support for modifying instance sizes \(p. 337\)](#).
- The input Reserved Instances are all Standard Reserved Instances or all Convertible Reserved Instances, not some of each type
- The input Reserved Instances must expire within the same hour, if they are Standard Reserved Instances
- The Reserved Instance is not a G4 instance.

Support for modifying instance sizes

You can modify the instance size of a Reserved Instance if the following requirements are met.

Requirements

- The platform is Linux/UNIX.
- You must select another instance size in the same instance family. For example, you cannot modify an Reserved Instance from t2 to t3, whether you use the same size or a different size.

You cannot modify the instance size of Reserved Instances for the following instances, because each of these instance families has only one size:

- `cc2.8xlarge`
- `cr1.8xlarge`
- `hs1.8xlarge`
- `t1.micro`
- The original and modified Reserved Instance must have the same instance size footprint.

Contents

- [Instance size footprint \(p. 338\)](#)
- [Normalization factors for bare metal instances \(p. 340\)](#)

Instance size footprint

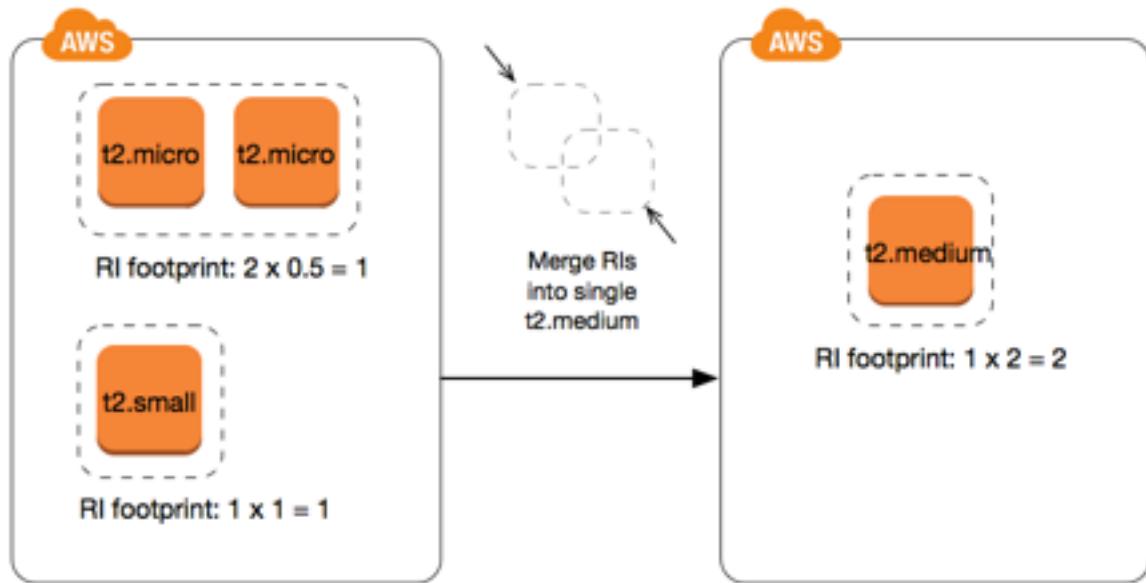
Each Reserved Instance has an *instance size footprint*, which is determined by the normalization factor of the instance size and the number of instances in the reservation. When you modify the instance sizes in an Reserved Instance, the footprint of the target configuration must match that of the original configuration, otherwise the modification request is not processed.

To calculate the instance size footprint of a Reserved Instance, multiply the number of instances by the normalization factor. In the Amazon EC2 console, the normalization factor is measured in units. The following table describes the normalization factor for the instance sizes in an instance family. For example, `t2.medium` has a normalization factor of 2, so a reservation for four `t2.medium` instances has a footprint of 8 units.

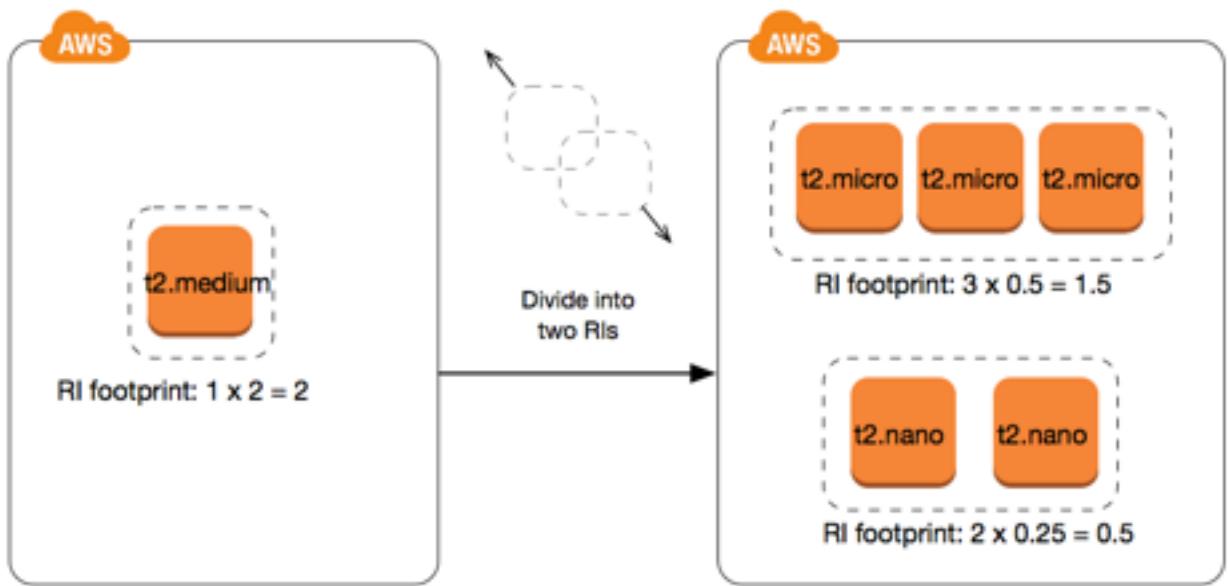
Instance size	Normalization factor
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
4xlarge	32
8xlarge	64
9xlarge	72
10xlarge	80
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256

You can allocate your reservations into different instance sizes across the same instance family as long as the instance size footprint of your reservation remains the same. For example, you can divide a reservation for one `t2.large` (1 @ 4 units) instance into four `t2.small` (4 @ 1 unit) instances. Similarly, you can combine a reservation for four `t2.small` instances into one `t2.large` instance. However, you cannot change your reservation for two `t2.small` instances into one `t2.large` instance because the footprint of the modified reservation (4 units) is larger than the footprint of the existing reservation (2 units).

In the following example, you have a reservation with two `t2.micro` instances (1 unit) and a reservation with one `t2.small` instance (1 unit). If you merge both of these reservations to a single reservation with one `t2.medium` instance (2 units), the footprint of the modified reservation equals the footprint of the combined reservations.



You can also modify a reservation to divide it into two or more reservations. In the following example, you have a reservation with a `t2.medium` instance (2 units). You can divide the reservation into two reservations, one with two `t2.nano` instances (.5 units) and the other with three `t2.micro` instances (1.5 units).



Normalization factors for bare metal instances

You can modify a reservation with `metal` instances using other sizes within the same instance family. Similarly, you can modify a reservation with instances other than bare metal instances using the `metal` size within the same instance family. Generally, a bare metal instance is the same size as the largest available instance size within the same instance family. For example, an `i3.metal` instance is the same size as an `i3.16xlarge` instance, so they have the same normalization factor.

The following table describes the normalization factor for the bare metal instance sizes in the instance families that have bare metal instances. The normalization factor for `metal` instances depends on the instance family, unlike the other instance sizes.

Bare metal instance size	Normalization factor
<code>c5.metal</code>	192
<code>i3.metal</code>	128
<code>r5.metal</code>	192
<code>r5d.metal</code>	192
<code>z1d.metal</code>	96
<code>m5.metal</code>	192
<code>m5d.metal</code>	192

For example, an `i3.metal` instance has a normalization factor of 128. If you purchase an `i3.metal` default tenancy Amazon Linux/Unix Reserved Instance, you can divide the reservation as follows:

- An `i3.16xlarge` is the same size as an `i3.metal` instance, so its normalization factor is 128 (128/1). The reservation for one `i3.metal` instance can be modified into one `i3.16xlarge` instance.
- An `i3.8xlarge` is half the size of an `i3.metal` instance, so its normalization factor is 64 (128/2). The reservation for one `i3.metal` instance can be divided into two `i3.8xlarge` instances.

- An *i3.4xlarge* is a quarter the size of an *i3.metal* instance, so its normalization factor is 32 (128/4). The reservation for one *i3.metal* instance can be divided into four *i3.4xlarge* instances.

Submitting modification requests

Before you modify your Reserved Instances, ensure that you have read the applicable [restrictions \(p. 336\)](#). Before you modify the instance size, calculate the total [instance size footprint \(p. 337\)](#) of the reservations that you want to modify and ensure that it matches the total instance size footprint of your target configurations.

To modify your Reserved Instances using the AWS Management Console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Reserved Instances** page, select one or more Reserved Instances to modify, and choose **Actions, Modify Reserved Instances**.

Note

If your Reserved Instances are not in the active state or cannot be modified, **Modify Reserved Instances** is disabled.

3. The first entry in the modification table displays attributes of selected Reserved Instances, and at least one target configuration beneath it. The **Units** column displays the total instance size footprint. Choose **Add** for each new configuration to add. Modify the attributes as needed for each configuration, and then choose **Continue**:
 - **Scope:** Choose whether the configuration applies to an Availability Zone or to the whole Region.
 - **Availability Zone:** Choose the required Availability Zone. Not applicable for regional Reserved Instances.
 - **Instance Type:** Select the required instance type. The combined configurations must equal the instance size footprint of your original configurations.
 - **Count:** Specify the number of instances. To split the Reserved Instances into multiple configurations, reduce the count, choose **Add**, and specify a count for the additional configuration. For example, if you have a single configuration with a count of 10, you can change its count to 6 and add a configuration with a count of 4. This process retires the original Reserved Instance after the new Reserved Instances are activated.
4. To confirm your modification choices when you finish specifying your target configurations, choose **Submit Modifications**.
5. You can determine the status of your modification request by looking at the **State** column in the Reserved Instances screen. The following are the possible states.
 - **active (pending modification)** — Transition state for original Reserved Instances
 - **retired (pending modification)** — Transition state for original Reserved Instances while new Reserved Instances are being created
 - **retired** — Reserved Instances successfully modified and replaced
 - **active** — One of the following:
 - New Reserved Instances created from a successful modification request
 - Original Reserved Instances after a failed modification request

To modify your Reserved Instances using the command line

1. To modify your Reserved Instances, you can use one of the following commands:
 - [modify-reserved-instances](#) (AWS CLI)
 - [Edit-EC2ReservedInstance](#) (AWS Tools for Windows PowerShell)

2. To get the status of your modification request (processing, fulfilled, or failed), use one of the following commands:
 - [describe-reserved-instances-modifications](#) (AWS CLI)
 - [Get-EC2ReservedInstancesModification](#) (AWS Tools for Windows PowerShell)

Troubleshooting modification requests

If the target configuration settings that you requested were unique, you receive a message that your request is being processed. At this point, Amazon EC2 has only determined that the parameters of your modification request are valid. Your modification request can still fail during processing due to unavailable capacity.

In some situations, you might get a message indicating incomplete or failed modification requests instead of a confirmation. Use the information in such messages as a starting point for resubmitting another modification request. Ensure that you have read the applicable [restrictions \(p. 336\)](#) before submitting the request.

Not all selected Reserved Instances can be processed for modification

Amazon EC2 identifies and lists the Reserved Instances that cannot be modified. If you receive a message like this, go to the **Reserved Instances** page in the Amazon EC2 console and check the information for the Reserved Instances.

Error in processing your modification request

You submitted one or more Reserved Instances for modification and none of your requests can be processed. Depending on the number of reservations you are modifying, you can get different versions of the message.

Amazon EC2 displays the reasons why your request cannot be processed. For example, you might have specified the same target configuration—a combination of Availability Zone and platform—for one or more subsets of the Reserved Instances you are modifying. Try submitting the modification requests again, but ensure that the instance details of the reservations match, and that the target configurations for all subsets being modified are unique.

Exchanging Convertible Reserved Instances

You can exchange one or more Convertible Reserved Instances for another Convertible Reserved Instance with a different configuration, including instance family, operating system, and tenancy. There are no limits to how many times you perform an exchange, as long as the target Convertible Reserved Instance is of an equal or higher value than the Convertible Reserved Instances that you are exchanging.

When you exchange your Convertible Reserved Instance, the number of instances for your current reservation is exchanged for a number of instances that cover the equal or higher value of the configuration of the target Convertible Reserved Instance. Amazon EC2 calculates the number of Reserved Instances that you can receive as a result of the exchange.

Contents

- [Requirements for exchanging Convertible Reserved Instances \(p. 343\)](#)
- [Calculating Convertible Reserved Instances exchanges \(p. 344\)](#)
- [Merging Convertible Reserved Instances \(p. 344\)](#)
- [Exchanging a portion of a Convertible Reserved Instance \(p. 345\)](#)
- [Submitting exchange requests \(p. 346\)](#)

Requirements for exchanging Convertible Reserved Instances

If the following conditions are met, Amazon EC2 processes your exchange request. Your Convertible Reserved Instance must be:

- Active
- Not pending a previous exchange request

The following rules apply:

- Convertible Reserved Instances can only be exchanged for other Convertible Reserved Instances currently offered by AWS.
- Convertible Reserved Instances are associated with a specific Region, which is fixed for the duration of the reservation's term. You cannot exchange a Convertible Reserved Instance for a Convertible Reserved Instance in a different Region.
- You can exchange one or more Convertible Reserved Instances at a time for one Convertible Reserved Instance only.
- To exchange a portion of a Convertible Reserved Instance, you can modify it into two or more reservations, and then exchange one or more of the reservations for a new Convertible Reserved Instance. For more information, see [Exchanging a portion of a Convertible Reserved Instance \(p. 345\)](#). For more information about modifying your Reserved Instances, see [Modifying Reserved Instances \(p. 335\)](#).
- All Upfront Convertible Reserved Instances can be exchanged for Partial Upfront Convertible Reserved Instances, and vice versa.

Note

If the total upfront payment required for the exchange (true-up cost) is less than \$0.00, AWS automatically gives you a quantity of instances in the Convertible Reserved Instance that ensures that true-up cost is \$0.00 or more.

Note

If the total value (upfront price + hourly price * number of remaining hours) of the new Convertible Reserved Instance is less than the total value of the exchanged Convertible Reserved Instance, AWS automatically gives you a quantity of instances in the Convertible Reserved Instance that ensures that the total value is the same or higher than that of the exchanged Convertible Reserved Instance.

- To benefit from better pricing, you can exchange a No Upfront Convertible Reserved Instance for an All Upfront or Partial Upfront Convertible Reserved Instance.
- You cannot exchange All Upfront and Partial Upfront Convertible Reserved Instances for No Upfront Convertible Reserved Instances.
- You can exchange a No Upfront Convertible Reserved Instance for another No Upfront Convertible Reserved Instance only if the new Convertible Reserved Instance's hourly price is the same or higher than the exchanged Convertible Reserved Instance's hourly price.

Note

If the total value (hourly price * number of remaining hours) of the new Convertible Reserved Instance is less than the total value of the exchanged Convertible Reserved Instance, AWS automatically gives you a quantity of instances in the Convertible Reserved Instance that ensures that the total value is the same or higher than that of the exchanged Convertible Reserved Instance.

- If you exchange multiple Convertible Reserved Instances that have different expiration dates, the expiration date for the new Convertible Reserved Instance is the date that's furthest in the future.
- If you exchange a single Convertible Reserved Instance, it must have the same term (1-year or 3-years) as the new Convertible Reserved Instance. If you merge multiple Convertible Reserved Instances with different term lengths, the new Convertible Reserved Instance has a 3-year term. For more information, see [Merging Convertible Reserved Instances \(p. 344\)](#).

- After you exchange a Convertible Reserved Instance, the original reservation is retired. Its end date is the start date of the new reservation, and the end date of the new reservation is the same as the end date of the original Convertible Reserved Instance. For example, if you modify a three-year reservation that had 16 months left in its term, the resulting modified reservation is a 16-month reservation with the same end date as the original one.

Calculating Convertible Reserved Instances exchanges

Exchanging Convertible Reserved Instances is free. However, you may be required to pay a true-up cost, which is a prorated upfront cost of the difference between the Convertible Reserved Instances that you had and the Convertible Reserved Instances that you receive from the exchange.

Each Convertible Reserved Instance has a list value. This list value is compared to the list value of the Convertible Reserved Instances that you want in order to determine how many instance reservations you can receive from the exchange.

For example: You have 1 x \$35-list value Convertible Reserved Instance that you want to exchange for a new instance type with a list value of \$10.

\$35/\$10 = 3.5

You can exchange your Convertible Reserved Instance for three \$10 Convertible Reserved Instances. It's not possible to purchase half reservations; therefore you must purchase an additional Convertible Reserved Instance to cover the remainder:

3.5 = 3 whole Convertible Reserved Instances + 1 additional Convertible Reserved Instance.

The fourth Convertible Reserved Instance has the same end date as the other three. If you are exchanging Partial or All Upfront Convertible Reserved Instances, you pay the true-up cost for the fourth reservation. If the remaining upfront cost of your Convertible Reserved Instances is \$500, and the target reservation would normally cost \$600 on a prorated basis, you are charged \$100.

\$600 prorated upfront cost of new reservations - \$500 remaining upfront cost of original reservations = \$100 difference.

Merging Convertible Reserved Instances

If you merge two or more Convertible Reserved Instances, the term of the new Convertible Reserved Instance must be the same as the original Convertible Reserved Instances, or the highest of the original Convertible Reserved Instances. The expiration date for the new Convertible Reserved Instance is the expiration date that's furthest in the future.

For example, you have the following Convertible Reserved Instances in your account:

Reserved Instance ID	Term	Expiration date
aaaa1111	1-year	2018-12-31
bbbb2222	1-year	2018-07-31
cccc3333	3-year	2018-06-30
dddd4444	3-year	2019-12-31

- You can merge aaaa1111 and bbbb2222 and exchange them for a 1-year Convertible Reserved Instance. You cannot exchange them for a 3-year Convertible Reserved Instance. The expiration date of the new Convertible Reserved Instance is 2018-12-31.
- You can merge bbbb2222 and cccc3333 and exchange them for a 3-year Convertible Reserved Instance. You cannot exchange them for a 1-year Convertible Reserved Instance. The expiration date of the new Convertible Reserved Instance is 2018-07-31.
- You can merge cccc3333 and dddd4444 and exchange them for a 3-year Convertible Reserved Instance. You cannot exchange them for a 1-year Convertible Reserved Instance. The expiration date of the new Convertible Reserved Instance is 2019-12-31.

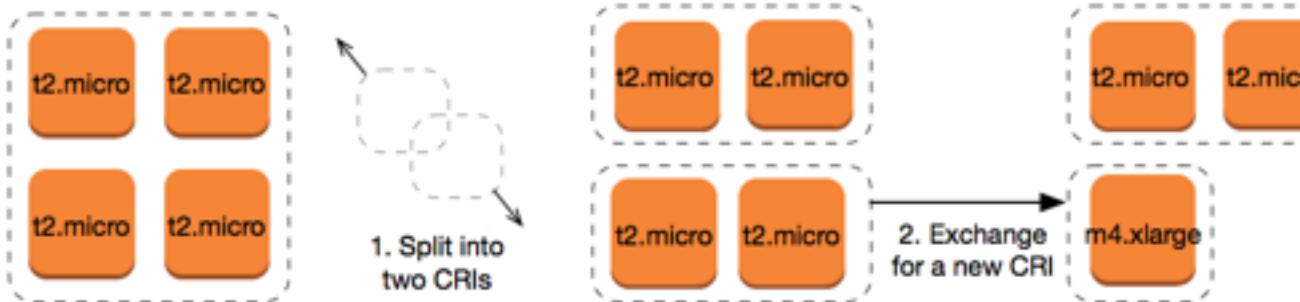
Exchanging a portion of a Convertible Reserved Instance

You can use the modification process to split your Convertible Reserved Instance into smaller reservations, and then exchange one or more of the new reservations for a new Convertible Reserved Instance. The following examples demonstrate how you can do this.

Example Example: Convertible Reserved Instance with multiple instances

In this example, you have a t2.micro Convertible Reserved Instance with four instances in the reservation. To exchange two t2.micro instances for an m4.xlarge instance:

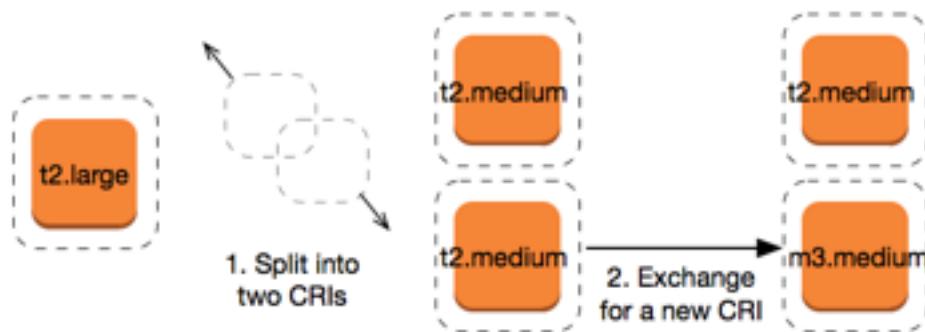
1. Modify the t2.micro Convertible Reserved Instance by splitting it into two t2.micro Convertible Reserved Instances with two instances each.
2. Exchange one of the new t2.micro Convertible Reserved Instances for an m4.xlarge Convertible Reserved Instance.



Example Example: Convertible Reserved Instance with a single instance

In this example, you have a t2.large Convertible Reserved Instance. To change it to a smaller t2.medium instance and a m3.medium instance:

1. Modify the t2.large Convertible Reserved Instance by splitting it into two t2.medium Convertible Reserved Instances. A single t2.large instance has the same instance size footprint as two t2.medium instances.
2. Exchange one of the new t2.medium Convertible Reserved Instances for an m3.medium Convertible Reserved Instance.



For more information, see [Support for modifying instance sizes \(p. 337\)](#) and [Submitting exchange requests \(p. 346\)](#).

Submitting exchange requests

You can exchange your Convertible Reserved Instances using the Amazon EC2 console or a command line tool.

Exchanging a Convertible Reserved Instance using the console

You can search for Convertible Reserved Instances offerings and select your new configuration from the choices provided.

To exchange Convertible Reserved Instances using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Reserved Instances**, select the Convertible Reserved Instances to exchange, and choose **Actions, Exchange Reserved Instance**.
3. Select the attributes of the desired configuration using the drop-down menus, and choose **Find Offering**.
4. Select a new Convertible Reserved Instance. The **Instance Count** column displays the number of Reserved Instances that you receive for the exchange. When you have selected a Convertible Reserved Instance that meets your needs, choose **Exchange**.

The Reserved Instances that were exchanged are retired, and the new Reserved Instances are displayed in the Amazon EC2 console. This process can take a few minutes to propagate.

Exchanging a Convertible Reserved Instance using the command line interface

To exchange a Convertible Reserved Instance, first find a target Convertible Reserved Instance that meets your needs:

- [describe-reserved-instances-offerings](#) (AWS CLI)
- [Get-EC2ReservedInstancesOffering](#) (Tools for Windows PowerShell)

Get a quote for the exchange, which includes the number of Reserved Instances you get from the exchange, and the true-up cost for the exchange:

- [get-reserved-instances-exchange-quote](#) (AWS CLI)
- [GetEC2ReservedInstancesExchangeQuote](#) (Tools for Windows PowerShell)

Finally, perform the exchange:

- [accept-reserved-instances-exchange-quote \(AWS CLI\)](#)
- [Confirm-EC2ReservedInstancesExchangeQuote \(Tools for Windows PowerShell\)](#)

Scheduled Reserved Instances

Scheduled Reserved Instances (Scheduled Instances) enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified start time and duration, for a one-year term. You reserve the capacity in advance, so that you know it is available when you need it. You pay for the time that the instances are scheduled, even if you do not use them.

Scheduled Instances are a good choice for workloads that do not run continuously, but do run on a regular schedule. For example, you can use Scheduled Instances for an application that runs during business hours or for batch processing that runs at the end of the week.

If you require a capacity reservation on a continuous basis, Reserved Instances might meet your needs and decrease costs. For more information, see [Reserved Instances \(p. 308\)](#). If you are flexible about when your instances run, Spot Instances might meet your needs and decrease costs. For more information, see [Spot Instances \(p. 351\)](#).

Contents

- [How Scheduled Instances work \(p. 347\)](#)
- [Service-linked roles for Scheduled Instances \(p. 348\)](#)
- [Purchasing a Scheduled Instance \(p. 348\)](#)
- [Launching a Scheduled Instance \(p. 349\)](#)
- [Scheduled Instance limits \(p. 350\)](#)

How Scheduled Instances work

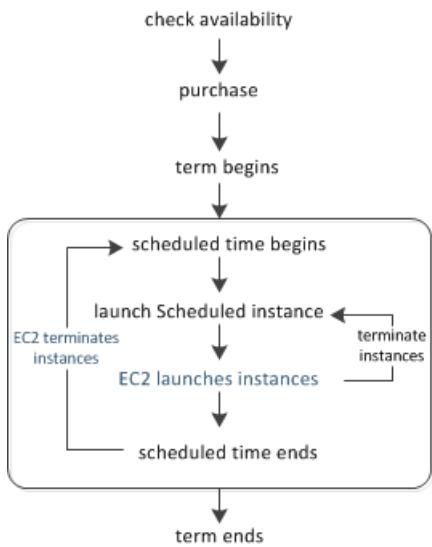
Amazon EC2 sets aside pools of EC2 instances in each Availability Zone for use as Scheduled Instances. Each pool supports a specific combination of instance type, operating system, and network.

To get started, you must search for an available schedule. You can search across multiple pools or a single pool. After you locate a suitable schedule, purchase it.

You must launch your Scheduled Instances during their scheduled time periods, using a launch configuration that matches the following attributes of the schedule that you purchased: instance type, Availability Zone, network, and platform. When you do so, Amazon EC2 launches EC2 instances on your behalf, based on the specified launch specification. Amazon EC2 must ensure that the EC2 instances have terminated by the end of the current scheduled time period so that the capacity is available for any other Scheduled Instances it is reserved for. Therefore, Amazon EC2 terminates the EC2 instances three minutes before the end of the current scheduled time period.

You can't stop or reboot Scheduled Instances, but you can terminate them manually as needed. If you terminate a Scheduled Instance before its current scheduled time period ends, you can launch it again after a few minutes. Otherwise, you must wait until the next scheduled time period.

The following diagram illustrates the lifecycle of a Scheduled Instance.



Service-linked roles for Scheduled Instances

Amazon EC2 creates a service-linked role when you purchase a Scheduled Instance. A service-linked role includes all the permissions that Amazon EC2 requires to call other AWS services on your behalf. For more information, see [Using Service-Linked Roles](#) in the *IAM User Guide*.

Amazon EC2 uses the service-linked role named **AWSServiceRoleForEC2ScheduledInstances** to complete the following actions:

- **ec2:TerminateInstances** - Terminate Scheduled Instances after their schedules complete
- **ec2:CreateTags** - Add system tags to Scheduled Instances

If you purchased Scheduled Instances before October 2017, when Amazon EC2 began supporting this service-linked role, Amazon EC2 created the **AWSServiceRoleForEC2ScheduledInstances** role in your AWS account. For more information, see [A New Role Appeared in My Account](#) in the *IAM User Guide*.

If you no longer need to use Scheduled Instances, we recommend that you delete the **AWSServiceRoleForEC2ScheduledInstances** role. After this role is deleted from your account, Amazon EC2 will create the role again if you purchase Scheduled Instances.

Purchasing a Scheduled Instance

To purchase a Scheduled Instance, you can use the Scheduled Reserved Instances Reservation Wizard.

Warning

After you purchase a Scheduled Instance, you can't cancel, modify, or resell your purchase.

To purchase a Scheduled Instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **INSTANCES**, choose **Scheduled Instances**. If the currently selected Region does not support Scheduled Instances, the page is unavailable. [Learn more \(p. 350\)](#)
3. Choose **Purchase Scheduled Instances**.
4. On the **Find available schedules** page, do the following:
 - a. Under **Create a schedule**, select the starting date from **Starting on**, the schedule recurrence (daily, weekly, or monthly) from **Recurring**, and the minimum duration from **for duration**. Note

that the console ensures that you specify a value for the minimum duration that meets the minimum required utilization for your Scheduled Instance (1,200 hours per year).

Create a schedule

Starting on for duration 4 +/- 2 hours

Recurring

- b. Under **Instance details**, select the operating system and network from **Platform**. To narrow the results, select one or more instance types from **Instance type** or one or more Availability Zones from **Availability Zone**.

Instance details

Platform Instance type

Availability Zone

- c. Choose **Find schedules**.
- d. Under **Available schedules**, select one or more schedules. For each schedule that you select, set the quantity of instances and choose **Add to Cart**.
- e. Your cart is displayed at the bottom of the page. When you are finished adding and removing schedules from your cart, choose **Review and purchase**.
5. On the **Review and purchase** page, verify your selections and edit them as needed. When you are finished, choose **Purchase**.

To purchase a Scheduled Instance (AWS CLI)

Use the [describe-scheduled-instance-availability](#) command to list the available schedules that meet your needs, and then use the [purchase-scheduled-instances](#) command to complete the purchase.

Launching a Scheduled Instance

After you purchase a Scheduled Instance, it is available for you to launch during its scheduled time periods.

To launch a Scheduled Instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **INSTANCES**, choose **Scheduled Instances**. If the currently selected Region does not support Scheduled Instances, the page is unavailable. [Learn more \(p. 350\)](#)
3. Select the Scheduled Instance and choose **Launch Scheduled Instances**.
4. On the **Configure** page, complete the launch specification for your Scheduled Instances and choose **Review**.

Important

The launch specification must match the instance type, Availability Zone, network, and platform of the schedule that you purchased.

5. On the **Review** page, verify the launch configuration and modify it as needed. When you are finished, choose **Launch**.

To launch a Scheduled Instance (AWS CLI)

Use the [describe-scheduled-instances](#) command to list your Scheduled Instances, and then use the [run-scheduled-instances](#) command to launch each Scheduled Instance during its scheduled time periods.

Scheduled Instance limits

Scheduled Instances are subject to the following limits:

- The following are the only supported instance types: C3, C4, M4, and R3.
- The required term is 365 days (one year).
- The minimum required utilization is 1,200 hours per year.
- You can purchase a Scheduled Instance up to three months in advance.
- They are available in the following Regions: US East (N. Virginia), US West (Oregon), and Europe (Ireland).

Spot Instances

A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance is called a Spot price. The Spot price of each instance type in each Availability Zone is set by Amazon EC2, and is adjusted gradually based on the long-term supply of and demand for Spot Instances. Your Spot Instance runs whenever capacity is available and the maximum price per hour for your request exceeds the Spot price.

Spot Instances are a cost-effective choice if you can be flexible about when your applications run and if your applications can be interrupted. For example, Spot Instances are well-suited for data analysis, batch jobs, background processing, and optional tasks. For more information, see [Amazon EC2 Spot Instances](#).

Topics

- [Concepts \(p. 351\)](#)
- [How to get started \(p. 352\)](#)
- [Related services \(p. 353\)](#)
- [Pricing and savings \(p. 353\)](#)

Concepts

Before you get started with Spot Instances, you should be familiar with the following concepts:

- *Spot Instance pool* – A set of unused EC2 instances with the same instance type (for example, `m5.large`), operating system, Availability Zone, and network platform.
- *Spot price* – The current price of a Spot Instance per hour.
- *Spot Instance request* – Requests a Spot Instance. The request provides the maximum price per hour that you are willing to pay for a Spot Instance. If you don't specify a maximum price, the default maximum price is the On-Demand price. When the maximum price per hour for your request exceeds the Spot price, Amazon EC2 fulfills your request if capacity is available. A Spot Instance request is either *one-time* or *persistent*. Amazon EC2 automatically resubmits a persistent Spot Instance request after the Spot Instance associated with the request is terminated. Your Spot Instance request can optionally specify a duration for the Spot Instances.
- *Spot Fleet* – A set of Spot Instances that is launched based on criteria that you specify. The Spot Fleet selects the Spot Instance pools that meet your needs and launches Spot Instances to meet the target capacity for the fleet. By default, Spot Fleets are set to *maintain* target capacity by launching replacement instances after Spot Instances in the fleet are terminated. You can submit a Spot Fleet as a one-time *request*, which does not persist after the instances have been terminated. You can include On-Demand Instance requests in a Spot Fleet request.
- *Spot Instance interruption* – Amazon EC2 terminates, stops, or hibernates your Spot Instance when the Spot price exceeds the maximum price for your request or capacity is no longer available. Amazon EC2 provides a Spot Instance interruption notice, which gives the instance a two-minute warning before it is interrupted.

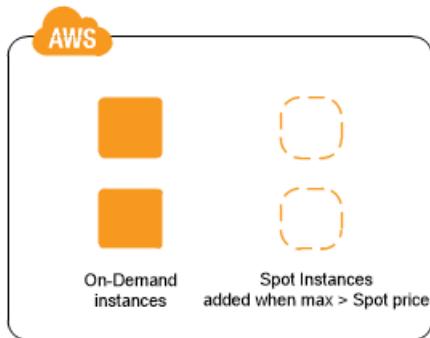
Key differences between Spot Instances and On-Demand Instances

The following table lists the key differences between Spot Instances and On-Demand Instances.

	Spot Instances	On-Demand Instances
Launch time	Can only be launched immediately if the Spot Request is active and capacity is available.	Can only be launched immediately if you make a manual launch request and capacity is available.
Available capacity	If capacity is not available, the Spot Request continues to automatically make the launch request until capacity becomes available.	If capacity is not available when you make a launch request, you get an insufficient capacity error (ICE).
Hourly price	The hourly price for Spot Instances varies based on demand.	The hourly price for On-Demand Instances is static.
Instance interruption	You can stop and start an Amazon EBS-backed Spot Instance. In addition, the Amazon EC2 Spot service can interrupt (p. 426) an individual Spot Instance if capacity is no longer available, the Spot price exceeds your maximum price, or demand for Spot Instances increases.	You determine when an On-Demand Instance is interrupted (stopped, hibernated, or terminated).

Strategies for using Spot Instances

One strategy is to maintain a minimum level of guaranteed compute resources for your applications by launching a core group of On-Demand Instances, and supplementing them with Spot Instances when the opportunity arises.



Another strategy is to launch Spot Instances with a specified duration (also known as Spot blocks), which are designed not to be interrupted and will run continuously for the duration you select. In rare situations, Spot blocks may be interrupted due to Amazon EC2 capacity needs. In these cases, we provide a two-minute warning before we terminate an instance, and you are not charged for the terminated instances even if you used them. For more information, see [Defining a duration for your Spot Instances \(p. 369\)](#).

How to get started

The first thing you need to do is get set up to use Amazon EC2. It can also be helpful to have experience launching On-Demand Instances before launching Spot Instances.

Get up and running

- [Setting up with Amazon EC2 \(p. 26\)](#)
- [Tutorial: Getting started with Amazon EC2 Linux instances \(p. 30\)](#)

Spot basics

- [How Spot Instances work \(p. 356\)](#)
- [How Spot Fleet works \(p. 358\)](#)

Working with Spot Instances

- [Preparing for interruptions \(p. 430\)](#)
- [Creating a Spot Instance request \(p. 371\)](#)
- [Getting request status information \(p. 424\)](#)

Working with Spot Fleets

- [Spot Fleet permissions \(p. 387\)](#)
- [Creating a Spot Fleet request \(p. 391\)](#)

Related services

You can provision Spot Instances directly using Amazon EC2. You can also provision Spot Instances using other services in AWS. For more information, see the following documentation.

Amazon EC2 Auto Scaling and Spot Instances

You can create launch templates or configurations with the maximum price that you are willing to pay, so that Amazon EC2 Auto Scaling can launch Spot Instances. For more information, see [Launching Spot Instances in Your Auto Scaling Group](#) and [Using Multiple Instance Types and Purchase Options in the Amazon EC2 Auto Scaling User Guide](#).

Amazon EMR and Spot Instances

There are scenarios where it can be useful to run Spot Instances in an Amazon EMR cluster. For more information, see [Spot Instances](#) and [When Should You Use Spot Instances](#) in the *Amazon EMR Management Guide*.

AWS CloudFormation templates

AWS CloudFormation enables you to create and manage a collection of AWS resources using a template in JSON format. AWS CloudFormation templates can include the maximum price you are willing to pay. For more information, see [EC2 Spot Instance Updates - Auto Scaling and CloudFormation Integration](#).

AWS SDK for Java

You can use the Java programming language to manage your Spot Instances. For more information, see [Tutorial: Amazon EC2 Spot Instances](#) and [Tutorial: Advanced Amazon EC2 Spot Request Management](#).

AWS SDK for .NET

You can use the .NET programming environment to manage your Spot Instances. For more information, see [Tutorial: Amazon EC2 Spot Instances](#).

Pricing and savings

You pay the Spot price for Spot Instances, which is set by Amazon EC2 and adjusted gradually based on the long-term supply of and demand for Spot Instances. If the maximum price for your request exceeds the current Spot price, Amazon EC2 fulfills your request if capacity is available. Your Spot Instances run

until you terminate them, capacity is no longer available, the Spot price exceeds your maximum price, or your Amazon EC2 Auto Scaling group terminates them during [scale in](#).

Spot Instances with a predefined duration use a fixed hourly price that remains in effect for the Spot Instance while it runs.

If you or Amazon EC2 interrupts a running Spot Instance, you are charged for the seconds used or the full hour, or you receive no charge, depending on the operating system used and who interrupted the Spot Instance. For more information, see [Billing for interrupted Spot Instances \(p. 433\)](#).

View prices

To view the current (updated every five minutes) lowest Spot price per AWS Region and instance type, see the [Spot Instances Pricing](#) page.

To view the Spot price history for the past three months, use the Amazon EC2 console or the `describe-spot-price-history` command (AWS CLI). For more information, see [Spot Instance pricing history \(p. 364\)](#).

We independently map Availability Zones to codes for each AWS account. Therefore, you can get different results for the same Availability Zone code (for example, `us-west-2a`) between different accounts.

View savings

You can view the savings made from using Spot Instances for a single Spot Fleet or for all Spot Instances. You can view the savings made in the last hour or the last three days, and you can view the average cost per vCPU hour and per memory (GiB) hour. Savings are estimated and may differ from actual savings because they do not include the billing adjustments for your usage. For more information about viewing savings information, see [Savings from purchasing Spot Instances \(p. 365\)](#).

View billing

To review your bill, go to your [AWS Account Activity page](#). Your bill contains links to usage reports that provide details about your bill. For more information, see [AWS Account Billing](#).

If you have questions concerning AWS billing, accounts, and events, [contact AWS Support](#).

Best practices for EC2 Spot

Amazon EC2 Spot Instances are spare EC2 compute capacity in the AWS Cloud that are available to you at savings of up to 90% off compared to On-Demand prices. The only difference between On-Demand Instances and Spot Instances is that Spot Instances can be interrupted by Amazon EC2, with two minutes of notification, when Amazon EC2 needs the capacity back.

Spot Instances are recommended for stateless, fault-tolerant, flexible applications. For example, Spot Instances work well for big data, containerized workloads, CI/CD, stateless web servers, high performance computing (HPC), and rendering workloads.

While running, Spot Instances are exactly the same as On-Demand Instances. However, Spot does not guarantee that you can keep your running instances long enough to finish your workloads. Spot also does not guarantee that you can get immediate availability of the instances that you are looking for, or that you can always get the aggregate capacity that you requested. Moreover, Spot Instance interruptions and capacity can change over time because Spot Instance availability varies based on supply and demand, and past performance isn't a guarantee of future results.

Spot Instances are not suitable for workloads that are inflexible, stateful, fault-intolerant, or tightly coupled between instance nodes. It's also not recommended for workloads that are intolerant of

occasional periods when the target capacity is not completely available. We strongly warn against using Spot Instances for these workloads or attempting to fail-over to On-Demand Instances to handle interruptions.

Regardless of whether you're an experienced Spot user or new to Spot Instances, if you are currently experiencing issues with Spot Instance interruptions or availability, we recommend that you follow these best practices to have the best experience using the Spot service.

Spot best practices

- [Prepare individual instances for interruptions \(p. 355\)](#)
- [Be flexible about instance types and Availability Zones \(p. 355\)](#)
- [Use EC2 Auto Scaling groups or Spot Fleet to manage your aggregate capacity \(p. 355\)](#)
- [Use the capacity optimized allocation strategy \(p. 356\)](#)
- [Use integrated AWS services to manage your Spot Instances \(p. 356\)](#)

Prepare individual instances for interruptions

The best way for you to gracefully handle Spot Instance interruptions is to architect your application to be fault-tolerant. To accomplish this, you can take advantage of Spot Instance interruption notices. A Spot Instance interruption notice is a warning that is issued two minutes before Amazon EC2 interrupts a Spot Instance. We recommend that you create a rule in [Amazon EventBridge](#) that captures the interruption notification, and then triggers a checkpoint for the progress of your workload or gracefully handles the interruption. For a detailed example that walks you through how to create and use event rules, see [Taking Advantage of Amazon EC2 Spot Instance Interruption Notices](#).

If your workload is "time-flexible," you can also configure your Spot Instances to be stopped or hibernated when they are interrupted. Amazon EC2 automatically stops or hibernates your Spot Instances on interruption, and automatically resumes the instances when we have available capacity.

For more information, see [Spot Instance interruptions \(p. 426\)](#).

Be flexible about instance types and Availability Zones

A Spot Instance pool is a set of unused EC2 instances with the same instance type (for example, m5.large) and Availability Zone (for example, us-east-1a). You should be flexible about which instance types you request and in which Availability Zones you can deploy your workload. This gives Spot a better chance to find and allocate your required amount of compute capacity. For example, don't just ask for c5.large if you'd be willing to use larges from the c4, m5, and m4 families.

Depending on your specific needs, you can evaluate which instance types you can be flexible across to fulfill your compute requirements. If a workload can be vertically scaled, you should include larger instance types (more vCPUs and memory) in your requests. If you can only scale horizontally, you should include older generation instance types because they are less in demand from On-Demand customers.

A good rule of thumb is to be flexible across at least 10 instance types for each workload. In addition, make sure that all Availability Zones are configured for use in your VPC and selected for your workload.

Use EC2 Auto Scaling groups or Spot Fleet to manage your aggregate capacity

Spot enables you to think in terms of aggregate capacity—in units that include vCPUs, memory, storage, or network throughput—rather than thinking in terms of individual instances. Auto Scaling groups and Spot Fleet enable you to launch and maintain a target capacity, and to automatically request resources to replace any that are disrupted or manually terminated. When you configure an Auto Scaling group or a Spot Fleet, you need only specify the instance types and target capacity based on your application needs. For more information, see [Auto Scaling Groups](#) in the *Amazon EC2 Auto Scaling User Guide* and [Creating a Spot Fleet request \(p. 391\)](#) in this user guide.

Use the capacity optimized allocation strategy

Allocation strategies in Auto Scaling groups help you to provision your target capacity without the need to manually look for the Spot Instance pools with spare capacity. We recommend using the capacity optimized strategy because this strategy automatically provisions instances from the most-available Spot Instance pools. You can also take advantage of the capacity optimized allocation strategy in Spot Fleet. Because your Spot Instance capacity is sourced from pools with optimal capacity, this decreases the possibility that your Spot Instances are reclaimed. For more information about allocation strategies, see [Spot Instances](#) in the *Amazon EC2 Auto Scaling User Guide* and [Configuring Spot Fleet for capacity optimization \(p. 359\)](#) in this user guide.

Use integrated AWS services to manage your Spot Instances

Other AWS services integrate with Spot to reduce overall compute costs without the need to manage the individual instances or fleets. We recommend that you consider the following solutions for your applicable workloads: Amazon EMR, Amazon ECS, AWS Batch, Amazon EKS, SageMaker, AWS Elastic Beanstalk, and Amazon GameLift. To learn more about Spot best practices with these services, see the [Amazon EC2 Spot Instances Workshops Website](#).

How Spot Instances work

To launch a Spot Instance, either you create a *Spot Instance request*, or Amazon EC2 creates a Spot Instance request on your behalf. The Spot Instance launches when the Spot Instance request is fulfilled.

You can launch a Spot Instance using several different services. For more information, see [Getting Started with Amazon EC2 Spot Instances](#). In this user guide, we describe the following ways to launch a Spot Instance using EC2:

- You can create a Spot Instance request. For more information, see [Creating a Spot Instance request \(p. 371\)](#).
- You can create a Spot Fleet request, in which you specify the desired number of Spot Instances. Amazon EC2 creates a Spot Instance request on your behalf for every Spot Instance that is specified in the Spot Fleet request. For more information, see [Creating a Spot Fleet request \(p. 391\)](#).
- You can create an EC2 Fleet, in which you specify the desired number of Spot Instances. Amazon EC2 creates a Spot Instance request on your behalf for every Spot Instance that is specified in the EC2 Fleet. For more information, see [Creating an EC2 Fleet \(p. 543\)](#).

The Spot Instance request must include the maximum price that you're willing to pay per hour per instance. If you don't specify a price, the price defaults to the On-Demand price. The request can include other constraints such as the instance type and Availability Zone.

Your Spot Instance launches if the maximum price that you're willing to pay exceeds the Spot price, and if there is available capacity. If the maximum price you're willing to pay is lower than the Spot price, then your instance does not launch. However, because Amazon EC2 gradually adjusts the Spot price based on the long-term supply of and demand for Spot Instances, the maximum price you're willing to pay might eventually exceed the Spot price, in which case your instance will launch.

Your Spot Instance runs until you stop or terminate it, or until Amazon EC2 interrupts it (known as a *Spot Instance interruption*).

When you use Spot Instances, you must be prepared for interruptions. Amazon EC2 can interrupt your Spot Instance when the Spot price exceeds your maximum price, when the demand for Spot Instances rises, or when the supply of Spot Instances decreases. When Amazon EC2 interrupts a Spot Instance, it provides a Spot Instance interruption notice, which gives the instance a two-minute warning before Amazon EC2 interrupts it. You can't enable termination protection for Spot Instances. For more information, see [Spot Instance interruptions \(p. 426\)](#).

You can stop, start, reboot, or terminate an Amazon EBS-backed Spot Instance. The Spot service can stop, terminate, or hibernate a Spot Instance when it interrupts it.

Contents

- [Launching Spot Instances in a launch group \(p. 357\)](#)
- [Launching Spot Instances in an Availability Zone group \(p. 357\)](#)
- [Launching Spot Instances in a VPC \(p. 357\)](#)

Launching Spot Instances in a launch group

Specify a launch group in your Spot Instance request to tell Amazon EC2 to launch a set of Spot Instances only if it can launch them all. In addition, if the Spot service must terminate one of the instances in a launch group (for example, if the Spot price exceeds your maximum price), it must terminate them all. However, if you terminate one or more of the instances in a launch group, Amazon EC2 does not terminate the remaining instances in the launch group.

Although this option can be useful, adding this constraint can decrease the chances that your Spot Instance request is fulfilled and increase the chances that your Spot Instances are terminated. For example, your launch group includes instances in multiple Availability Zones. If capacity in one of these Availability Zones decreases and is no longer available, then Amazon EC2 terminates all instances for the launch group.

If you create another successful Spot Instance request that specifies the same (existing) launch group as an earlier successful request, then the new instances are added to the launch group. Subsequently, if an instance in this launch group is terminated, all instances in the launch group are terminated, which includes instances launched by the first and second requests.

Launching Spot Instances in an Availability Zone group

Specify an Availability Zone group in your Spot Instance request to tell the Spot service to launch a set of Spot Instances in the same Availability Zone. Amazon EC2 need not interrupt all instances in an Availability Zone group at the same time. If Amazon EC2 must interrupt one of the instances in an Availability Zone group, the others remain running.

Although this option can be useful, adding this constraint can lower the chances that your Spot Instance request is fulfilled.

If you specify an Availability Zone group but don't specify an Availability Zone in the Spot Instance request, the result depends on the network you specified.

Default VPC

Amazon EC2 uses the Availability Zone for the specified subnet. If you don't specify a subnet, it selects an Availability Zone and its default subnet, but not necessarily the lowest-priced zone. If you deleted the default subnet for an Availability Zone, then you must specify a different subnet.

Nondefault VPC

Amazon EC2 uses the Availability Zone for the specified subnet.

Launching Spot Instances in a VPC

You specify a subnet for your Spot Instances the same way that you specify a subnet for your On-Demand Instances.

- You should use the default maximum price (the On-Demand price), or base your maximum price on the Spot price history of Spot Instances in a VPC.

- [Default VPC] If you want your Spot Instance launched in a specific low-priced Availability Zone, you must specify the corresponding subnet in your Spot Instance request. If you do not specify a subnet, Amazon EC2 selects one for you, and the Availability Zone for this subnet might not have the lowest Spot price.
- [Nondefault VPC] You must specify the subnet for your Spot Instance.

How Spot Fleet works

A *Spot Fleet* is a collection, or fleet, of Spot Instances, and optionally On-Demand Instances.

The Spot Fleet attempts to launch the number of Spot Instances and On-Demand Instances to meet the target capacity that you specified in the Spot Fleet request. The request for Spot Instances is fulfilled if there is available capacity and the maximum price you specified in the request exceeds the current Spot price. The Spot Fleet also attempts to maintain its target capacity fleet if your Spot Instances are interrupted.

You can also set a maximum amount per hour that you're willing to pay for your fleet, and Spot Fleet launches instances until it reaches the maximum amount. When the maximum amount you're willing to pay is reached, the fleet stops launching instances even if it hasn't met the target capacity.

A *Spot Instance pool* is a set of unused EC2 instances with the same instance type (for example, `m5.1.large`), operating system, Availability Zone, and network platform. When you make a Spot Fleet request, you can include multiple launch specifications, that vary by instance type, AMI, Availability Zone, or subnet. The Spot Fleet selects the Spot Instance pools that are used to fulfill the request, based on the launch specifications included in your Spot Fleet request, and the configuration of the Spot Fleet request. The Spot Instances come from the selected pools.

Contents

- [On-Demand in Spot Fleet \(p. 358\)](#)
- [Allocation strategy for Spot Instances \(p. 359\)](#)
- [Spot price overrides \(p. 360\)](#)
- [Control spending \(p. 360\)](#)
- [Spot Fleet instance weighting \(p. 361\)](#)
- [Walkthrough: Using Spot Fleet with instance weighting \(p. 362\)](#)

On-Demand in Spot Fleet

To ensure that you always have instance capacity, you can include a request for On-Demand capacity in your Spot Fleet request. In your Spot Fleet request, you specify your desired target capacity and how much of that capacity must be On-Demand. The balance comprises Spot capacity, which is launched if there is available Amazon EC2 capacity and availability. For example, if in your Spot Fleet request you specify target capacity as 10 and On-Demand capacity as 8, Amazon EC2 launches 8 capacity units as On-Demand, and 2 capacity units ($10-8=2$) as Spot.

Prioritizing instance types for On-Demand capacity

When Spot Fleet attempts to fulfill your On-Demand capacity, it defaults to launching the lowest-priced instance type first. If `OnDemandAllocationStrategy` is set to `prioritized`, Spot Fleet uses priority to determine which instance type to use first in fulfilling On-Demand capacity. The priority is assigned to the launch template override, and the highest priority is launched first.

For example, you have configured three launch template overrides, each with a different instance type: `c3.1.large`, `c4.1.large`, and `c5.1.large`. The On-Demand price for `c5.1.large` is less than for `c4.1.large`. `c3.1.large` is the cheapest. If you do not use priority to determine the order, the fleet fulfills On-Demand

capacity by starting with `c3.large`, and then `c5.large`. Because you often have unused Reserved Instances for `c4.large`, you can set the launch template override priority so that the order is `c4.large`, `c3.large`, and then `c5.large`.

Allocation strategy for Spot Instances

The allocation strategy for the Spot Instances in your Spot Fleet determines how it fulfills your Spot Fleet request from the possible Spot Instance pools represented by its launch specifications. The following are the allocation strategies that you can specify in your Spot Fleet request:

`lowestPrice`

The Spot Instances come from the pool with the lowest price. This is the default strategy.
`diversified`

The Spot Instances are distributed across all pools.

`capacityOptimized`

The Spot Instances come from the pool with optimal capacity for the number of instances that are launching.

`InstancePoolsToUseCount`

The Spot Instances are distributed across the number of Spot pools that you specify. This parameter is valid only when used in combination with `lowestPrice`.

Maintaining target capacity

After Spot Instances are terminated due to a change in the Spot price or available capacity of a Spot Instance pool, a Spot Fleet of type `maintain` launches replacement Spot Instances. If the allocation strategy is `lowestPrice`, the fleet launches replacement instances in the pool where the Spot price is currently the lowest. If the allocation strategy is `diversified`, the fleet distributes the replacement Spot Instances across the remaining pools. If the allocation strategy is `lowestPrice` in combination with `InstancePoolsToUseCount`, the fleet selects the Spot pools with the lowest price and launches Spot Instances across the number of Spot pools that you specify.

Configuring Spot Fleet for cost optimization

To optimize the costs for your use of Spot Instances, specify the `lowestPrice` allocation strategy so that Spot Fleet automatically deploys the least expensive combination of instance types and Availability Zones based on the current Spot price.

For On-Demand Instance target capacity, Spot Fleet always selects the least expensive instance type based on the public On-Demand price, while continuing to follow the allocation strategy (either `lowestPrice`, `capacityOptimized`, or `diversified`) for Spot Instances.

Configuring Spot Fleet for cost optimization and diversification

To create a fleet of Spot Instances that is both cheap and diversified, use the `lowestPrice` allocation strategy in combination with `InstancePoolsToUseCount`. Spot Fleet automatically deploys the cheapest combination of instance types and Availability Zones based on the current Spot price across the number of Spot pools that you specify. This combination can be used to avoid the most expensive Spot Instances.

Configuring Spot Fleet for capacity optimization

With Spot Instances, pricing changes slowly over time based on long-term trends in supply and demand, but capacity fluctuates in real time. The `capacityOptimized` strategy automatically launches Spot

Instances into the most available pools by looking at real-time capacity data and predicting which are the most available. This works well for workloads such as big data and analytics, image and media rendering, machine learning, and high performance computing that may have a higher cost of interruption associated with restarting work and checkpointing. By offering the possibility of fewer interruptions, the `capacityOptimized` strategy can lower the overall cost of your workload.

Choosing an appropriate allocation strategy

You can optimize your Spot Fleets based on your use case.

If your fleet is small or runs for a short time, the probability that your Spot Instances may be interrupted is low, even with all the instances in a single Spot Instance pool. Therefore, the `lowestPrice` strategy is likely to meet your needs while providing the lowest cost.

If your fleet is large or runs for a long time, you can improve the availability of your fleet by distributing the Spot Instances across multiple pools. For example, if your Spot Fleet request specifies 10 pools and a target capacity of 100 instances, the fleet launches 10 Spot Instances in each pool. If the Spot price for one pool exceeds your maximum price for this pool, only 10% of your fleet is affected. Using this strategy also makes your fleet less sensitive to increases in the Spot price in any one pool over time.

With the `diversified` strategy, the Spot Fleet does not launch Spot Instances into any pools with a Spot price that is equal to or higher than the [On-Demand price](#).

To create a cheap and diversified fleet, use the `lowestPrice` strategy in combination with `InstancePoolsToUseCount`. You can use a low or high number of Spot pools across which to allocate your Spot Instances. For example, if you run batch processing, we recommend specifying a low number of Spot pools (for example, `InstancePoolsToUseCount=2`) to ensure that your queue always has compute capacity while maximizing savings. If you run a web service, we recommend specifying a high number of Spot pools (for example, `InstancePoolsToUseCount=10`) to minimize the impact if a Spot Instance pool becomes temporarily unavailable.

If your fleet runs workloads that may have a higher cost of interruption associated with restarting work and checkpointing, then use the `capacityOptimized` strategy. This strategy offers the possibility of fewer interruptions, which can lower the overall cost of your workload.

Spot price overrides

Each Spot Fleet request can include a global maximum price, or use the default (the On-Demand price). Spot Fleet uses this as the default maximum price for each of its launch specifications.

You can optionally specify a maximum price in one or more launch specifications. This price is specific to the launch specification. If a launch specification includes a specific price, the Spot Fleet uses this maximum price, overriding the global maximum price. Any other launch specifications that do not include a specific maximum price still use the global maximum price.

Control spending

Spot Fleet stops launching instances when it has either reached the target capacity or the maximum amount you're willing to pay. To control the amount you pay per hour for your fleet, you can specify the `SpotMaxTotalPrice` for Spot Instances and the `OnDemandMaxTotalPrice` for On-Demand Instances. When the maximum total price is reached, Spot Fleet stops launching instances even if it hasn't met the target capacity.

The following examples show two different scenarios. In the first, Spot Fleet stops launching instances when it has met the target capacity. In the second, Spot Fleet stops launching instances when it has reached the maximum amount you're willing to pay.

Example: Stop launching instances when target capacity is reached

Given a request for `m4.large` On-Demand Instances, where:

- On-Demand Price: \$0.10 per hour
- `OnDemandTargetCapacity`: 10
- `OnDemandMaxTotalPrice`: \$1.50

Spot Fleet launches 10 On-Demand Instances because the total of \$1.00 (10 instances x \$0.10) does not exceed the `OnDemandMaxTotalPrice` of \$1.50.

Example: Stop launching instances when maximum total price is reached

Given a request for `m4.large` On-Demand Instances, where:

- On-Demand Price: \$0.10 per hour
- `OnDemandTargetCapacity`: 10
- `OnDemandMaxTotalPrice`: \$0.80

If Spot Fleet launches the On-Demand target capacity (10 On-Demand Instances), the total cost per hour would be \$1.00. This is more than the amount (\$0.80) specified for `OnDemandMaxTotalPrice`. To prevent spending more than you're willing to pay, Spot Fleet launches only 8 On-Demand Instances (below the On-Demand target capacity) because launching more would exceed the `OnDemandMaxTotalPrice`.

Spot Fleet instance weighting

When you request a fleet of Spot Instances, you can define the capacity units that each instance type would contribute to your application's performance, and adjust your maximum price for each Spot Instance pool accordingly using *instance weighting*.

By default, the price that you specify is *per instance hour*. When you use the instance weighting feature, the price that you specify is *per unit hour*. You can calculate your price per unit hour by dividing your price for an instance type by the number of units that it represents. Spot Fleet calculates the number of Spot Instances to launch by dividing the target capacity by the instance weight. If the result isn't an integer, the Spot Fleet rounds it up to the next integer, so that the size of your fleet is not below its target capacity. Spot Fleet can select any pool that you specify in your launch specification, even if the capacity of the instances launched exceeds the requested target capacity.

The following tables provide examples of calculations to determine the price per unit for a Spot Fleet request with a target capacity of 10.

Instance type	Instance weight	Price per instance hour	Price per unit hour	Number of instances launched
r3.xlarge	2	\$0.05	.025 (.05 divided by 2)	5 (10 divided by 2)

Instance type	Instance weight	Price per instance hour	Price per unit hour	Number of instances launched
r3.8xlarge	8	\$0.10	.0125	2

Instance type	Instance weight	Price per instance hour	Price per unit hour	Number of instances launched
			(.10 divided by 8)	(10 divided by 8, result rounded up)

Use Spot Fleet instance weighting as follows to provision the target capacity that you want in the pools with the lowest price per unit at the time of fulfillment:

1. Set the target capacity for your Spot Fleet either in instances (the default) or in the units of your choice, such as virtual CPUs, memory, storage, or throughput.
2. Set the price per unit.
3. For each launch configuration, specify the weight, which is the number of units that the instance type represents toward the target capacity.

Instance weighting example

Consider a Spot Fleet request with the following configuration:

- A target capacity of 24
- A launch specification with an instance type `r3.2xlarge` and a weight of 6
- A launch specification with an instance type `c3.xlarge` and a weight of 5

The weights represent the number of units that instance type represents toward the target capacity. If the first launch specification provides the lowest price per unit (price for `r3.2xlarge` per instance hour divided by 6), the Spot Fleet would launch four of these instances (24 divided by 6).

If the second launch specification provides the lowest price per unit (price for `c3.xlarge` per instance hour divided by 5), the Spot Fleet would launch five of these instances (24 divided by 5, result rounded up).

Instance weighting and allocation strategy

Consider a Spot Fleet request with the following configuration:

- A target capacity of 30
- A launch specification with an instance type `c3.2xlarge` and a weight of 8
- A launch specification with an instance type `m3.xlarge` and a weight of 8
- A launch specification with an instance type `r3.xlarge` and a weight of 8

The Spot Fleet would launch four instances (30 divided by 8, result rounded up). With the `lowestPrice` strategy, all four instances come from the pool that provides the lowest price per unit. With the `diversified` strategy, the Spot Fleet launches one instance in each of the three pools, and the fourth instance in whichever pool provides the lowest price per unit.

Walkthrough: Using Spot Fleet with instance weighting

This walkthrough uses a fictitious company called Example Corp to illustrate the process of requesting a Spot Fleet using instance weighting.

Objective

Example Corp, a pharmaceutical company, wants to leverage the computational power of Amazon EC2 for screening chemical compounds that might be used to fight cancer.

Planning

Example Corp first reviews [Spot Best Practices](#). Next, Example Corp determines the following requirements for their Spot Fleet.

Instance types

Example Corp has a compute- and memory-intensive application that performs best with at least 60 GB of memory and eight virtual CPUs (vCPUs). They want to maximize these resources for the application at the lowest possible price. Example Corp decides that any of the following EC2 instance types would meet their needs:

Instance type	Memory (GiB)	vCPUs
r3.2xlarge	61	8
r3.4xlarge	122	16
r3.8xlarge	244	32

Target capacity in units

With instance weighting, target capacity can equal a number of instances (the default) or a combination of factors such as cores (vCPUs), memory (GiBs), and storage (GBs). By considering the base for their application (60 GB of RAM and eight vCPUs) as 1 unit, Example Corp decides that 20 times this amount would meet their needs. So the company sets the target capacity of their Spot Fleet request to 20.

Instance weights

After determining the target capacity, Example Corp calculates instance weights. To calculate the instance weight for each instance type, they determine the units of each instance type that are required to reach the target capacity as follows:

- r3.2xlarge (61.0 GB, 8 vCPUs) = 1 unit of 20
- r3.4xlarge (122.0 GB, 16 vCPUs) = 2 units of 20
- r3.8xlarge (244.0 GB, 32 vCPUs) = 4 units of 20

Therefore, Example Corp assigns instance weights of 1, 2, and 4 to the respective launch configurations in their Spot Fleet request.

Price per unit hour

Example Corp uses the [On-Demand price](#) per instance hour as a starting point for their price. They could also use recent Spot prices, or a combination of the two. To calculate the price per unit hour, they divide their starting price per instance hour by the weight. For example:

Instance type	On-Demand price	Instance weight	Price per unit hour
r3.2xLarge	\$0.7	1	\$0.7
r3.4xLarge	\$1.4	2	\$0.7
r3.8xLarge	\$2.8	4	\$0.7

Example Corp could use a global price per unit hour of \$0.7 and be competitive for all three instance types. They could also use a global price per unit hour of \$0.7 and a specific price per unit hour of \$0.9 in the r3.8xlarge launch specification.

Verifying permissions

Before creating a Spot Fleet request, Example Corp verifies that it has an IAM role with the required permissions. For more information, see [Spot Fleet permissions \(p. 387\)](#).

Creating the request

Example Corp creates a file, config.json, with the following configuration for its Spot Fleet request:

```
{  
    "SpotPrice": "0.70",  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.2xlarge",  
            "SubnetId": "subnet-482e4972",  
            "WeightedCapacity": 1  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.4xlarge",  
            "SubnetId": "subnet-482e4972",  
            "WeightedCapacity": 2  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.8xlarge",  
            "SubnetId": "subnet-482e4972",  
            "SpotPrice": "0.90",  
            "WeightedCapacity": 4  
        }  
    ]  
}
```

Example Corp creates the Spot Fleet request using the [request-spot-fleet](#) command.

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

For more information, see [Spot Fleet requests \(p. 385\)](#).

Fulfillment

The allocation strategy determines which Spot Instance pools your Spot Instances come from.

With the `lowestPrice` strategy (which is the default strategy), the Spot Instances come from the pool with the lowest price per unit at the time of fulfillment. To provide 20 units of capacity, the Spot Fleet launches either 20 `r3.2xlarge` instances (20 divided by 1), 10 `r3.4xlarge` instances (20 divided by 2), or 5 `r3.8xlarge` instances (20 divided by 4).

If Example Corp used the `diversified` strategy, the Spot Instances would come from all three pools. The Spot Fleet would launch 6 `r3.2xlarge` instances (which provide 6 units), 3 `r3.4xlarge` instances (which provide 6 units), and 2 `r3.8xlarge` instances (which provide 8 units), for a total of 20 units.

Spot Instance pricing history

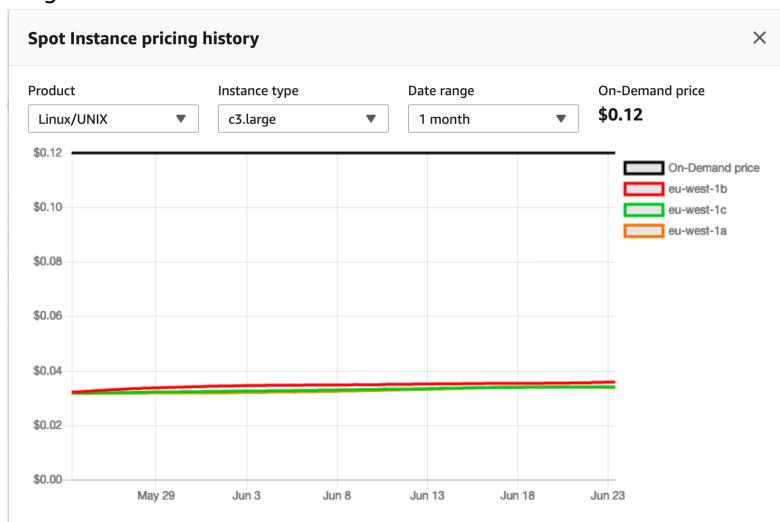
When you request Spot Instances, we recommend that you use the default maximum price (the On-Demand price). If you want to specify a maximum price, we recommend that you review the Spot price

history before you do so. You can view the Spot price history for the last 90 days, filtering by instance type, operating system, and Availability Zone.

Spot Instance prices are set by Amazon EC2 and adjust gradually based on long-term trends in supply and demand for Spot Instance capacity. For the *current* Spot Instance prices see [Amazon EC2 Spot Instances Pricing](#).

To view the Spot price history (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. If you are new to Spot Instances, you see a welcome page. Choose **Get started**, scroll to the bottom of the screen, and then choose **Cancel**.
4. Choose **Pricing history**.
5. Choose the operating system (**Product**), **Instance type**, and **Date range** for which to view the price history. Move your pointer over the graph to display the prices at specific times in the selected date range.



6. (Optional) To review the Spot price history for a specific Availability Zone, you can filter the Availability Zones by removing Availability Zones from the graph. To remove an Availability Zone from the graph, select the zone to remove it. You can also select a different product, instance type, or date range.

To view the Spot price history using the command line

You can use one of the following commands. For more information, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-spot-price-history](#) (AWS CLI)
- [Get-EC2SpotPriceHistory](#) (AWS Tools for Windows PowerShell)

Savings from purchasing Spot Instances

You can view the usage and savings information for Spot Instances at the per-fleet level, or for all running Spot Instances. At the per-fleet level, the usage and savings information includes all instances launched and terminated by the fleet. You can view this information from the last hour or the last three days.

The following screenshot from the Spot Requests page shows the Spot usage and savings information for a Spot Fleet.

Spot usage and savings																											
4 Spot Instances	266 vCPU-hours	700 Mem(GiB)-hours	\$9.55 On-Demand total	\$2.99 Spot total	69% Savings																						
				\$0.0112 Average cost per vCPU-hour	\$0.0043 Average cost per mem(GiB)-hour																						
Details																											
<table border="1"> <tr> <td>t3.medium (1)</td><td>2 vCPU hours</td><td>4 mem(GiB)-hours</td><td>\$0.01 total</td><td>70% savings</td><td></td><td></td></tr> <tr> <td>m4.large (1)</td><td>144 vCPU hours</td><td>576 mem(GiB)-hours</td><td>\$2.52 total</td><td>68% savings</td><td></td><td></td></tr> <tr> <td>t2.micro (2)</td><td>120 vCPU hours</td><td>120 mem(GiB)-hours</td><td>\$0.46 total</td><td>70% savings</td><td></td><td></td></tr> </table>							t3.medium (1)	2 vCPU hours	4 mem(GiB)-hours	\$0.01 total	70% savings			m4.large (1)	144 vCPU hours	576 mem(GiB)-hours	\$2.52 total	68% savings			t2.micro (2)	120 vCPU hours	120 mem(GiB)-hours	\$0.46 total	70% savings		
t3.medium (1)	2 vCPU hours	4 mem(GiB)-hours	\$0.01 total	70% savings																							
m4.large (1)	144 vCPU hours	576 mem(GiB)-hours	\$2.52 total	68% savings																							
t2.micro (2)	120 vCPU hours	120 mem(GiB)-hours	\$0.46 total	70% savings																							

You can view the following usage and savings information:

- **Spot Instances** – The number of Spot Instances launched and terminated by the Spot Fleet. When viewing the savings summary, the number represents all your running Spot Instances.
- **vCPU-hours** – The number of vCPU hours used across all the Spot Instances for the selected time frame.
- **Mem(GiB)-hours** – The number of GiB hours used across all the Spot Instances for the selected time frame.
- **On-Demand total** – The total amount you would've paid for the selected time frame had you launched these instances as On-Demand Instances.
- **Spot total** – The total amount to pay for the selected time frame.
- **Savings** – The percentage that you are saving by not paying the On-Demand price.
- **Average cost per vCPU-hour** – The average hourly cost of using the vCPUs across all the Spot Instances for the selected time frame, calculated as follows: **Average cost per vCPU-hour = Spot total / vCPU-hours**.
- **Average cost per mem(GiB)-hour** – The average hourly cost of using the GiBs across all the Spot Instances for the selected time frame, calculated as follows: **Average cost per mem(GiB)-hour = Spot total / Mem(GiB)-hours**.
- **Details** table – The different instance types (the number of instances per instance type is in parentheses) that comprise the Spot Fleet. When viewing the savings summary, these comprise all your running Spot Instances.

Savings information can only be viewed using the Amazon EC2 console.

To view the savings information for a Spot Fleet (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, choose **Spot Requests**.
3. Select a Spot Fleet request and choose **Savings**.
4. By default, the page displays usage and savings information for the last three days. You can choose **last hour** or the **last three days**. For Spot Fleets that were launched less than an hour ago, the page shows the estimated savings for the hour.

To view the savings information for all running Spot Instances (console)

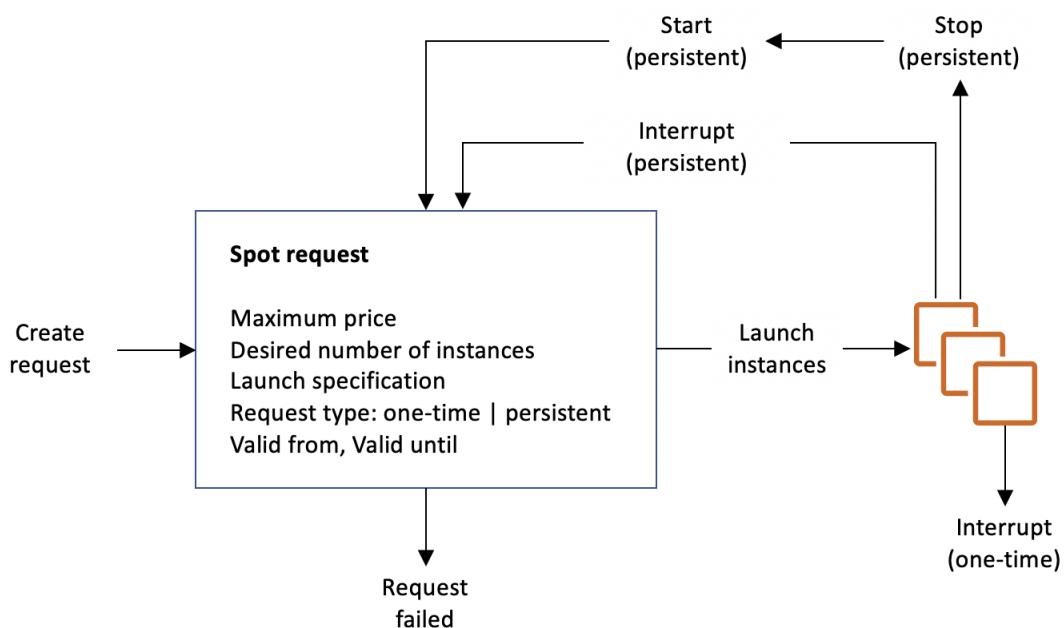
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. On the navigation pane, choose **Spot Requests**.
3. Choose **Savings Summary**.

Spot Instance requests

To use Spot Instances, you create a Spot Instance request that includes the desired number of instances, the instance type, the Availability Zone, and the maximum price that you are willing to pay per instance hour. If your maximum price exceeds the current Spot price, Amazon EC2 fulfills your request immediately if capacity is available. Otherwise, Amazon EC2 waits until your request can be fulfilled or until you cancel the request.

The following illustration shows how Spot requests work. Notice that the request type (one-time or persistent) determines whether the request is opened again when Amazon EC2 interrupts a Spot Instance or if you stop a Spot Instance. If the request is persistent, the request is opened again after your Spot Instance is interrupted. If the request is persistent and you stop your Spot Instance, the request only opens after you start your Spot Instance.



Contents

- [Spot Instance request states \(p. 368\)](#)
- [Defining a duration for your Spot Instances \(p. 369\)](#)
- [Specifying a tenancy for your Spot Instances \(p. 369\)](#)
- [Service-linked role for Spot Instance requests \(p. 370\)](#)
- [Creating a Spot Instance request \(p. 371\)](#)
- [Finding running Spot Instances \(p. 374\)](#)
- [Tagging Spot Instance requests \(p. 375\)](#)
- [Canceling a Spot Instance request \(p. 380\)](#)
- [Stopping a Spot Instance \(p. 381\)](#)
- [Starting a Spot Instance \(p. 381\)](#)
- [Terminating a Spot Instance \(p. 382\)](#)

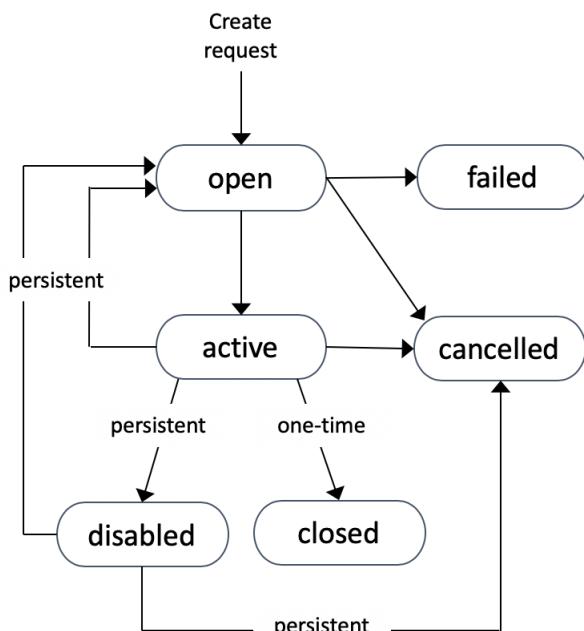
- Spot Instance request example launch specifications (p. 383)

Spot Instance request states

A Spot Instance request can be in one of the following states:

- **open** – The request is waiting to be fulfilled.
 - **active** – The request is fulfilled and has an associated Spot Instance.
 - **failed** – The request has one or more bad parameters.
 - **closed** – The Spot Instance was interrupted or terminated.
 - **disabled** – You stopped the Spot Instance.
 - **cancelled** – You canceled the request, or the request expired.

The following illustration represents the transitions between the request states. Notice that the transitions depend on the request type (one-time or persistent).



A one-time Spot Instance request remains active until Amazon EC2 launches the Spot Instance, the request expires, or you cancel the request. If the Spot price exceeds your maximum price or capacity is not available, your Spot Instance is terminated and the Spot Instance request is closed.

A persistent Spot Instance request remains active until it expires or you cancel it, even if the request is fulfilled. If the Spot price exceeds your maximum price or capacity is not available, your Spot Instance is interrupted. After your instance is interrupted, when your maximum price exceeds the Spot price or capacity becomes available again, the Spot Instance is started if stopped or resumed if hibernated. You can stop a Spot Instance and start it again if capacity is available and your maximum price exceeds the current Spot price. If the Spot Instance is terminated (irrespective of whether the Spot Instance is in a stopped or running state), the Spot Instance request is opened again and Amazon EC2 launches a new Spot Instance. For more information, see [Stopping a Spot Instance \(p. 381\)](#), [Starting a Spot Instance \(p. 381\)](#), and [Terminating a Spot Instance \(p. 382\)](#).

You can track the status of your Spot Instance requests, as well as the status of the Spot Instances launched, through the status. For more information, see [Spot request status \(p. 420\)](#).

Defining a duration for your Spot Instances

Spot Instances with a defined duration (also known as Spot blocks) are designed not to be interrupted and will run continuously for the duration you select. This makes them ideal for jobs that take a finite time to complete, such as batch processing, encoding and rendering, modeling and analysis, and continuous integration.

You can use a duration of 1, 2, 3, 4, 5, or 6 hours. The price that you pay depends on the specified duration. To view the current prices for a 1-hour duration or a 6-hour duration, see [Spot Instance Prices](#). You can use these prices to estimate the cost of the 2, 3, 4, and 5-hour durations. When a request with a duration is fulfilled, the price for your Spot Instance is fixed, and this price remains in effect until the instance terminates. You are billed at this price for each hour or partial hour that the instance is running. A partial instance hour is billed to the nearest second.

When you define a duration in your Spot request, the duration period for each Spot Instance starts as soon as the instance receives its instance ID. The Spot Instance runs until you terminate it or the duration period ends. At the end of the duration period, Amazon EC2 marks the Spot Instance for termination and provides a Spot Instance termination notice, which gives the instance a two-minute warning before it terminates. In rare situations, Spot blocks may be interrupted due to Amazon EC2 capacity needs. In these cases, we provide a two-minute warning before we terminate an instance, and you are not charged for the terminated instances even if you used them.

New accounts or accounts with no previous billing history with AWS are not eligible for Spot Instances with a defined duration (also known as Spot blocks).

To launch Spot Instances with a defined duration (console)

Follow the [Creating a Spot Fleet request \(p. 391\)](#) procedure. To launch Spot Instances with a defined duration, for **Tell us your application or task need**, choose **Defined duration workloads**.

To launch Spot Instances with a defined duration (AWS CLI)

To specify a duration for your Spot Instances, include the `--block-duration-minutes` option with the `request-spot-instances` command. For example, the following command creates a Spot request that launches Spot Instances that run for two hours.

```
aws ec2 request-spot-instances \
  --instance-count 5 \
  --block-duration-minutes 120 \
  --type "one-time" \
  --launch-specification file://specification.json
```

To retrieve the cost for Spot Instances with a defined duration (AWS CLI)

Use the `describe-spot-instance-requests` command to retrieve the fixed cost for your Spot Instances with a specified duration. The information is in the `actualBlockHourlyPrice` field.

Specifying a tenancy for your Spot Instances

You can run a Spot Instance on single-tenant hardware. Dedicated Spot Instances are physically isolated from instances that belong to other AWS accounts. For more information, see [Dedicated Instances \(p. 469\)](#) and the [Amazon EC2 Dedicated Instances](#) product page.

To run a Dedicated Spot Instance, do one of the following:

- Specify a tenancy of dedicated when you create the Spot Instance request. For more information, see [Creating a Spot Instance request \(p. 371\)](#).
- Request a Spot Instance in a VPC with an instance tenancy of dedicated. For more information, see [Creating a VPC with an Instance Tenancy of Dedicated \(p. 471\)](#). You cannot request a Spot Instance with a tenancy of default if you request it in a VPC with an instance tenancy of dedicated.

The following instance types support Dedicated Spot Instances.

Current generation

- c4.8xlarge
- d2.8xlarge
- i3.16xlarge
- m4.10xlarge
- m4.16xlarge
- p2.16xlarge
- r4.16xlarge
- x1.32xlarge

Previous generation

- c3.8xlarge
- cc2.8xlarge
- cr1.8xlarge
- g2.8xlarge
- i2.8xlarge
- r3.8xlarge

Service-linked role for Spot Instance requests

Amazon EC2 uses service-linked roles for the permissions that it requires to call other AWS services on your behalf. A service-linked role is a unique type of IAM role that is linked directly to an AWS service. Service-linked roles provide a secure way to delegate permissions to AWS services because only the linked service can assume a service-linked role. For more information, see [Using Service-Linked Roles](#) in the *IAM User Guide*.

Amazon EC2 uses the service-linked role named **AWSServiceRoleForEC2Spot** to launch and manage Spot Instances on your behalf.

Permissions granted by AWSServiceRoleForEC2Spot

Amazon EC2 uses **AWSServiceRoleForEC2Spot** to complete the following actions:

- `ec2:DescribeInstances` – Describe Spot Instances
- `ec2:StopInstances` – Stop Spot Instances
- `ec2:StartInstances` – Start Spot Instances

Create the service-linked role

Under most circumstances, you don't need to manually create a service-linked role. Amazon EC2 creates the **AWSServiceRoleForEC2Spot** service-linked role the first time you request a Spot Instance using the console.

If you had an active Spot Instance request before October 2017, when Amazon EC2 began supporting this service-linked role, Amazon EC2 created the **AWSServiceRoleForEC2Spot** role in your AWS account. For more information, see [A New Role Appeared in My Account](#) in the *IAM User Guide*.

Ensure that this role exists before you use the AWS CLI or an API to request a Spot Instance. To create the role, use the IAM console as follows.

To manually create the **AWSServiceRoleForEC2Spot** service-linked role

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**.
3. Choose **Create role**.
4. On the **Select type of trusted entity** page, choose **EC2, EC2 - Spot Instances, Next: Permissions**.
5. On the next page, choose **Next:Review**.
6. On the **Review** page, choose **Create role**.

If you no longer need to use Spot Instances, we recommend that you delete the **AWSServiceRoleForEC2Spot** role. After this role is deleted from your account, Amazon EC2 will create the role again if you request Spot Instances.

Granting access to CMKs for use with encrypted AMIs and EBS snapshots

If you specify an [encrypted AMI \(p. 157\)](#) or an [encrypted Amazon EBS snapshot \(p. 1121\)](#) for your Spot Instances and you use a customer managed customer master key (CMK) for encryption, you must grant the **AWSServiceRoleForEC2Spot** role permission to use the CMK so that Amazon EC2 can launch Spot Instances on your behalf. To do this, you must add a grant to the CMK, as shown in the following procedure.

When providing permissions, grants are an alternative to key policies. For more information, see [Using Grants](#) and [Using Key Policies in AWS KMS](#) in the *AWS Key Management Service Developer Guide*.

To grant the **AWSServiceRoleForEC2Spot** role permissions to use the CMK

- Use the `create-grant` command to add a grant to the CMK and to specify the principal (the **AWSServiceRoleForEC2Spot** service-linked role) that is given permission to perform the operations that the grant permits. The CMK is specified by the `key-id` parameter and the ARN of the CMK. The principal is specified by the `grantee-principal` parameter and the ARN of the **AWSServiceRoleForEC2Spot** service-linked role.

```
aws kms create-grant \
  --region us-east-1 \
  --key-id arn:aws:kms:us-
east-1:44445556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
  --grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2Spot \
  --operations "Decrypt" "Encrypt" "GenerateDataKey"
  "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
  "ReEncryptTo"
```

Creating a Spot Instance request

The procedure for requesting a Spot Instance is similar to the procedure for launching an On-Demand Instance. You can request a Spot Instance in the following ways:

- To request a Spot Instance using the console, use the launch instance wizard. For more information, see [To create a Spot Instance request \(console\) \(p. 372\)](#).
- To request a Spot Instance using the CLI, use the `request-spot-instances` command or the `run-instances` command. For more information, see [To create a Spot Instance request using request-spot-instances \(AWS CLI\)](#) and [To create a Spot Instance request using run-instances \(AWS CLI\)](#).
- To request a Spot Instance with a defined duration using the console, follow the [Creating a Spot Fleet request \(p. 391\)](#) procedure. For **Tell us your application or task need**, choose **Defined duration workloads**. For more information, see [Defining a duration for your Spot Instances \(p. 369\)](#).

- To request a Spot Instance with a defined duration using the CLI, use the [request-spot-instances](#) command and specify the `--block-duration-minutes` parameter. For more information, see [Defining a duration for your Spot Instances \(p. 369\)](#).

After you've submitted your Spot Instance request, you can't change the parameters of the request. This means that you can't make changes to the maximum price that you're willing to pay.

If you request multiple Spot Instances at one time, Amazon EC2 creates separate Spot Instance requests so that you can track the status of each request separately. For more information about tracking Spot Instance requests, see [Spot request status \(p. 420\)](#).

To launch a fleet that includes Spot Instances and On-Demand Instances, see [Creating a Spot Fleet request \(p. 391\)](#).

Note

You can't launch a Spot Instance and an On-Demand Instance in the same call using the launch instance wizard or the [run-instances](#) command.

Prerequisites

Before you begin, decide on your maximum price, how many Spot Instances you'd like, and what instance type to use. To review Spot price trends, see [Spot Instance pricing history \(p. 364\)](#).

To create a Spot Instance request (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation bar at the top of the screen, select a Region.
3. From the Amazon EC2 console dashboard, choose **Launch Instance**.
4. On the **Choose an Amazon Machine Image (AMI)** page, choose an AMI. For more information, see [Step 1: Choose an Amazon Machine Image \(AMI\) \(p. 499\)](#).
5. On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch, and then choose **Next: Configure Instance Details**. For more information, see [Step 2: Choose an Instance Type \(p. 500\)](#).
6. On the **Configure Instance Details** page, configure the Spot Instance request as follows:

- **Number of instances:** Enter the number of instances to launch.

Note

Amazon EC2 creates a separate request for each Spot Instance.

- (Optional) To help ensure that you maintain the correct number of instances to handle demand on your application, you can choose **Launch into Auto Scaling Group** to create a launch configuration and an Auto Scaling group. Auto Scaling scales the number of instances in the group according to your specifications. For more information, see the [Amazon EC2 Auto Scaling User Guide](#).
- **Purchasing option:** Choose **Request Spot instances** to launch a Spot Instance. When you choose this option, the following fields appear.
- **Current price:** The current Spot price in each Availability Zone is displayed for the instance type that you selected.
- (Optional) **Maximum price:** You can leave the field empty, or you can specify the maximum amount you're willing to pay.
 - If you leave the field empty, then the maximum price defaults to the current On-Demand price. Your Spot Instance launches at the current Spot price, not exceeding the On-Demand price.
 - If you specify a maximum price that is more than the current Spot Price, your Spot Instance launches and is charged at the current Spot price.
 - If you specify a maximum price that is lower than the Spot price, your Spot Instance is not launched.

- **Persistent request:** Choose **Persistent request** to resubmit the Spot Instance request if your Spot Instance is interrupted.
- **Interruption behavior:** By default, the Spot service terminates a Spot Instance when it is interrupted. If you choose **Persistent request**, you can then specify that the Spot service stops or hibernates your Spot Instance when it's interrupted. For more information, see [Interruption behaviors \(p. 427\)](#).
- (Optional) **Request valid to:** Choose **Edit** to specify when the Spot Instance request expires.

For more information about configuring your Spot Instance, see [Step 3: Configure Instance Details \(p. 500\)](#).

7. The AMI you selected includes one or more volumes of storage, including the root device volume. On the **Add Storage** page, you can specify additional volumes to attach to the instance by choosing **Add New Volume**. For more information, see [Step 4: Add Storage \(p. 503\)](#).
8. On the **Add Tags** page, specify [tags \(p. 1229\)](#) by providing key and value combinations. For more information, see [Step 5: Add Tags \(p. 503\)](#).
9. On the **Configure Security Group** page, use a security group to define firewall rules for your instance. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored. (For more information about security groups, see [Amazon EC2 security groups for Linux instances \(p. 999\)](#).) Select or create a security group, and then choose **Review and Launch**. For more information, see [Step 6: Configure Security Group \(p. 503\)](#).
10. On the **Review Instance Launch** page, check the details of your instance, and make any necessary changes by choosing the appropriate **Edit** link. When you are ready, choose **Launch**. For more information, see [Step 7: Review Instance Launch and Select Key Pair \(p. 504\)](#).
11. In the **Select an existing key pair or create a new key pair** dialog box, you can choose an existing key pair, or create a new one. For example, choose **Choose an existing key pair**, then select the key pair that you created when getting set up. For more information, see [Amazon EC2 key pairs and Linux instances \(p. 985\)](#).

Important

If you choose the **Proceed without key pair** option, you won't be able to connect to the instance unless you choose an AMI that is configured to allow users another way to log in.

12. To launch your instance, select the acknowledgment check box, then choose **Launch Instances**.

If the instance fails to launch or the state immediately goes to **terminated** instead of **running**, see [Troubleshooting instance launch issues \(p. 1244\)](#).

To create a Spot Instance request using [request-spot-instances \(AWS CLI\)](#)

Use the [request-spot-instances](#) command to create a one-time request.

```
aws ec2 request-spot-instances \
--instance-count 5 \
--type "one-time" \
--launch-specification file://specification.json
```

Use the [request-spot-instances](#) command to create a persistent request.

```
aws ec2 request-spot-instances \
--instance-count 5 \
--type "persistent" \
--launch-specification file://specification.json
```

For example launch specification files to use with these commands, see [Spot Instance request example launch specifications \(p. 383\)](#). If you download a launch specification file from the console, you must use the [request-spot-fleet](#) command instead (the console specifies a Spot request using a Spot Fleet).

To create a Spot Instance request using `run-instances` (AWS CLI)

Use the `run-instances` command and specify the Spot Instance options in the `--instance-market-options` parameter.

```
aws ec2 run-instances \
--image-id ami-0abcdef1234567890 \
--instance-type t2.micro \
--count 5 \
--subnet-id subnet-08fc749671b2d077c \
--key-name MyKeyPair \
--security-group-ids sg-0b0384b66d7d692f9 \
--instance-market-options file://spot-options.json
```

The following is the data structure to specify in the JSON file for `--instance-market-options`. You can also specify `BlockDurationMinutes`, `ValidUntil`, and `InstanceInterruptionBehavior`. If you do not specify a field in the data structure, the default value is used. This example creates a one-time request and specifies `0.02` as the maximum price you're willing to pay for the Spot Instance.

```
{
  "MarketType": "spot",
  "SpotOptions": {
    "MaxPrice": "0.02",
    "SpotInstanceType": "one-time"
  }
}
```

Finding running Spot Instances

Amazon EC2 launches a Spot Instance when the maximum price exceeds the Spot price and capacity is available. A Spot Instance runs until it is interrupted or you terminate it yourself. If your maximum price is exactly equal to the Spot price, there is a chance that your Spot Instance remains running, depending on demand.

To find running Spot Instances (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**. You can see both Spot Instance requests and Spot Fleet requests. If a Spot Instance request has been fulfilled, **Capacity** is the ID of the Spot Instance. For a Spot Fleet, **Capacity** indicates how much of the requested capacity has been fulfilled. To view the IDs of the instances in a Spot Fleet, choose the expand arrow, or select the fleet and choose **Instances**.

Note

For Spot Instance requests that are created by a Spot Fleet, the requests are not tagged instantly with the system tag that indicates the Spot Fleet to which they belong, and for a period of time may appear separate from Spot Fleet request.

Alternatively, in the navigation pane, choose **Instances**. In the top right corner, choose the settings icon (), and then under **Attribute columns**, select **Instance lifecycle**. For each instance, **Instance lifecycle** is either normal, spot, or scheduled.

To find running Spot Instances (AWS CLI)

To enumerate your Spot Instances, use the `describe-spot-instance-requests` command with the `--query` option.

```
aws ec2 describe-spot-instance-requests \
--query "SpotInstanceRequests[*].{ID:InstanceId}"
```

The following is example output:

```
[  
  {  
    "ID": "i-1234567890abcdef0"  
  },  
  {  
    "ID": "i-0598c7d356eba48d7"  
  }  
]
```

Alternatively, you can enumerate your Spot Instances using the [describe-instances](#) command with the `--filters` option.

```
aws ec2 describe-instances \
--filters "Name=instance-lifecycle,Values=spot"
```

To describe a single Spot Instance instance, use the [describe-spot-instance-requests](#) command with the `--spot-instance-request-ids` option.

```
aws ec2 describe-spot-instance-requests \
--spot-instance-request-ids sir-08b93456
```

Tagging Spot Instance requests

To help categorize and manage your Spot Instance requests, you can tag them with custom metadata. You can assign a tag to a Spot Instance request when you create it, or afterward. You can assign tags using the Amazon EC2 console or a command line tool.

When you tag a Spot Instance request, the instances and volumes that are launched by the Spot Instance request are not automatically tagged. You need to explicitly tag the instances and volumes launched by the Spot Instance request. You can assign a tag to a Spot Instance and volumes during launch, or afterward.

For more information about how tags work, see [Tagging your Amazon EC2 resources \(p. 1229\)](#).

Contents

- [Prerequisites \(p. 375\)](#)
- [Tagging a new Spot Instance request \(p. 377\)](#)
- [Tagging an existing Spot Instance request \(p. 378\)](#)
- [Viewing Spot Instance request tags \(p. 378\)](#)

Prerequisites

Grant the IAM user the permission to tag resources. For more information about IAM policies and example policies, see [Example: Tagging resources \(p. 958\)](#).

The IAM policy you create is determined by which method you use for creating a Spot Instance request.

- If you use the launch instance wizard or `run-instances` to request Spot Instances, see [To grant an IAM user the permission to tag resources when using the launch instance wizard or run-instances](#).

- If you use the Spot console to request Spot Instances with a defined duration or use the `request-spot-instances` command to request Spot Instances, see [To grant an IAM user the permission to tag resources when using request-spot-instances](#).

To grant an IAM user the permission to tag resources when using the launch instance wizard or run-instances

Create a IAM policy that includes the following:

- The `ec2:RunInstances` action. This grants the IAM user permission to launch an instance.
- For Resource, specify `spot-instances-request`. This allows users to create Spot Instance requests, which request Spot Instances.
- The `ec2:CreateTags` action. This grants the IAM user permission to create tags.
- For Resource, specify `*`. This allows users to tag all resources that are created during instance launch.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowLaunchInstances",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances"  
            ],  
            "Resource": [  
                "arn:aws:ec2:us-east-1::image/*",  
                "arn:aws:ec2:us-east-1::subnet/*",  
                "arn:aws:ec2:us-east-1::network-interface/*",  
                "arn:aws:ec2:us-east-1::security-group/*",  
                "arn:aws:ec2:us-east-1::key-pair/*",  
                "arn:aws:ec2:us-east-1::volume/*",  
                "arn:aws:ec2:us-east-1::instance/*",  
                "arn:aws:ec2:us-east-1::spot-instances-request/*"  
            ]  
        },  
        {  
            "Sid": "TagSpotInstanceRequests",  
            "Effect": "Allow",  
            "Action": "ec2:CreateTags",  
            "Resource": "*"  
        }  
    ]  
}
```

Note

When you use the `RunInstances` action to create Spot Instance requests and tag the Spot Instance requests on create, you need to be aware of how Amazon EC2 evaluates the `spot-instances-request` resource in the `RunInstances` statement.

The `spot-instances-request` resource is evaluated in the IAM policy as follows:

- If you don't tag a Spot Instance request on create, Amazon EC2 does not evaluate the `spot-instances-request` resource in the `RunInstances` statement.
- If you tag a Spot Instance request on create, Amazon EC2 evaluates the `spot-instances-request` resource in the `RunInstances` statement.

Therefore, for the `spot-instances-request` resource, the following rules apply to the IAM policy:

- If you use RunInstances to create a Spot Instance request and you don't intend to tag the Spot Instance request on create, you don't need to explicitly allow the `spot-instances-request` resource; the call will succeed.
- If you use RunInstances to create a Spot Instance request and intend to tag the Spot Instance request on create, you must include the `spot-instances-request` resource in the RunInstances allow statement, otherwise the call will fail.
- If you use RunInstances to create a Spot Instance request and intend to tag the Spot Instance request on create, you must specify the `spot-instances-request` resource or include a * wildcard in the CreateTags allow statement, otherwise the call will fail.

For example IAM policies, including policies that are not supported for Spot Instance requests, see [Working with Spot Instances \(p. 952\)](#).

To grant an IAM user the permission to tag resources when using request-spot-instances

Create a IAM policy that includes the following:

- The `ec2:RequestSpotInstances` action. This grants the IAM user permission to create a Spot Instance request.
- The `ec2:CreateTags` action. This grants the IAM user permission to create tags.
- For Resource, specify `spot-instances-request`. This allows users to tag only the Spot Instance request.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "TagSpotInstanceRequest",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RequestSpotInstances",  
                "ec2:CreateTags"  
            ],  
            "Resource": "arn:aws:ec2:us-east-1:111122223333:spot-instances-request/*"  
        }  
    ]  
}
```

Tagging a new Spot Instance request

To tag a new Spot Instance request using the console

1. Follow the [Creating a Spot Instance request \(p. 371\)](#) procedure.
2. To add a tag, on the **Add Tags** page, choose **Add Tag**, and enter the key and value for the tag. Choose **Add another tag** for each additional tag.
For each tag, you can tag the Spot Instance request, the Spot Instances, and the volumes with the same tag. To tag all three, ensure that **Instances**, **Volumes**, and **Spot Instance Requests** are selected. To tag only one or two, ensure that the resources you want to tag are selected, and the other resources are cleared.
3. Complete the required fields to create a Spot Instance request, and then choose **Launch**. For more information, see [Creating a Spot Instance request \(p. 371\)](#).

To tag a new Spot Instance request using the AWS CLI

To tag a Spot Instance request when you create it, configure the Spot Instance request configuration as follows:

- Specify the tags for the Spot Instance request using the `--tag-specification` parameter.
- For `ResourceType`, specify `spot-instances-request`. If you specify another value, the Spot Instance request will fail.
- For `Tags`, specify the key-value pair. You can specify more than one key-value pair.

In the following example, the Spot Instance request is tagged with two tags: Key=Environment and Value=Production, and Key=Cost-Center and Value=123.

```
aws ec2 request-spot-instances \
  --instance-count 5 \
  --type "one-time" \
  --launch-specification file://specification.json \
  --tag-specification 'ResourceType=spot-instances-
request,Tags=[{Key=Environment,Value=Production},{Key=Cost-Center,Value=123}]'
```

Tagging an existing Spot Instance request

To tag an existing Spot Instance request using the console

After you have created a Spot Instance request, you can add tags to the Spot Instance request using the console.

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot>.
2. Select your Spot Instance request.
3. Choose the **Tags** tab and choose **Create Tag**.

To tag an existing Spot Instance using the console

After your Spot Instance request has launched your Spot Instance, you can add tags to the instance using the console. For more information, see [Adding and deleting tags on an individual resource \(p. 1235\)](#).

To tag an existing Spot Instance request or Spot Instance using the AWS CLI

Use the `create-tags` command to tag existing resources. In the following example, the existing Spot Instance request and the Spot Instance are tagged with Key=purpose and Value=test.

```
aws ec2 create-tags \
  --resources sir-08b93456 i-1234567890abcdef0 \
  --tags Key=purpose,Value=test
```

Viewing Spot Instance request tags

To view Spot Instance request tags using the console

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot>.
2. Select your Spot Instance request and choose the **Tags** tab.

To describe Spot Instance request tags

Use the `describe-tags` command to view the tags for the specified resource. In the following example, you describe the tags for the specified request.

```
aws ec2 describe-tags \
  --filters "Name=resource-id,Values=sir-11112222-3333-4444-5555-66666EXAMPLE"
```

```
{  
    "Tags": [  
        {  
            "Key": "Environment",  
            "ResourceId": "sir-11112222-3333-4444-5555-66666EXAMPLE",  
            "ResourceType": "spot-instances-request",  
            "Value": "Production"  
        },  
        {  
            "Key": "Another key",  
            "ResourceId": "sir-11112222-3333-4444-5555-66666EXAMPLE",  
            "ResourceType": "spot-instances-request",  
            "Value": "Another value"  
        }  
    ]  
}
```

You can also view the tags of a Spot Instance request by describing the Spot Instance request.

Use the [describe-spot-instance-requests](#) command to view the configuration of the specified Spot Instance request, which includes any tags that were specified for the request.

```
aws ec2 describe-spot-instance-requests \  
--spot-instance-request-ids sir-11112222-3333-4444-5555-66666EXAMPLE
```

```
{  
    "SpotInstanceRequests": [  
        {  
            "CreateTime": "2020-06-24T14:22:11+00:00",  
            "InstanceId": "i-1234567890EXAMPLE",  
            "LaunchSpecification": {  
                "SecurityGroups": [  
                    {  
                        "GroupName": "launch-wizard-6",  
                        "GroupId": "sg-1234567890EXAMPLE"  
                    }  
                ],  
                "BlockDeviceMappings": [  
                    {  
                        "DeviceName": "/dev/xvda",  
                        "Ebs": {  
                            "DeleteOnTermination": true,  
                            "VolumeSize": 8,  
                            "VolumeType": "gp2"  
                        }  
                    }  
                ],  
                "ImageId": "ami-1234567890EXAMPLE",  
                "InstanceType": "t2.micro",  
                "KeyName": "my-key-pair",  
                "NetworkInterfaces": [  
                    {  
                        "DeleteOnTermination": true,  
                        "DeviceIndex": 0,  
                        "SubnetId": "subnet-11122233"  
                    }  
                ],  
                "Placement": {  
                    "AvailabilityZone": "eu-west-1c",  
                    "Tenancy": "default"  
                },  
                "Monitoring": {  
                    "Enabled": false  
                }  
            }  
        }  
    ]  
}
```

```
        "Enabled": false
    },
},
"LaunchedAvailabilityZone": "eu-west-1c",
"ProductDescription": "Linux/UNIX",
"SpotInstanceRequestId": "sir-1234567890EXAMPLE",
"SpotPrice": "0.012600",
"State": "active",
"Status": {
    "Code": "fulfilled",
    "Message": "Your spot request is fulfilled.",
    "UpdateTime": "2020-06-25T18:30:21+00:00"
},
"Tags": [
    {
        "Key": "Environment",
        "Value": "Production"
    },
    {
        "Key": "Another key",
        "Value": "Another value"
    }
],
>Type": "one-time",
"InstanceInterruptionBehavior": "terminate"
}
]
}
```

Canceling a Spot Instance request

If you no longer want your Spot Instance request, you can cancel it. You can only cancel Spot Instance requests that are open, active, or disabled.

- Your Spot Instance request is **open** when your request has not yet been fulfilled and no instances have been launched.
- Your Spot Instance request is **active** when your request has been fulfilled and Spot Instances have launched as a result.
- Your Spot Instance request is **disabled** when you stop your Spot Instance.

If your Spot Instance request is active and has an associated running Spot Instance, canceling the request does not terminate the instance. For more information about terminating a Spot Instance, see [Terminating a Spot Instance \(p. 382\)](#).

To cancel a Spot Instance request (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests** and select the Spot request.
3. Choose **Actions, Cancel request**.
4. (Optional) If you are finished with the associated Spot Instances, you can terminate them. In the **Cancel Spot request** dialog box, select **Terminate instances**, and then choose **Confirm**.

To cancel a Spot Instance request (AWS CLI)

- Use the `cancel-spot-instance-requests` command to cancel the specified Spot request.

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

Stopping a Spot Instance

If you don't need your Spot Instances now, but you want to restart them later without losing the data persisted in the Amazon EBS volume, you can stop them. The steps for stopping a Spot Instance are similar to the steps for stopping an On-Demand Instance. You can only stop a Spot Instance if the Spot Instance was launched from a persistent Spot Instance request.

Note

While a Spot Instance is stopped, you can modify some of its instance attributes, but not the instance type.

We don't charge usage for a stopped Spot Instance, or data transfer fees, but we do charge for the storage for any Amazon EBS volumes.

Limitations

- You can't stop a Spot Instance if it is part of a fleet or launch group, Availability Zone group, or Spot block.

New console

To stop a Spot Instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select the Spot Instance.
3. Choose **Actions, Instance state, Stop instance**.
4. In the confirmation window, choose **Stop**.

Old console

To stop a Spot Instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select the Spot Instance.
3. Choose **Actions, Instance State, Stop**.

AWS CLI

To stop a Spot Instance (AWS CLI)

- Use the `stop-instances` command to manually stop one or more Spot Instances.

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

Starting a Spot Instance

You can start a Spot Instance that you previously stopped. The steps for starting a Spot Instance are similar to the steps for starting an On-Demand Instance.

Prerequisites

You can only start a Spot Instance if:

- You manually stopped the Spot Instance.

- The Spot Instance is an EBS-backed instance.
- Spot Instance capacity is available.
- The Spot price is lower than your maximum price.

Limitations

- You can't start a Spot Instance if it is part of fleet or launch group, Availability Zone group, or Spot block.

New console

To start a Spot Instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select the Spot Instance.
3. Choose **Actions, Instance state, Start instance**.

Old console

To start a Spot Instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select the Spot Instance.
3. Choose **Actions, Instance State, Start**.

AWS CLI

To start a Spot Instance (AWS CLI)

- Use the `start-instances` command to manually start one or more Spot Instances.

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

Terminating a Spot Instance

If your Spot Instance request is active and has an associated running Spot Instance, or your Spot Instance request is disabled and has an associated stopped Spot Instance, canceling the request does not terminate the instance; you must terminate the running Spot Instance manually.

If you terminate a running or stopped Spot Instance that was launched by a persistent Spot request, the Spot request returns to the open state so that a new Spot Instance can be launched. To cancel a persistent Spot request and terminate its Spot Instances, you must cancel the Spot request first and then terminate the Spot Instances. Otherwise, the persistent Spot request can launch a new instance. For more information about canceling a Spot Instance request, see [Canceling a Spot Instance request \(p. 380\)](#).

New console

To manually terminate a Spot Instance using the console

1. Before you terminate an instance, verify that you won't lose any data by checking that your Amazon EBS volumes won't be deleted on termination and that you've copied any data that you

need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.

2. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
3. In the navigation pane, choose **Instances**.
4. To confirm that the instance is a Spot Instance, check that **spot** appears in the **Instance lifecycle** column.
5. Select the instance, and choose **Actions, Instance state, Terminate instance**.
6. Choose **Terminate** when prompted for confirmation.

Old console

To manually terminate a Spot Instance using the console

1. Before you terminate an instance, verify that you won't lose any data by checking that your Amazon EBS volumes won't be deleted on termination and that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.
2. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
3. In the navigation pane, choose **Instances**.
4. To confirm that the instance is a Spot Instance, check that **spot** appears in the **Lifecycle** column.
5. Select the instance, and choose **Actions, Instance State, Terminate**.
6. Choose **Yes, Terminate** when prompted for confirmation.

AWS CLI

To manually terminate a Spot Instance using the AWS CLI

- Use the `terminate-instances` command to manually terminate Spot Instances.

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7
```

Spot Instance request example launch specifications

The following examples show launch configurations that you can use with the `request-spot-instances` command to create a Spot Instance request. For more information, see [Creating a Spot Instance request \(p. 371\)](#).

1. [Launch Spot Instances \(p. 383\)](#)
2. [Launch Spot Instances in the specified Availability Zone \(p. 384\)](#)
3. [Launch Spot Instances in the specified subnet \(p. 384\)](#)
4. [Launch a Dedicated Spot Instance \(p. 385\)](#)

Example 1: Launch Spot Instances

The following example does not include an Availability Zone or subnet. Amazon EC2 selects an Availability Zone for you. Amazon EC2 launches the instances in the default subnet of the selected Availability Zone.

```
{  
    "ImageId": "ami-1a2b3c4d",  
    "KeyName": "my-key-pair",
```

```
"SecurityGroupIds": [ "sg-1a2b3c4d" ],
"InstanceType": "m3.medium",
"IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
}
}
```

Example 2: Launch Spot Instances in the specified Availability Zone

The following example includes an Availability Zone. Amazon EC2 launches the instances in the default subnet of the specified Availability Zone.

```
{
    "ImageId": "ami-1a2b3c4d",
    "KeyName": "my-key-pair",
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],
    "InstanceType": "m3.medium",
    "Placement": {
        "AvailabilityZone": "us-west-2a"
    },
    "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
    }
}
```

Example 3: Launch Spot Instances in the specified subnet

The following example includes a subnet. Amazon EC2 launches the instances in the specified subnet. If the VPC is a nondefault VPC, the instance does not receive a public IPv4 address by default.

```
{
    "ImageId": "ami-1a2b3c4d",
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],
    "InstanceType": "m3.medium",
    "SubnetId": "subnet-1a2b3c4d",
    "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
    }
}
```

To assign a public IPv4 address to an instance in a nondefault VPC, specify the `AssociatePublicIpAddress` field as shown in the following example. When you specify a network interface, you must include the subnet ID and security group ID using the network interface, rather than using the `SubnetId` and `SecurityGroupIds` fields shown in example 3.

```
{
    "ImageId": "ami-1a2b3c4d",
    "KeyName": "my-key-pair",
    "InstanceType": "m3.medium",
    "NetworkInterfaces": [
        {
            "DeviceIndex": 0,
            "SubnetId": "subnet-1a2b3c4d",
            "Groups": [ "sg-1a2b3c4d" ],
            "AssociatePublicIpAddress": true
        }
    ],
    "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
    }
}
```

Example 4: Launch a Dedicated Spot Instance

The following example requests Spot Instance with a tenancy of dedicated. A Dedicated Spot Instance must be launched in a VPC.

```
{  
    "ImageId": "ami-1a2b3c4d",  
    "KeyName": "my-key-pair",  
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],  
    "InstanceType": "c3.8xlarge",  
    "SubnetId": "subnet-1a2b3c4d",  
    "Placement": {  
        "Tenancy": "dedicated"  
    }  
}
```

Spot Fleet requests

To use a Spot Fleet, you create a Spot Fleet request that includes the target capacity, an optional On-Demand portion, one or more launch specifications for the instances, and the maximum price that you are willing to pay. Amazon EC2 attempts to maintain your Spot Fleet's target capacity as Spot prices change. For more information, see [How Spot Fleet works \(p. 358\)](#).

There are two types of Spot Fleet requests: `request` and `maintain`. You can create a Spot Fleet to submit a one-time request for your desired capacity, or require it to maintain a target capacity over time. Both types of requests benefit from Spot Fleet's allocation strategy.

When you make a one-time request, Spot Fleet places the required requests but does not attempt to replenish Spot Instances if capacity is diminished. If capacity is not available, Spot Fleet does not submit requests in alternative Spot pools.

To maintain a target capacity, Spot Fleet places requests to meet the target capacity and automatically replenish any interrupted instances.

It is not possible to modify the target capacity of a one-time request after it's been submitted. To change the target capacity, cancel the request and submit a new one.

A Spot Fleet request remains active until it expires or you cancel it. When you cancel a Spot Fleet request, you may specify whether canceling your Spot Fleet request terminates the Spot Instances in your Spot Fleet.

Each launch specification includes the information that Amazon EC2 needs to launch an instance, such as an AMI, instance type, subnet or Availability Zone, and one or more security groups.

Contents

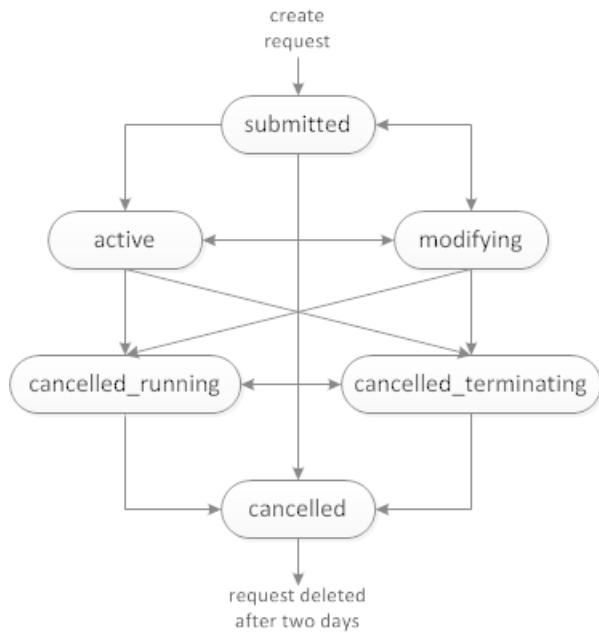
- [Spot Fleet request states \(p. 386\)](#)
- [Spot Fleet health checks \(p. 386\)](#)
- [Planning a Spot Fleet request \(p. 387\)](#)
- [Spot Fleet permissions \(p. 387\)](#)
- [Creating a Spot Fleet request \(p. 391\)](#)
- [Tagging a Spot Fleet \(p. 395\)](#)
- [Monitoring your Spot Fleet \(p. 401\)](#)
- [Modifying a Spot Fleet request \(p. 401\)](#)
- [Canceling a Spot Fleet request \(p. 402\)](#)
- [Spot Fleet example configurations \(p. 403\)](#)

Spot Fleet request states

A Spot Fleet request can be in one of the following states:

- submitted – The Spot Fleet request is being evaluated and Amazon EC2 is preparing to launch the target number of instances.
 - active – The Spot Fleet has been validated and Amazon EC2 is attempting to maintain the target number of running Spot Instances. The request remains in this state until it is modified or canceled.
 - modifying – The Spot Fleet request is being modified. The request remains in this state until the modification is fully processed or the Spot Fleet is canceled. A one-time request cannot be modified, and this state does not apply to such Spot requests.
 - cancelled_running – The Spot Fleet is canceled and does not launch additional Spot Instances. Its existing Spot Instances continue to run until they are interrupted or terminated. The request remains in this state until all instances are interrupted or terminated.
 - cancelled_terminating – The Spot Fleet is canceled and its Spot Instances are terminating. The request remains in this state until all instances are terminated.
 - cancelled – The Spot Fleet is canceled and has no running Spot Instances. The Spot Fleet request is deleted two days after its instances were terminated.

The following illustration represents the transitions between the request states. If you exceed your Spot Fleet limits, the request is canceled immediately.



Spot Fleet health checks

Spot Fleet checks the health status of the Spot Instances in the fleet every two minutes. The health status of an instance is either healthy or unhealthy. Spot Fleet determines the health status of an instance using the status checks provided by Amazon EC2. If the status of either the instance status check or the system status check is impaired for three consecutive health checks, the health status of the instance is unhealthy. Otherwise, the health status is healthy. For more information, see [Status checks for your instances \(p. 697\)](#).

You can configure your Spot Fleet to replace unhealthy instances. After enabling health check replacement, an instance is replaced after its health status is reported as unhealthy. The Spot Fleet could go below its target capacity for up to a few minutes while an unhealthy instance is being replaced.

Requirements

- Health check replacement is supported only with Spot Fleets that maintain a target capacity, not with one-time Spot Fleets.
- You can configure your Spot Fleet to replace unhealthy instances only when you create it.
- IAM users can use health check replacement only if they have permission to call the `ec2:DescribeInstanceStatus` action.

Planning a Spot Fleet request

Before you create a Spot Fleet request, review [Spot Best Practices](#). Use these best practices when you plan your Spot Fleet request so that you can provision the type of instances you want at the lowest possible price. We also recommend that you do the following:

- Determine whether you want to create a Spot Fleet that submits a one-time request for the desired target capacity, or one that maintains a target capacity over time.
- Determine the instance types that meet your application requirements.
- Determine the target capacity for your Spot Fleet request. You can set the target capacity in instances or in custom units. For more information, see [Spot Fleet instance weighting \(p. 361\)](#).
- Determine what portion of the Spot Fleet target capacity must be On-Demand capacity. You can specify 0 for On-Demand capacity.
- Determine your price per unit, if you are using instance weighting. To calculate the price per unit, divide the price per instance hour by the number of units (or weight) that this instance represents. If you are not using instance weighting, the default price per unit is the price per instance hour.
- Review the possible options for your Spot Fleet request. For more information, see the `request-spot-fleet` command in the *AWS CLI Command Reference*. For additional examples, see [Spot Fleet example configurations \(p. 403\)](#).

Spot Fleet permissions

If your IAM users will create or manage a Spot Fleet, you need to grant them the required permissions.

If you use the Amazon EC2 console to create a Spot Fleet, it creates a service-linked role named `AWSServiceRoleForEC2SpotFleet` and a role named `aws-ec2-spot-fleet-tagging-role` that grant the Spot Fleet the permissions to request, launch, terminate, and tag resources on your behalf. If you use the AWS CLI or an API, you must ensure that these roles exist.

Use the following instructions to grant the required permissions and create the roles.

Permissions and roles

- [Granting permission to IAM users for Spot Fleet \(p. 387\)](#)
- [Service-linked role for Spot Fleet \(p. 389\)](#)
- [IAM role for Spot Fleet \(p. 391\)](#)

Granting permission to IAM users for Spot Fleet

If your IAM users will create or manage a Spot Fleet, be sure to grant them the required permissions as follows.

To grant an IAM user permissions for Spot Fleet

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.

2. In the navigation pane, choose **Policies, Create policy**.
3. On the **Create policy** page, choose **JSON**, and replace the text with the following.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances",  
                "ec2:CreateTags",  
                "ec2:RequestSpotFleet",  
                "ec2:ModifySpotFleetRequest",  
                "ec2:CancelSpotFleetRequests",  
                "ec2:DescribeSpotFleetRequests",  
                "ec2:DescribeSpotFleetInstances",  
                "ec2:DescribeSpotFleetRequestHistory"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "iam:PassRole",  
            "Resource": "arn:aws:iam::*:role/aws-ec2-spot-fleet-tagging-role"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iam:CreateServiceLinkedRole",  
                "iam>ListRoles",  
                "iam>ListInstanceProfiles"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

The preceding example policy grants an IAM user the permissions required for most Spot Fleet use cases. To limit the user to specific API actions, specify only those API actions instead.

Required EC2 and IAM APIs

The following APIs must be included in the policy:

- **ec2:RunInstances** – Required to launch instances in a Spot Fleet
- **ec2:CreateTags** – Required to tag the Spot Fleet request, instances, or volumes
- **iam:PassRole** – Required to specify the Spot Fleet role
- **iam:CreateServiceLinkedRole** – Required to create the service-linked role
- **iam>ListRoles** – Required to enumerate existing IAM roles
- **iam>ListInstanceProfiles** – Required to enumerate existing instance profiles

Important

If you specify a role for the IAM instance profile in the launch specification or launch template, you must grant the IAM user the permission to pass the role to the service. To do this, in the IAM policy include "arn:aws:iam::*:role/*IamInstanceProfile-role*" as a resource for the **iam:PassRole** action. For more information, see [Granting a User Permissions to Pass a Role to an AWS Service](#) in the *IAM User Guide*.

Spot Fleet APIs

Add the following Spot Fleet API actions to your policy, as needed:

- `ec2:RequestSpotFleet`
- `ec2:ModifySpotFleetRequest`
- `ec2:CancelSpotFleetRequests`
- `ec2:DescribeSpotFleetRequests`
- `ec2:DescribeSpotFleetInstances`
- `ec2:DescribeSpotFleetRequestHistory`

Optional IAM APIs

(Optional) To enable an IAM user to create roles or instance profiles using the IAM console, you must add the following actions to the policy:

- `iam:AddRoleToInstanceProfile`
- `iam:AttachRolePolicy`
- `iam>CreateInstanceProfile`
- `iam:CreateRole`
- `iam:GetRole`
- `iam>ListPolicies`

4. Choose **Review policy**.
5. On the **Review policy** page, enter a policy name and description, and choose **Create policy**.
6. In the navigation pane, choose **Users** and select the user.
7. Choose **Permissions, Add permissions**.
8. Choose **Attach existing policies directly**. Select the policy that you created earlier and choose **Next: Review**.
9. Choose **Add permissions**.

Service-linked role for Spot Fleet

Amazon EC2 uses service-linked roles for the permissions that it requires to call other AWS services on your behalf. A service-linked role is a unique type of IAM role that is linked directly to an AWS service. Service-linked roles provide a secure way to delegate permissions to AWS services because only the linked service can assume a service-linked role. For more information, see [Using Service-Linked Roles](#) in the *IAM User Guide*.

Amazon EC2 uses the service-linked role named **AWSServiceRoleForEC2SpotFleet** to launch and manage instances on your behalf.

Important

If you specify an [encrypted AMI \(p. 157\)](#) or an [encrypted Amazon EBS snapshot \(p. 1121\)](#) in your Spot Fleet, you must grant the **AWSServiceRoleForEC2SpotFleet** role permission to use the CMK so that Amazon EC2 can launch instances on your behalf. For more information, see [Granting access to CMKs for use with encrypted AMIs and EBS snapshots \(p. 390\)](#).

Permissions granted by AWSServiceRoleForEC2SpotFleet

Amazon EC2 uses **AWSServiceRoleForEC2SpotFleet** to complete the following actions:

- `ec2:RequestSpotInstances` - Request Spot Instances
- `ec2:RunInstances` - Launch instances

- `ec2:TerminateInstances` - Terminate instances
- `ec2:DescribeImages` - Describe Amazon Machine Images (AMIs) for the instances
- `ec2:DescribeInstanceStatus` - Describe the status of the instances
- `ec2:DescribeSubnets` - Describe the subnets for the instances
- `ec2:CreateTags` - Add tags to the Spot Fleet request, instances, and volumes
- `elasticloadbalancing:RegisterInstancesWithLoadBalancer` - Add the specified instances to the specified load balancer
- `elasticloadbalancing:RegisterTargets` - Register the specified targets with the specified target group

Creating the service-linked role

Under most circumstances, you don't need to manually create a service-linked role. Amazon EC2 creates the **AWSServiceRoleForEC2SpotFleet** service-linked role the first time you create a Spot Fleet using the console.

If you use the AWS CLI or an API, you must ensure that this role exists.

If you had an active Spot Fleet request before October 2017, when Amazon EC2 began supporting this service-linked role, Amazon EC2 created the **AWSServiceRoleForEC2SpotFleet** role in your AWS account. For more information, see [A New Role Appeared in My AWS Account](#) in the *IAM User Guide*.

Ensure that this role exists before you use the AWS CLI or an API to create a Spot Fleet. To create the role, use the IAM console as follows.

To manually create the **AWSServiceRoleForEC2SpotFleet** service-linked role

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**.
3. Choose **Create role**.
4. For **Select type of trusted entity**, choose **AWS service**.
5. In the list of services, choose **EC2**.
6. In the **Select your use case** section, choose **EC2 - Spot Fleet**
7. Choose **Next: Permissions**.
8. On the next page, choose **Next:Review**.
9. On the **Review** page, choose **Create role**.

If you no longer need to use Spot Fleet, we recommend that you delete the **AWSServiceRoleForEC2SpotFleet** role. After this role is deleted from your account, Amazon EC2 will create the role again if you request a Spot Fleet using the console. For more information, see [Deleting a Service-Linked Role](#) in the *IAM User Guide*.

Granting access to CMKs for use with encrypted AMIs and EBS snapshots

If you specify an [encrypted AMI \(p. 157\)](#) or an [encrypted Amazon EBS snapshot \(p. 1121\)](#) in your Spot Fleet request and you use a customer managed customer master key (CMK) for encryption, you must grant the **AWSServiceRoleForEC2SpotFleet** role permission to use the CMK so that Amazon EC2 can launch instances on your behalf. To do this, you must add a grant to the CMK, as shown in the following procedure.

When providing permissions, grants are an alternative to key policies. For more information, see [Using Grants](#) and [Using Key Policies in AWS KMS](#) in the *AWS Key Management Service Developer Guide*.

To grant the **AWSServiceRoleForEC2SpotFleet** role permissions to use the CMK

- Use the [create-grant](#) command to add a grant to the CMK and to specify the principal (the **AWSServiceRoleForEC2SpotFleet** service-linked role) that is given permission to perform the operations that the grant permits. The CMK is specified by the `key-id` parameter and the ARN of the CMK. The principal is specified by the `grantee-principal` parameter and the ARN of the **AWSServiceRoleForEC2SpotFleet** service-linked role.

```
aws kms create-grant \
    --region us-east-1 \
    --key-id arn:aws:kms:us-
east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2SpotFleet \
    --operations "Decrypt" "Encrypt" "GenerateDataKey"
    "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
    "ReEncryptTo"
```

IAM role for Spot Fleet

The `aws-ec2-spot-fleet-tagging-role` IAM role grants the Spot Fleet permission to tag the Spot Fleet request, instances, and volumes. For more information, see [Tagging a Spot Fleet \(p. 395\)](#).

Important

If you choose to tag instances in the fleet and you choose to maintain target capacity (the Spot Fleet request is of type `maintain`), the differences in permissions of the IAM user and the `IamFleetRole` might lead to inconsistent tagging behavior of instances in the fleet. If the `IamFleetRole` does not include the `CreateTags` permission, some of the instances launched by the fleet might not be tagged. While we are working to fix this inconsistency, to ensure that all instances launched by the fleet are tagged, we recommend that you use the `aws-ec2-spot-fleet-tagging-role` role for the `IamFleetRole`. Alternatively, to use an existing role, attach the `AmazonEC2SpotFleetTaggingRole` AWS Managed Policy to the existing role. Otherwise, you need to manually add the `CreateTags` permission to your existing policy.

To create the IAM role for tagging a Spot Fleet

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**.
3. On the **Select type of trusted entity** page, choose **AWS service, EC2, EC2 - Spot Fleet Tagging**, **Next: Permissions**.
4. On the **Attached permissions policy** page, choose **Next:Review**.
5. On the **Review** page, type a name for the role (for example, `aws-ec2-spot-fleet-tagging-role`) and choose **Create role**.

Creating a Spot Fleet request

Using the AWS Management Console, quickly create a Spot Fleet request by choosing only your application or task need and minimum compute specs. Amazon EC2 configures a fleet that best meets your needs and follows Spot best practice. For more information, see [Quickly create a Spot Fleet request \(console\) \(p. 391\)](#). Otherwise, you can modify any of the default settings. For more information, see [Create a Spot Fleet request using defined parameters \(console\) \(p. 392\)](#).

Quickly create a Spot Fleet request (console)

Follow these steps to quickly create a Spot Fleet request.

To create a Spot Fleet request using the recommended settings (console)

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot>.
2. If you are new to Spot, you see a welcome page; choose **Get started**. Otherwise, choose **Request Spot Instances**.
3. For **Tell us your application or task need**, choose **Load balancing workloads**, **Flexible workloads**, **Big data workloads**, or **Defined duration workloads**.
4. Under **Configure your instances**, for **Minimum compute unit**, choose the minimum hardware specifications (vCPUs, memory, and storage) that you need for your application or task, either **as specs** or **as an instance type**.
 - For **as specs**, specify the required number of vCPUs and amount of memory.
 - For **as an instance type**, accept the default instance type, or choose **Change instance type** to choose a different instance type.
5. Under **Tell us how much capacity you need**, for **Total target capacity**, specify the number of units to request for target capacity. You can choose instances or vCPUs.
6. Review the recommended **Fleet request settings** based on your application or task selection, and choose **Launch**.

Create a Spot Fleet request using defined parameters (console)

You can create a Spot Fleet using the parameters that you define.

To create a Spot Fleet request using defined parameters (console)

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot>.
2. If you are new to Spot, you see a welcome page; choose **Get started**. Otherwise, choose **Request Spot Instances**.
3. For **Tell us your application or task need**, choose **Load balancing workloads**, **Flexible workloads**, **Big data workloads**, or **Defined duration workloads**.
4. For **Configure your instances**, do the following:
 - a. (Optional) For **Launch template**, choose a launch template. The launch template must specify an Amazon Machine Image (AMI), as you cannot override the AMI using Spot Fleet if you specify a launch template.

Important
If you intend to specify **Optional On-Demand portion**, you must choose a launch template.
 - b. For **AMI**, choose one of the basic AMIs provided by AWS, or choose **Search for AMI** to use an AMI from our user community, the AWS Marketplace, or one of your own.
 - c. For **Minimum compute unit**, choose the minimum hardware specifications (vCPUs, memory, and storage) that you need for your application or task, either **as specs** or **as an instance type**.
 - For **as specs**, specify the required number of vCPUs and amount of memory.
 - For **as an instance type**, accept the default instance type, or choose **Change instance type** to choose a different instance type.
 - d. For **Network**, choose an existing VPC or create a new one.

[Existing VPC] Choose the VPC.

[New VPC] Choose **Create new VPC** to go the Amazon VPC console. When you are done, return to the wizard and refresh the list.

- e. (Optional) For **Availability Zone**, let AWS choose the Availability Zones for your Spot Instances, or specify one or more Availability Zones.

If you have more than one subnet in an Availability Zone, choose the appropriate subnet from **Subnet**. To add subnets, choose **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and refresh the list.

- f. (Optional) For **Key pair name**, choose an existing key pair or create a new one.

[Existing key pair] Choose the key pair.

[New key pair] Choose **Create new key pair** to go the Amazon VPC console. When you are done, return to the wizard and refresh the list.

5. (Optional) For **Additional configurations**, do the following:

- a. (Optional) To add storage, specify additional instance store volumes or Amazon EBS volumes, depending on the instance type.
- b. (Optional) To enable Amazon EBS optimization, for **EBS-optimized**, choose **Launch EBS-optimized instances**.
- c. (Optional) To add temporary block-level storage for your instances, for **Instance store**, choose **Attach at launch**.
- d. (Optional) By default, basic monitoring is enabled for your instances. To enable detailed monitoring, for **Monitoring**, choose **Enable CloudWatch detailed monitoring**.
- e. (Optional) To replace unhealthy instances, for **Health check**, choose **Replace unhealthy instances**. To enable this option, you must first choose **Maintain target capacity**.
- f. (Optional) To run a Dedicated Spot Instance, for **Tenancy**, choose **Dedicated - run a dedicated instance**.
- g. (Optional) For **Security groups**, choose one or more security groups or create a new one.

[Existing security group] Choose one or more security groups.

[New security group] Choose **Create new security group** to go the Amazon VPC console. When you are done, return to the wizard and refresh the list.

- h. (Optional) To make your instances reachable from the internet, for **Auto-assign IPv4 Public IP**, choose **Enable**.
- i. (Optional) To launch your Spot Instances with an IAM role, for **IAM instance profile**, choose the role.
- j. (Optional) To run a start-up script, copy it to **User data**.
- k. (Optional) To add a tag, choose **Add new tag** and enter the key and value for the tag. Repeat for each tag.

For each tag, to tag the instances and the Spot Fleet request with the same tag, ensure that both **Instance tags** and **Fleet tags** are selected. To tag only the instances launched by the fleet, clear **Fleet tags**. To tag only the Spot Fleet request, clear **Instance tags**.

6. For **Tell us how much capacity you need**, do the following:

- a. For **Total target capacity**, specify the number of units to request for target capacity. You can choose instances or vCPUs. To specify a target capacity of 0 so that you can add capacity later, choose **Maintain target capacity**.
- b. (Optional) For **Optional On-Demand portion**, specify the number of On-Demand units to request. The number must be less than the **Total target capacity**. Amazon EC2 calculates the difference, and allocates the difference to Spot units to request.

Important

To specify an optional On-Demand portion, you must first choose a launch template.

- c. (Optional) By default, the Spot service terminates Spot Instances when they are interrupted. To maintain the target capacity, select **Maintain target capacity**. You can then specify that the

Spot service terminates, stops, or hibernates Spot Instances when they are interrupted. To do so, choose the corresponding option from **Interruption behavior**.

- d. (Optional) To control the amount you pay per hour for the total Spot Instances in your fleet, select **Maintain target cost for Spot (advanced - optional)** and then enter the maximum total amount you're willing to pay per hour. When the maximum total amount is reached, Spot Fleet stops launching Spot Instances even if it hasn't met the target capacity. For more information, see [Control spending \(p. 360\)](#).
7. For **Fleet request settings**, do the following:
 - a. Review the fleet request and fleet allocation strategy based on your application or task selection. To change the instance types or allocation strategy, clear **Apply recommendations**.
 - b. (Optional) To remove instance types, for **Fleet request**, choose **Remove**. To add instance types, choose **Select instance types**.
 - c. (Optional) For **Fleet allocation strategy**, choose the strategy that meets your needs. For more information, see [Allocation strategy for Spot Instances \(p. 359\)](#).
8. For **Additional request details**, do the following:
 - a. Review the additional request details. To make changes, clear **Apply defaults**.
 - b. (Optional) For **IAM fleet role**, you can use the default role or choose a different role. To use the default role after changing the role, choose **Use default role**.
 - c. (Optional) For **Maximum price**, you can use the default maximum price (the On-Demand price) or specify the maximum price you are willing to pay. If your maximum price is lower than the Spot price for the instance types that you selected, your Spot Instances are not launched.
 - d. (Optional) To create a request that is valid only during a specific time period, edit **Request valid from** and **Request valid until**.
 - e. (Optional) By default, we terminate your Spot Instances when the request expires. To keep them running after your request expires, clear **Terminate the instances when the request expires**.
 - f. (Optional) To register your Spot Instances with a load balancer, choose **Receive traffic from one or more load balancers** and choose one or more Classic Load Balancers or target groups.
9. (Optional) To download a copy of the launch configuration for use with the AWS CLI, choose **JSON config**.
10. Choose **Launch**.

The Spot Fleet request type is `fleet`. When the request is fulfilled, requests of type `instance` are added, where the state is `active` and the status is `fulfilled`.

To create a Spot Fleet request using the AWS CLI

- Use the [request-spot-fleet](#) command to create a Spot Fleet request.

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

For example configuration files, see [Spot Fleet example configurations \(p. 403\)](#).

The following is example output:

```
{  
    "SpotFleetRequestId": "sfr-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE"  
}
```

Tagging a Spot Fleet

To help categorize and manage your Spot Fleet requests, you can tag them with custom metadata. You can assign a tag to a Spot Fleet request when you create it, or afterward. You can assign tags using the Amazon EC2 console or a command line tool.

When you tag a Spot Fleet request, the instances and volumes that are launched by the Spot Fleet are not automatically tagged. You need to explicitly tag the instances and volumes launched by the Spot Fleet. You can choose to assign tags to only the Spot Fleet request, or to only the instances launched by the fleet, or to only the volumes attached to the instances launched by the fleet, or to all three.

Note

Volume tags are only supported for volumes that are attached to On-Demand Instances. You can't tag volumes that are attached to Spot Instances.

For more information about how tags work, see [Tagging your Amazon EC2 resources \(p. 1229\)](#).

Contents

- [Prerequisite \(p. 395\)](#)
- [Tagging a new Spot Fleet \(p. 396\)](#)
- [Tagging a new Spot Fleet and the instances and volumes that it launches \(p. 397\)](#)
- [Tagging an existing Spot Fleet \(p. 399\)](#)
- [Viewing Spot Fleet request tags \(p. 399\)](#)

Prerequisite

Grant the IAM user the permission to tag resources. For more information, see [Example: Tagging resources \(p. 958\)](#).

To grant an IAM user the permission to tag resources

Create a IAM policy that includes the following:

- The `ec2:CreateTags` action. This grants the IAM user permission to create tags.
- The `ec2:RequestSpotFleet` action. This grants the IAM user permission to create a Spot Fleet request.
- For `Resource`, you must specify `"*"`. This allows users to tag all resource types.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "TagSpotFleetRequest",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags",  
                "ec2:RequestSpotFleet"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Important

We currently do not support resource-level permissions for the `spot-fleet-request` resource. If you specify `spot-fleet-request` as a resource, you will get an unauthorized exception when you try to tag the fleet. The following example illustrates how *not* to set the policy.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "ec2:CreateTags",  
        "ec2:RequestSpotFleet"  
    ],  
    "Resource": "arn:aws:ec2:us-east-1:111122223333:spot-fleet-request/*"  
}
```

Tagging a new Spot Fleet

To tag a new Spot Fleet request using the console

1. Follow the [Create a Spot Fleet request using defined parameters \(console\) \(p. 392\)](#) procedure.
2. To add a tag, expand **Additional configurations**, choose **Add new tag**, and enter the key and value for the tag. Repeat for each tag.

For each tag, you can tag the Spot Fleet request and the instances with the same tag. To tag both, ensure that both **Instance tags** and **Fleet tags** are selected. To tag only the Spot Fleet request, clear **Instance tags**. To tag only the instances launched by the fleet, clear **Fleet tags**.

3. Complete the required fields to create a Spot Fleet request, and then choose **Launch**. For more information, see [Create a Spot Fleet request using defined parameters \(console\) \(p. 392\)](#).

To tag a new Spot Fleet request using the AWS CLI

To tag a Spot Fleet request when you create it, configure the Spot Fleet request configuration as follows:

- Specify the tags for the Spot Fleet request in `SpotFleetRequestConfig`.
- For `ResourceType`, specify `spot-fleet-request`. If you specify another value, the fleet request will fail.
- For `Tags`, specify the key-value pair. You can specify more than one key-value pair.

In the following example, the Spot Fleet request is tagged with two tags: `Key=Environment` and `Value=Production`, and `Key=Cost-Center` and `Value=123`.

```
{  
    "SpotFleetRequestConfig": {  
        "AllocationStrategy": "lowestPrice",  
        "ExcessCapacityTerminationPolicy": "default",  
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-role",  
        "LaunchSpecifications": [  
            {  
                "ImageId": "ami-0123456789EXAMPLE",  
                "InstanceType": "c4.large"  
            }  
        ],  
        "SpotPrice": "5",  
        "TargetCapacity": 2,  
        "TerminateInstancesWithExpiration": true,  
        "Type": "maintain",  
        "ReplaceUnhealthyInstances": true,  
        "InstanceInterruptionBehavior": "terminate",  
        "InstancePoolsToUseCount": 1,  
        "TagSpecifications": [  
            {  
                "ResourceType": "spot-fleet-request",  
                "Tags": [  
                    {  
                        "Key": "Environment",  
                        "Value": "Production"  
                    },  
                    {  
                        "Key": "Cost-Center",  
                        "Value": "123"  
                    }  
                ]  
            }  
        ]  
    }  
}
```

```
        "Key": "Environment",
        "Value": "Production"
    },
    {
        "Key": "Cost-Center",
        "Value": "123"
    }
]
}
}
```

Tagging a new Spot Fleet and the instances and volumes that it launches

To tag a new Spot Fleet request and the instances and volumes that it launches using the AWS CLI

To tag a Spot Fleet request when you create it, and to tag the instances and volumes when they are launched by the fleet, configure the Spot Fleet request configuration as follows:

Spot Fleet request tags:

- Specify the tags for the Spot Fleet request in `SpotFleetRequestConfig`.
- For `ResourceType`, specify `spot-fleet-request`. If you specify another value, the fleet request will fail.
- For `Tags`, specify the key-value pair. You can specify more than one key-value pair.

Instance tags:

- Specify the tags for the instances in `LaunchSpecifications`.
- For `ResourceType`, specify `instance`. If you specify another value, the fleet request will fail.
- For `Tags`, specify the key-value pair. You can specify more than one key-value pair.

Alternatively, you can specify the tags for the instance in the [launch template \(p. 506\)](#) that is referenced in the Spot Fleet request.

Volume tags:

- Specify the tags for the volumes in the [launch template \(p. 506\)](#) that is referenced in the Spot Fleet request. Volume tagging in `LaunchSpecifications` is not supported.

In the following example, the Spot Fleet request is tagged with two tags: Key=Environment and Value=Production, and Key=Cost-Center and Value=123. The instances that are launched by the fleet are tagged with one tag (which is the same as one of the tags for the Spot Fleet request): Key=Cost-Center and Value=123.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-role",
        "LaunchSpecifications": [
            {
                "ImageId": "ami-0123456789EXAMPLE",
                "InstanceType": "c4.large",
                "TagSpecifications": [
                    {
                        "Tags": [
                            {
                                "Key": "Environment",
                                "Value": "Production"
                            },
                            {
                                "Key": "Cost-Center",
                                "Value": "123"
                            }
                        ]
                    }
                ]
            }
        ]
    }
}
```

```
"ResourceType": "instance",
"Tags": [
    {
        "Key": "Cost-Center",
        "Value": "123"
    }
]
},
"SpotPrice": "5",
"TargetCapacity": 2,
"TerminateInstancesWithExpiration": true,
>Type": "maintain",
"ReplaceUnhealthyInstances": true,
"InstanceInterruptionBehavior": "terminate",
"InstancePoolsToUseCount": 1,
"TagSpecifications": [
    {
        "ResourceType": "spot-fleet-request",
        "Tags": [
            {
                "Key": "Environment",
                "Value": "Production"
            },
            {
                "Key": "Cost-Center",
                "Value": "123"
            }
        ]
    }
]
}
```

To tag instances launched by a Spot Fleet using the AWS CLI

To tag instances when they are launched by the fleet, you can either specify the tags in the [launch template \(p. 506\)](#) that is referenced in the Spot Fleet request, or you can specify the tags in the Spot Fleet request configuration as follows:

- Specify the tags for the instances in `LaunchSpecifications`.
- For `ResourceType`, specify `instance`. If you specify another value, the fleet request will fail.
- For `Tags`, specify the key-value pair. You can specify more than one key-value pair.

In the following example, the instances that are launched by the fleet are tagged with one tag: `Key=Cost-Center` and `Value=123`.

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-role",
        "LaunchSpecifications": [
            {
                "ImageId": "ami-0123456789EXAMPLE",
                "InstanceType": "c4.large",
                "TagSpecifications": [
                    {
                        "ResourceType": "instance",
                        "Tags": [

```

```
        {
            "Key": "Cost-Center",
            "Value": "123"
        }
    ]
}
],
"SpotPrice": "5",
"TargetCapacity": 2,
"TerminateInstancesWithExpiration": true,
"Type": "maintain",
"ReplaceUnhealthyInstances": true,
"InstanceInterruptionBehavior": "terminate",
"InstancePoolsToUseCount": 1
}
}
```

To tag volumes attached to On-Demand Instances launched by a Spot Fleet using the AWS CLI

To tag volumes when they are created by the fleet, you must specify the tags in the [launch template \(p. 506\)](#) that is referenced in the Spot Fleet request.

Note

Volume tags are only supported for volumes that are attached to On-Demand Instances. You can't tag volumes that are attached to Spot Instances.

Volume tagging in [LaunchSpecifications](#) is not supported.

Tagging an existing Spot Fleet

To tag an existing Spot Fleet request using the console

After you have created a Spot Fleet request, you can add tags to the fleet request using the console.

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot>.
2. Select your Spot Fleet request.
3. Choose the **Tags** tab and choose **Create Tag**.

To tag an existing Spot Fleet request using the AWS CLI

You can use the [create-tags](#) command to tag existing resources. In the following example, the existing Spot Fleet request is tagged with Key=purpose and Value=test.

```
aws ec2 create-tags \
--resources sfr-11112222-3333-4444-5555-66666EXAMPLE \
--tags Key=purpose,Value=test
```

Viewing Spot Fleet request tags

To view Spot Fleet request tags using the console

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot>.
2. Select your Spot Fleet request and choose the **Tags** tab.

To describe Spot Fleet request tags

Use the [describe-tags](#) command to view the tags for the specified resource. In the following example, you describe the tags for the specified Spot Fleet request.

```
aws ec2 describe-tags \
--filters "Name=resource-id,Values=sfr-11112222-3333-4444-5555-66666EXAMPLE"
```

```
{
    "Tags": [
        {
            "Key": "Environment",
            "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "ResourceType": "spot-fleet-request",
            "Value": "Production"
        },
        {
            "Key": "Another key",
            "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "ResourceType": "spot-fleet-request",
            "Value": "Another value"
        }
    ]
}
```

You can also view the tags of a Spot Fleet request by describing the Spot Fleet request.

Use the [describe-spot-fleet-requests](#) command to view the configuration of the specified Spot Fleet request, which includes any tags that were specified for the fleet request.

```
aws ec2 describe-spot-fleet-requests \
--spot-fleet-request-ids sfr-11112222-3333-4444-5555-66666EXAMPLE
```

```
{
    "SpotFleetRequestConfigs": [
        {
            "ActivityStatus": "fulfilled",
            "CreateTime": "2020-02-13T02:49:19.709Z",
            "SpotFleetRequestConfig": {
                "AllocationStrategy": "capacityOptimized",
                "OnDemandAllocationStrategy": "lowestPrice",
                "ExcessCapacityTerminationPolicy": "Default",
                "FulfilledCapacity": 2.0,
                "OnDemandFulfilledCapacity": 0.0,
                "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-role",
                "LaunchSpecifications": [
                    {
                        "ImageId": "ami-0123456789EXAMPLE",
                        "InstanceType": "c4.large"
                    }
                ],
                "TargetCapacity": 2,
                "OnDemandTargetCapacity": 0,
                "Type": "maintain",
                "ReplaceUnhealthyInstances": false,
                "InstanceInterruptionBehavior": "terminate"
            },
            "SpotFleetRequestId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "SpotFleetRequestState": "active",
            "Tags": [
                {
                    "Key": "Environment",
                    "Value": "Production"
                },
                {
                    "Key": "Another key",

```

```
        "Value": "Another value"
    }
}
}
```

Monitoring your Spot Fleet

The Spot Fleet launches Spot Instances when your maximum price exceeds the Spot price and capacity is available. The Spot Instances run until they are interrupted or you terminate them.

To monitor your Spot Fleet (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request. To see the configuration details, choose **Description**.
4. To list the Spot Instances for the Spot Fleet, choose **Instances**.
5. To view the history for the Spot Fleet, choose **History**.

To monitor your Spot Fleet (AWS CLI)

Use the [describe-spot-fleet-requests](#) command to describe your Spot Fleet requests.

```
aws ec2 describe-spot-fleet-requests
```

Use the [describe-spot-fleet-instances](#) command to describe the Spot Instances for the specified Spot Fleet.

```
aws ec2 describe-spot-fleet-instances \
--spot-fleet-request-id sfr-73fb2ce-aa30-494c-8788-1cee4EXAMPLE
```

Use the [describe-spot-fleet-request-history](#) command to describe the history for the specified Spot Fleet request.

```
aws ec2 describe-spot-fleet-request-history \
--spot-fleet-request-id sfr-73fb2ce-aa30-494c-8788-1cee4EXAMPLE \
--start-time 2015-05-18T00:00:00Z
```

Modifying a Spot Fleet request

You can modify an active Spot Fleet request to complete the following tasks:

- Increase the target capacity and On-Demand portion
- Decrease the target capacity and On-Demand portion

Note

You can't modify a one-time Spot Fleet request. You can only modify a Spot Fleet request if you selected **Maintain target capacity** when you created the Spot Fleet request.

When you increase the target capacity, the Spot Fleet launches additional Spot Instances. When you increase the On-Demand portion, the Spot Fleet launches additional On-Demand Instances.

When you increase the target capacity, the Spot Fleet launches the additional Spot Instances according to the allocation strategy for its Spot Fleet request. If the allocation strategy is `lowestPrice`, the Spot

Fleet launches the instances from the lowest-priced Spot Instance pool in the Spot Fleet request. If the allocation strategy is diversified, the Spot Fleet distributes the instances across the pools in the Spot Fleet request.

When you decrease the target capacity, the Spot Fleet cancels any open requests that exceed the new target capacity. You can request that the Spot Fleet terminate Spot Instances until the size of the fleet reaches the new target capacity. If the allocation strategy is `lowestPrice`, the Spot Fleet terminates the instances with the highest price per unit. If the allocation strategy is diversified, the Spot Fleet terminates instances across the pools. Alternatively, you can request that the Spot Fleet keep the fleet at its current size, but not replace any Spot Instances that are interrupted or that you terminate manually.

When a Spot Fleet terminates an instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

To modify a Spot Fleet request (console)

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot/home/fleet>.
2. Select your Spot Fleet request.
3. Choose **Actions, Modify target capacity**.
4. In **Modify target capacity**, do the following:
 - a. Enter the new target capacity and On-Demand portion.
 - b. (Optional) If you are decreasing the target capacity but want to keep the fleet at its current size, clear **Terminate instances**.
 - c. Choose **Submit**.

To modify a Spot Fleet request using the AWS CLI

Use the `modify-spot-fleet-request` command to update the target capacity of the specified Spot Fleet request.

```
aws ec2 modify-spot-fleet-request \
--spot-fleet-request-id sfr-73fdb2ce-aa30-494c-8788-1cee4EXAMPLE \
--target-capacity 20
```

You can modify the previous command as follows to decrease the target capacity of the specified Spot Fleet without terminating any Spot Instances as a result.

```
aws ec2 modify-spot-fleet-request \
--spot-fleet-request-id sfr-73fdb2ce-aa30-494c-8788-1cee4EXAMPLE \
--target-capacity 10 \
--excess-capacity-termination-policy NoTermination
```

Canceling a Spot Fleet request

When you are finished using your Spot Fleet, you can cancel the Spot Fleet request. This cancels all Spot requests associated with the Spot Fleet, so that no new Spot Instances are launched for your Spot Fleet. You must specify whether the Spot Fleet should terminate its Spot Instances. If you terminate the instances, the Spot Fleet request enters the `cancelled_terminating` state. Otherwise, the Spot Fleet request enters the `cancelled_running` state and the instances continue to run until they are interrupted or you terminate them manually.

To cancel a Spot Fleet request (console)

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot/home/fleet>.

2. Select your Spot Fleet request.
3. Choose **Actions, Cancel spot request**.
4. In **Cancel spot request**, verify that you want to cancel the Spot Fleet. To keep the fleet at its current size, clear **Terminate instances**. When you are ready, choose **Confirm**.

To cancel a Spot Fleet request using the AWS CLI

Use the [cancel-spot-fleet-requests](#) command to cancel the specified Spot Fleet request and terminate the instances.

```
aws ec2 cancel-spot-fleet-requests \
--spot-fleet-request-ids sfr-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE \
--terminate-instances
```

The following is example output:

```
{  
    "SuccessfulFleetRequests": [  
        {  
            "SpotFleetRequestId": "sfr-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE",  
            "CurrentSpotFleetRequestState": "cancelled_terminating",  
            "PreviousSpotFleetRequestState": "active"  
        }  
    ],  
    "UnsuccessfulFleetRequests": []  
}
```

You can modify the previous command as follows to cancel the specified Spot Fleet request without terminating the instances.

```
aws ec2 cancel-spot-fleet-requests \
--spot-fleet-request-ids sfr-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE \
--no-terminate-instances
```

The following is example output:

```
{  
    "SuccessfulFleetRequests": [  
        {  
            "SpotFleetRequestId": "sfr-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE",  
            "CurrentSpotFleetRequestState": "cancelled_running",  
            "PreviousSpotFleetRequestState": "active"  
        }  
    ],  
    "UnsuccessfulFleetRequests": []  
}
```

Spot Fleet example configurations

The following examples show launch configurations that you can use with the [request-spot-fleet](#) command to create a Spot Fleet request. For more information, see [Creating a Spot Fleet request \(p. 391\)](#).

Note

For Spot Fleet, you can't specify an network interface ID in a launch specification. Make sure you omit the NetworkInterfaceID parameter in your launch specification.

1. Launch Spot Instances using the lowest-priced Availability Zone or subnet in the region (p. 404)
2. Launch Spot Instances using the lowest-priced Availability Zone or subnet in a specified list (p. 404)
3. Launch Spot Instances using the lowest-priced instance type in a specified list (p. 406)
4. Override the price for the request (p. 407)
5. Launch a Spot Fleet using the diversified allocation strategy (p. 408)
6. Launch a Spot Fleet using instance weighting (p. 410)
7. Launch a Spot Fleet with On-Demand capacity (p. 411)

Example 1: Launch Spot Instances using the lowest-priced Availability Zone or subnet in the Region

The following example specifies a single launch specification without an Availability Zone or subnet. The Spot Fleet launches the instances in the lowest-priced Availability Zone that has a default subnet. The price you pay does not exceed the On-Demand price.

```
{  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "KeyName": "my-key-pair",  
            "SecurityGroups": [  
                {  
                    "GroupId": "sg-1a2b3c4d"  
                }  
            ],  
            "InstanceType": "m3.medium",  
            "IamInstanceProfile": {  
                "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"  
            }  
        }  
    ]  
}
```

Example 2: Launch Spot Instances using the lowest-priced Availability Zone or subnet in a specified list

The following examples specify two launch specifications with different Availability Zones or subnets, but the same instance type and AMI.

Availability Zones

The Spot Fleet launches the instances in the default subnet of the lowest-priced Availability Zone that you specified.

```
{  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "KeyName": "my-key-pair",  
            "SecurityGroups": [  
                {  
                    "GroupId": "sg-1a2b3c4d"  
                }  
            ]  
        }  
    ]  
}
```

```
        ],
        "InstanceType": "m3.medium",
        "Placement": {
            "AvailabilityZone": "us-west-2a, us-west-2b"
        },
        "IamInstanceProfile": {
            "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
        }
    }
}
```

Subnets

You can specify default subnets or nondefault subnets, and the nondefault subnets can be from a default VPC or a nondefault VPC. The Spot service launches the instances in whichever subnet is in the lowest-priced Availability Zone.

You can't specify different subnets from the same Availability Zone in a Spot Fleet request.

```
{
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "KeyName": "my-key-pair",
            "SecurityGroups": [
                {
                    "GroupId": "sg-1a2b3c4d"
                }
            ],
            "InstanceType": "m3.medium",
            "SubnetId": "subnet-a61dafcf, subnet-65ea5f08",
            "IamInstanceProfile": {
                "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
            }
        }
    ]
}
```

If the instances are launched in a default VPC, they receive a public IPv4 address by default. If the instances are launched in a nondefault VPC, they do not receive a public IPv4 address by default. Use a network interface in the launch specification to assign a public IPv4 address to instances launched in a nondefault VPC. When you specify a network interface, you must include the subnet ID and security group ID using the network interface.

```
...
{
    "ImageId": "ami-1a2b3c4d",
    "KeyName": "my-key-pair",
    "InstanceType": "m3.medium",
    "NetworkInterfaces": [
        {
            "DeviceIndex": 0,
            "SubnetId": "subnet-1a2b3c4d",
            "Groups": [ "sg-1a2b3c4d" ],
            "AssociatePublicIpAddress": true
        }
    ],
    "IamInstanceProfile": {
        "Arn": "arn:aws:iam::880185128111:instance-profile/my-iam-role"
    }
}
```

```
        }
    ...
}
```

Example 3: Launch Spot Instances using the lowest-priced instance type in a specified list

The following examples specify two launch configurations with different instance types, but the same AMI and Availability Zone or subnet. The Spot Fleet launches the instances using the specified instance type with the lowest price.

Availability Zone

```
{
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-1a2b3c4d"
                }
            ],
            "InstanceType": "cc2.8xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-1a2b3c4d"
                }
            ],
            "InstanceType": "r3.8xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        }
    ]
}
```

Subnet

```
{
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [
                {
                    "GroupId": "sg-1a2b3c4d"
                }
            ],
            "InstanceType": "cc2.8xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "SecurityGroups": [

```

```
{  
    "GroupId": "sg-1a2b3c4d"  
}  
]  
,  
"InstanceType": "r3.8xlarge",  
"SubnetId": "subnet-1a2b3c4d"  
}  
]  
}
```

Example 4. Override the price for the request

We recommend that you use the default maximum price, which is the On-Demand price. If you prefer, you can specify a maximum price for the fleet request and maximum prices for individual launch specifications.

The following examples specify a maximum price for the fleet request and maximum prices for two of the three launch specifications. The maximum price for the fleet request is used for any launch specification that does not specify a maximum price. The Spot Fleet launches the instances using the instance type with the lowest price.

Availability Zone

```
{  
    "SpotPrice": "1.00",  
    "TargetCapacity": 30,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c3.2xlarge",  
            "Placement": {  
                "AvailabilityZone": "us-west-2b"  
            },  
            "SpotPrice": "0.10"  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c3.4xlarge",  
            "Placement": {  
                "AvailabilityZone": "us-west-2b"  
            },  
            "SpotPrice": "0.20"  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c3.8xlarge",  
            "Placement": {  
                "AvailabilityZone": "us-west-2b"  
            }  
        }  
    ]  
}
```

Subnet

```
{  
    "SpotPrice": "1.00",  
    "TargetCapacity": 30,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {
```

```
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "c3.2xlarge",
        "SubnetId": "subnet-1a2b3c4d",
        "SpotPrice": "0.10"
    },
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "c3.4xlarge",
        "SubnetId": "subnet-1a2b3c4d",
        "SpotPrice": "0.20"
    },
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "c3.8xlarge",
        "SubnetId": "subnet-1a2b3c4d"
    }
]
```

Example 5: Launch a Spot Fleet using the diversified allocation strategy

The following example uses the diversified allocation strategy. The launch specifications have different instance types but the same AMI and Availability Zone or subnet. The Spot Fleet distributes the 30 instances across the three launch specifications, such that there are 10 instances of each type. For more information, see [Allocation strategy for Spot Instances \(p. 359\)](#).

Availability Zone

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        }
    ]
}
```

Subnet

```
{  
    "SpotPrice": "0.70",
```

```
"TargetCapacity": 30,  
"AllocationStrategy": "diversified",  
"IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
"LaunchSpecifications": [  
    {  
        "ImageId": "ami-1a2b3c4d",  
        "InstanceType": "c4.2xlarge",  
        "SubnetId": "subnet-1a2b3c4d"  
    },  
    {  
        "ImageId": "ami-1a2b3c4d",  
        "InstanceType": "m3.2xlarge",  
        "SubnetId": "subnet-1a2b3c4d"  
    },  
    {  
        "ImageId": "ami-1a2b3c4d",  
        "InstanceType": "r3.2xlarge",  
        "SubnetId": "subnet-1a2b3c4d"  
    }  
]
```

A best practice to increase the chance that a spot request can be fulfilled by EC2 capacity in the event of an outage in one of the Availability Zones is to diversify across zones. For this scenario, include each Availability Zone available to you in the launch specification. And, instead of using the same subnet each time, use three unique subnets (each mapping to a different zone).

Availability Zone

```
{  
    "SpotPrice": "0.70",  
    "TargetCapacity": 30,  
    "AllocationStrategy": "diversified",  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c4.2xlarge",  
            "Placement": {  
                "AvailabilityZone": "us-west-2a"  
            }  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "m3.2xlarge",  
            "Placement": {  
                "AvailabilityZone": "us-west-2b"  
            }  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.2xlarge",  
            "Placement": {  
                "AvailabilityZone": "us-west-2c"  
            }  
        }  
    ]  
}
```

Subnet

```
{  
    "SpotPrice": "0.70",
```

```
"TargetCapacity": 30,
"AllocationStrategy": "diversified",
"IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
"LaunchSpecifications": [
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "c4.2xlarge",
        "SubnetId": "subnet-1a2b3c4d"
    },
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "m3.2xlarge",
        "SubnetId": "subnet-2a2b3c4d"
    },
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "r3.2xlarge",
        "SubnetId": "subnet-3a2b3c4d"
    }
]
```

Example 6: Launch a Spot Fleet using instance weighting

The following examples use instance weighting, which means that the price is per unit hour instead of per instance hour. Each launch configuration lists a different instance type and a different weight. The Spot Fleet selects the instance type with the lowest price per unit hour. The Spot Fleet calculates the number of Spot Instances to launch by dividing the target capacity by the instance weight. If the result isn't an integer, the Spot Fleet rounds it up to the next integer, so that the size of your fleet is not below its target capacity.

If the `r3.2xlarge` request is successful, Spot provisions 4 of these instances. Divide 20 by 6 for a total of 3.33 instances, then round up to 4 instances.

If the `c3.xlarge` request is successful, Spot provisions 7 of these instances. Divide 20 by 3 for a total of 6.66 instances, then round up to 7 instances.

For more information, see [Spot Fleet instance weighting \(p. 361\)](#).

Availability Zone

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            },
            "WeightedCapacity": 6
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            },
            "WeightedCapacity": 3
        }
    ]
}
```

```
}
```

Subnet

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "SubnetId": "subnet-1a2b3c4d",
            "WeightedCapacity": 6
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.xlarge",
            "SubnetId": "subnet-1a2b3c4d",
            "WeightedCapacity": 3
        }
    ]
}
```

Example 7: Launch a Spot Fleet with On-Demand capacity

To ensure that you always have instance capacity, you can include a request for On-Demand capacity in your Spot Fleet request. If there is capacity, the On-Demand request is always fulfilled. The balance of the target capacity is fulfilled as Spot if there is capacity and availability.

The following example specifies the desired target capacity as 10, of which 5 must be On-Demand capacity. Spot capacity is not specified; it is implied in the balance of the target capacity minus the On-Demand capacity. Amazon EC2 launches 5 capacity units as On-Demand, and 5 capacity units (10-5=5) as Spot if there is available Amazon EC2 capacity and availability.

For more information, see [On-Demand in Spot Fleet \(p. 358\)](#).

```
{
    "IamFleetRole": "arn:aws:iam::781603563322:role/aws-ec2-spot-fleet-tagging-role",
    "AllocationStrategy": "lowestPrice",
    "TargetCapacity": 10,
    "SpotPrice": null,
    "ValidFrom": "2018-04-04T15:58:13Z",
    "ValidUntil": "2019-04-04T15:58:13Z",
    "TerminateInstancesWithExpiration": true,
    "LaunchSpecifications": [],
    "Type": "maintain",
    "OnDemandTargetCapacity": 5,
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0dbb04d4a6cca5ad1",
                "Version": "2"
            },
            "Overrides": [
                {
                    "InstanceType": "t2.medium",
                    "WeightedCapacity": 1,
                    "SubnetId": "subnet-d0dc51fb"
                }
            ]
        }
    ]
}
```

]
}

CloudWatch metrics for Spot Fleet

Amazon EC2 provides Amazon CloudWatch metrics that you can use to monitor your Spot Fleet.

Important

To ensure accuracy, we recommend that you enable detailed monitoring when using these metrics. For more information, see [Enable or turn off detailed monitoring for your instances \(p. 715\)](#).

For more information about CloudWatch metrics provided by Amazon EC2, see [Monitoring your instances using CloudWatch \(p. 715\)](#).

Spot Fleet metrics

The AWS/EC2Spot namespace includes the following metrics, plus the CloudWatch metrics for the Spot Instances in your fleet. For more information, see [Instance metrics \(p. 718\)](#).

Metric	Description
AvailableInstancePoolsCount	The Spot Instance pools specified in the Spot Fleet request. Units: Count
BidsSubmittedForCapacity	The capacity for which Amazon EC2 has submitted Spot Fleet requests. Units: Count
EligibleInstancePoolCount	The Spot Instance pools specified in the Spot Fleet request where Amazon EC2 can fulfill requests. Amazon EC2 does not fulfill requests in pools where the maximum price you're willing to pay for Spot Instances is less than the Spot price or the Spot price is greater than the price for On-Demand Instances. Units: Count
FulfilledCapacity	The capacity that Amazon EC2 has fulfilled. Units: Count
MaxPercentCapacityAllocation	The maximum value of PercentCapacityAllocation across all Spot Fleet pools specified in the Spot Fleet request. Units: Percent
PendingCapacity	The difference between TargetCapacity and FulfilledCapacity. Units: Count
PercentCapacityAllocation	The capacity allocated for the Spot Instance pool for the specified dimensions. To get the maximum value recorded across all Spot Instance pools, use MaxPercentCapacityAllocation. Units: Percent

Metric	Description
TargetCapacity	The target capacity of the Spot Fleet request. Units: Count
TerminatingCapacity	The capacity that is being terminated because the provisioned capacity is greater than the target capacity. Units: Count

If the unit of measure for a metric is Count, the most useful statistic is Average.

Spot Fleet dimensions

To filter the data for your Spot Fleet, use the following dimensions.

Dimensions	Description
AvailabilityZone	Filter the data by Availability Zone.
FleetRequestId	Filter the data by Spot Fleet request.
InstanceType	Filter the data by instance type.

View the CloudWatch metrics for your Spot Fleet

You can view the CloudWatch metrics for your Spot Fleet using the Amazon CloudWatch console. These metrics are displayed as monitoring graphs. These graphs show data points if the Spot Fleet is active.

Metrics are grouped first by namespace, and then by the various combinations of dimensions within each namespace. For example, you can view all Spot Fleet metrics or Spot Fleet metrics groups by Spot Fleet request ID, instance type, or Availability Zone.

To view Spot Fleet metrics

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2 Spot** namespace.

Note

If the **EC2 Spot** namespace is not displayed, there are two reasons for this. Either you've not yet used Spot Fleet—only the AWS services that you're using send metrics to Amazon CloudWatch. Or, if you've not used Spot Fleet for the past two weeks, the namespace does not appear.

4. (Optional) To filter the metrics by dimension, select one of the following:
 - **Fleet Request Metrics** – Group by Spot Fleet request
 - **By Availability Zone** – Group by Spot Fleet request and Availability Zone
 - **By Instance Type** – Group by Spot Fleet request and instance type
 - **By Availability Zone/Instance Type** – Group by Spot Fleet request, Availability Zone, and instance type
5. To view the data for a metric, select the check box next to the metric.

FleetRequestId	Metric Name
sfr-4a707781-8fac-459b-a5ae-4701fcee47d7	AvailableInstancePoolsCount
sfr-4a707781-8fac-459b-a5ae-4701fcee47d7	BidsSubmittedForCapacity
<input checked="" type="checkbox"/> sfr-4a707781-8fac-459b-a5ae-4701fcee47d7	CPUUtilization
sfr-4a707781-8fac-459b-a5ae-4701fcee47d7	DiskReadBytes

Automatic scaling for Spot Fleet

Automatic scaling is the ability to increase or decrease the target capacity of your Spot Fleet automatically based on demand. A Spot Fleet can either launch instances (scale out) or terminate instances (scale in), within the range that you choose, in response to one or more scaling policies.

Spot Fleet supports the following types of automatic scaling:

- [Target tracking scaling \(p. 416\)](#) – Increase or decrease the current capacity of the fleet based on a target value for a specific metric. This is similar to the way that your thermostat maintains the temperature of your home—you select temperature and the thermostat does the rest.
- [Step scaling \(p. 417\)](#) – Increase or decrease the current capacity of the fleet based on a set of scaling adjustments, known as step adjustments, that vary based on the size of the alarm breach.
- [Scheduled scaling \(p. 418\)](#) – Increase or decrease the current capacity of the fleet based on the date and time.

If you are using [instance weighting \(p. 361\)](#), keep in mind that Spot Fleet can exceed the target capacity as needed. Fulfilled capacity can be a floating-point number but target capacity must be an integer, so Spot Fleet rounds up to the next integer. You must take these behaviors into account when you look at the outcome of a scaling policy when an alarm is triggered. For example, suppose that the target capacity is 30, the fulfilled capacity is 30.1, and the scaling policy subtracts 1. When the alarm is triggered, the automatic scaling process subtracts 1 from 30.1 to get 29.1 and then rounds it up to 30, so no scaling action is taken. As another example, suppose that you selected instance weights of 2, 4, and 8, and a target capacity of 10, but no weight 2 instances were available so Spot Fleet provisioned instances of weights 4 and 8 for a fulfilled capacity of 12. If the scaling policy decreases target capacity by 20% and an alarm is triggered, the automatic scaling process subtracts 12×0.2 from 12 to get 9.6 and then rounds it up to 10, so no scaling action is taken.

The scaling policies that you create for Spot Fleet support a cooldown period. This is the number of seconds after a scaling activity completes where previous trigger-related scaling activities can influence future scaling events. For scale-out policies, while the cooldown period is in effect, the capacity that has been added by the previous scale-out event that initiated the cooldown is calculated as part of the desired capacity for the next scale out. The intention is to continuously (but not excessively) scale out. For scale-in policies, the cooldown period is used to block subsequent scale-in requests until it has expired. The intention is to scale in conservatively to protect your application's availability. However, if another alarm triggers a scale-out policy during the cooldown period after a scale-in, automatic scaling scales out your scalable target immediately.

We recommend that you scale based on instance metrics with a 1-minute frequency because that ensures a faster response to utilization changes. Scaling on metrics with a 5-minute frequency can result in slower response time and scaling on stale metric data. To send metric data for your instances to CloudWatch in 1-minute periods, you must specifically enable detailed monitoring. For more information, see [Enable or turn off detailed monitoring for your instances \(p. 715\)](#) and [Create a Spot Fleet request using defined parameters \(console\) \(p. 392\)](#).

For more information about configuring scaling for Spot Fleet, see the following resources:

- [application-autoscaling](#) section of the *AWS CLI Command Reference*
- [Application Auto Scaling API Reference](#)
- [Application Auto Scaling User Guide](#)

IAM permissions required for Spot Fleet automatic scaling

Automatic scaling for Spot Fleet is made possible by a combination of the Amazon EC2, Amazon CloudWatch, and Application Auto Scaling APIs. Spot Fleet requests are created with Amazon EC2, alarms are created with CloudWatch, and scaling policies are created with Application Auto Scaling.

In addition to the [IAM permissions for Spot Fleet \(p. 387\)](#) and Amazon EC2, the IAM user that accesses fleet scaling settings must have the appropriate permissions for the services that support dynamic scaling. IAM users must have permissions to use the actions shown in the following example policy.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "application-autoscaling:*",  
                "ec2:DescribeSpotFleetRequests",  
                "ec2:ModifySpotFleetRequest",  
                "cloudwatch:DeleteAlarms",  
                "cloudwatch:DescribeAlarmHistory",  
                "cloudwatch:DescribeAlarms",  
                "cloudwatch:DescribeAlarmsForMetric",  
                "cloudwatch:GetMetricStatistics",  
                "cloudwatch>ListMetrics",  
                "cloudwatch:PutMetricAlarm",  
                "cloudwatch:DisableAlarmActions",  
                "cloudwatch:EnableAlarmActions",  
                "iam>CreateServiceLinkedRole",  
                "sns>CreateTopic",  
                "sns:Subscribe",  
                "sns:Get*",  
                "sns>List*"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

You can also create your own IAM policies that allow more fine-grained permissions for calls to the Application Auto Scaling API. For more information, see [Authentication and Access Control](#) in the *Application Auto Scaling User Guide*.

The Application Auto Scaling service also needs permission to describe your Spot Fleet and CloudWatch alarms, and permissions to modify your Spot Fleet target capacity on your behalf. If you enable automatic scaling for your Spot Fleet, it creates a service-linked role named `AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest`. This service-linked role

grants Application Auto Scaling permission to describe the alarms for your policies, to monitor the current capacity of the fleet, and to modify the capacity of the fleet. The original managed Spot Fleet role for Application Auto Scaling was `aws-ec2-spot-fleet-autoscale-role`, but it is no longer required. The service-linked role is the default role for Application Auto Scaling. For more information, see [Service-Linked Roles](#) in the *Application Auto Scaling User Guide*.

Scale Spot Fleet using a target tracking policy

With target tracking scaling policies, you select a metric and set a target value. Spot Fleet creates and manages the CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value. The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value. In addition to keeping the metric close to the target value, a target tracking scaling policy also adjusts to the fluctuations in the metric due to a fluctuating load pattern and minimizes rapid fluctuations in the capacity of the fleet.

You can create multiple target tracking scaling policies for a Spot Fleet, provided that each of them uses a different metric. The fleet scales based on the policy that provides the largest fleet capacity. This enables you to cover multiple scenarios and ensure that there is always enough capacity to process your application workloads.

To ensure application availability, the fleet scales out proportionally to the metric as fast as it can, but scales in more gradually.

When a Spot Fleet terminates an instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

Do not edit or delete the CloudWatch alarms that Spot Fleet manages for a target tracking scaling policy. Spot Fleet deletes the alarms automatically when you delete the target tracking scaling policy.

Limitation

- The Spot Fleet request must have a request type of `maintain`. Automatic scaling is not supported for one-time requests or Spot blocks.

To configure a target tracking policy (console)

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, choose **Spot Requests**.
- Select your Spot Fleet request and choose **Auto Scaling**.
- If automatic scaling is not configured, choose **Configure**.
- Use **Scale capacity between** to set the minimum and maximum capacity for your fleet. Automatic scaling does not scale your fleet below the minimum capacity or above the maximum capacity.
- For **Policy name**, type a name for the policy.
- Choose a **Target metric**.
- Enter a **Target value** for the metric.
- (Optional) Set **Cooldown period** to modify the default cooldown period.
- (Optional) Select **Disable scale-in** to omit creating a scale-in policy based on the current configuration. You can create a scale-in policy using a different configuration.
- Choose **Save**.

To configure a target tracking policy using the AWS CLI

- Register the Spot Fleet request as a scalable target using the `register-scalable-target` command.
- Create a scaling policy using the `put-scaling-policy` command.

Scale Spot Fleet using step scaling policies

With step scaling policies, you specify CloudWatch alarms to trigger the scaling process. For example, if you want to scale out when CPU utilization reaches a certain level, create an alarm using the `CPUUtilization` metric provided by Amazon EC2.

When you create a step scaling policy, you must specify one of the following scaling adjustment types:

- **Add** – Increase the target capacity of the fleet by a specified number of capacity units or a specified percentage of the current capacity.
- **Remove** – Decrease the target capacity of the fleet by a specified number of capacity units or a specified percentage of the current capacity.
- **Set to** – Set the target capacity of the fleet to the specified number of capacity units.

When an alarm is triggered, the automatic scaling process calculates the new target capacity using the fulfilled capacity and the scaling policy, and then updates the target capacity accordingly. For example, suppose that the target capacity and fulfilled capacity are 10 and the scaling policy adds 1. When the alarm is triggered, the automatic scaling process adds 1 to 10 to get 11, so Spot Fleet launches 1 instance.

When a Spot Fleet terminates an instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

Limitation

- The Spot Fleet request must have a request type of `maintain`. Automatic scaling is not supported for one-time requests or Spot blocks.

Prerequisites

- Consider which CloudWatch metrics are important to your application. You can create CloudWatch alarms based on metrics provided by AWS or your own custom metrics.
- For the AWS metrics that you will use in your scaling policies, enable CloudWatch metrics collection if the service that provides the metrics does not enable it by default.

To create a CloudWatch alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Alarms**.
3. Choose **Create alarm**.
4. On the **Specify metric and conditions** page, choose **Select metric**.
5. Choose **EC2 Spot, Fleet Request Metrics**, select a metric (for example, `CPUUtilization`), and then choose **Select metric**.

The **Specify metric and conditions** page appears, showing a graph and other information about the metric you selected.

6. For **Period**, choose the evaluation period for the alarm, for example, 1 minute. When evaluating the alarm, each period is aggregated into one data point.

Note

A shorter period creates a more sensitive alarm.

7. For **Conditions**, define the alarm by defining the threshold condition. For example, you can define a threshold to trigger the alarm whenever the value of the metric is greater than or equal to 80 percent.

8. Under **Additional configuration**, for **Datapoints to alarm**, specify how many datapoints (evaluation periods) must be in the ALARM state to trigger the alarm, for example, 1 evaluation period or 2 out of 3 evaluation periods. This creates an alarm that goes to ALARM state if that many consecutive periods are breaching. For more information, see [Evaluating an Alarm](#) in the *Amazon CloudWatch User Guide*.
9. For **Missing data treatment**, choose one of the options (or leave the default of **Treat missing data as missing**). For more information, see [Configuring How CloudWatch Alarms Treat Missing Data](#) in the *Amazon CloudWatch User Guide*.
10. Choose **Next**.
11. (Optional) To receive notification of a scaling event, for **Notification**, you can choose or create the Amazon SNS topic you want to use to receive notifications. Otherwise, you can delete the notification now and add one later as needed.
12. Choose **Next**.
13. Under **Add a description**, enter a name and description for the alarm and choose **Next**.
14. Choose **Create alarm**.

To configure a step scaling policy for your Spot Fleet (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request and choose **Auto Scaling**.
4. If automatic scaling is not configured, choose **Configure**.
5. Use **Scale capacity between** to set the minimum and maximum capacity for your fleet. Automatic scaling does not scale your fleet below the minimum capacity or above the maximum capacity.
6. Initially, **Scaling policies** contains policies named ScaleUp and ScaleDown. You can complete these policies, or choose **Remove policy** to delete them. You can also choose **Add policy**.
7. To define a policy, do the following:
 - a. For **Policy name**, type a name for the policy.
 - b. For **Policy trigger**, select an existing alarm or choose **Create new alarm** to open the Amazon CloudWatch console and create an alarm.
 - c. For **Modify capacity**, select a scaling adjustment type, select a number, and select a unit.
 - d. (Optional) To perform step scaling, choose **Define steps**. By default, an add policy has a lower bound of -infinity and an upper bound of the alarm threshold. By default, a remove policy has a lower bound of the alarm threshold and an upper bound of +infinity. To add another step, choose **Add step**.
 - e. (Optional) To modify the default value for the cooldown period, select a number from **Cooldown period**.
8. Choose **Save**.

To configure step scaling policies for your Spot Fleet using the AWS CLI

1. Register the Spot Fleet request as a scalable target using the [register-scalable-target](#) command.
2. Create a scaling policy using the [put-scaling-policy](#) command.
3. Create an alarm that triggers the scaling policy using the [put-metric-alarm](#) command.

Scale Spot Fleet using scheduled scaling

Scaling based on a schedule enables you to scale your application in response to predictable changes in demand. To use scheduled scaling, you create *scheduled actions*, which tell Spot Fleet to perform

scaling activities at specific times. When you create a scheduled action, you specify the Spot Fleet, when the scaling activity should occur, minimum capacity, and maximum capacity. You can create scheduled actions that scale one time only or that scale on a recurring schedule.

Limits

- The Spot Fleet request must have a request type of `maintain`. Automatic scaling is not supported for one-time requests or Spot blocks.

To create a one-time scheduled action

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, choose **Spot Requests**.
- Select your Spot Fleet request and choose **Scheduled Scaling**.
- Choose **Create Scheduled Action**.
- For **Name**, specify a name for the scheduled action.
- Enter a value for **Minimum capacity**, **Maximum capacity**, or both.
- For **Recurrence**, choose **Once**.
- (Optional) Choose a date and time for **Start time**, **End time**, or both.
- Choose **Submit**.

To scale on a recurring schedule

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, choose **Spot Requests**.
- Select your Spot Fleet request and choose **Scheduled Scaling**.
- For **Recurrence**, choose one of the predefined schedules (for example, **Every day**), or choose **Custom** and type a cron expression. For more information about the cron expressions supported by scheduled scaling, see [Cron Expressions](#) in the *Amazon CloudWatch Events User Guide*.
- (Optional) Choose a date and time for **Start time**, **End time**, or both.
- Choose **Submit**.

To edit a scheduled action

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, choose **Spot Requests**.
- Select your Spot Fleet request and choose **Scheduled Scaling**.
- Select the scheduled action and choose **Actions**, **Edit**.
- Make the needed changes and choose **Submit**.

To delete a scheduled action

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, choose **Spot Requests**.
- Select your Spot Fleet request and choose **Scheduled Scaling**.
- Select the scheduled action and choose **Actions**, **Delete**.
- When prompted for confirmation, choose **Delete**.

To manage scheduled scaling using the AWS CLI

Use the following commands:

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Spot request status

To help you track your Spot Instance requests and plan your use of Spot Instances, use the request status provided by Amazon EC2. For example, the request status can provide the reason why your Spot request isn't fulfilled yet, or list the constraints that are preventing the fulfillment of your Spot request.

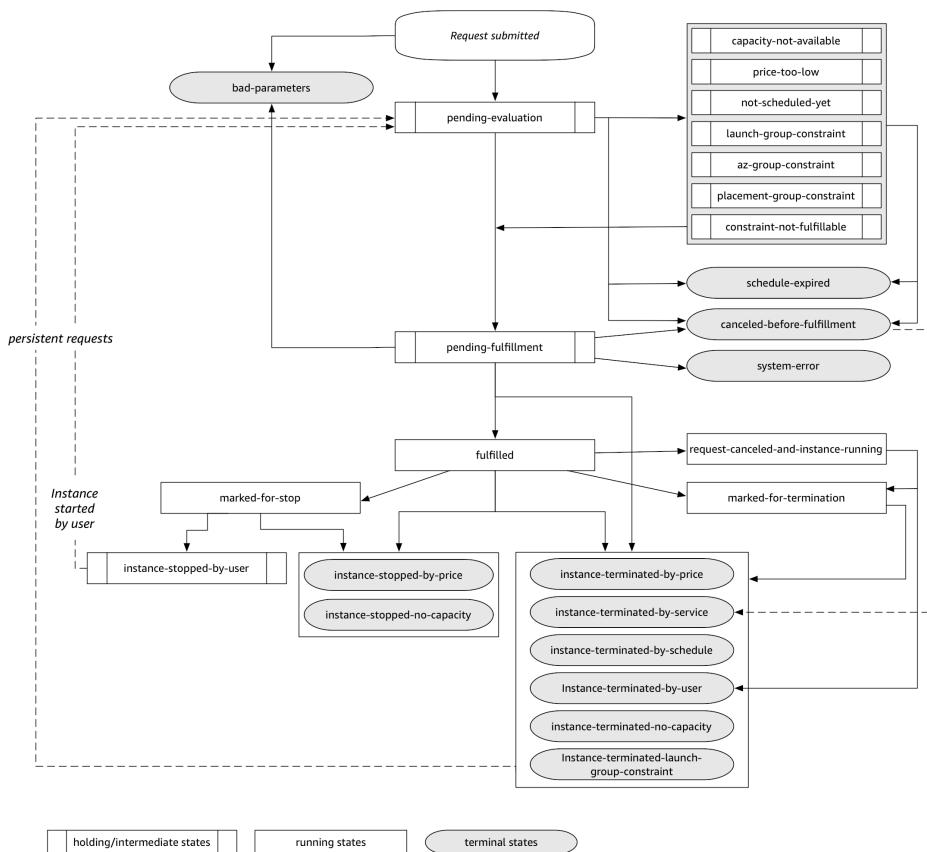
At each step of the process—also called the Spot request *lifecycle*—specific events determine successive request states.

Contents

- [Lifecycle of a Spot request \(p. 420\)](#)
- [Getting request status information \(p. 424\)](#)
- [Spot request status codes \(p. 424\)](#)

Lifecycle of a Spot request

The following diagram shows you the paths that your Spot request can follow throughout its lifecycle, from submission to termination. Each step is depicted as a node, and the status code for each node describes the status of the Spot request and Spot Instance.



Pending evaluation

As soon as you create a Spot Instance request, it goes into the **pending-evaluation** state unless one or more request parameters are not valid (**bad-parameters**).

Status code	Request state	Instance state
pending-evaluation	open	n/a
bad-parameters	closed	n/a

Holding

If one or more request constraints are valid but can't be met yet, or if there is not enough capacity, the request goes into a holding state waiting for the constraints to be met. The request options affect the likelihood of the request being fulfilled. For example, if you specify a maximum price below the current Spot price, your request stays in a holding state until the Spot price goes below your maximum price. If you specify an Availability Zone group, the request stays in a holding state until the Availability Zone constraint is met.

In the event of an outage of one of the Availability Zones, there is a chance that the spare EC2 capacity available for Spot Instance requests in other Availability Zones can be affected.

Status code	Request state	Instance state
capacity-not-available	open	n/a

Status code	Request state	Instance state
price-too-low	open	n/a
not-scheduled-yet	open	n/a
launch-group-constraint	open	n/a
az-group-constraint	open	n/a
placement-group-constraint	open	n/a
constraint-not-fulfillable	open	n/a

Pending evaluation/fulfillment-terminal

Your Spot Instance request can go to a terminal state if you create a request that is valid only during a specific time period and this time period expires before your request reaches the pending fulfillment phase. It might also happen if you cancel the request, or if a system error occurs.

Status code	Request state	Instance state
schedule-expired	cancelled	n/a
canceled-before-fulfillment*	cancelled	n/a
bad-parameters	failed	n/a
system-error	closed	n/a

* If you cancel the request.

Pending fulfillment

When the constraints you specified (if any) are met and your maximum price is equal to or higher than the current Spot price, your Spot request goes into the pending-fulfillment state.

At this point, Amazon EC2 is getting ready to provision the instances that you requested. If the process stops at this point, it is likely to be because it was canceled by the user before a Spot Instance was launched. It may also be because an unexpected system error occurred.

Status code	Request state	Instance state
pending-fulfillment	open	n/a

Fulfilled

When all the specifications for your Spot Instances are met, your Spot request is fulfilled. Amazon EC2 launches the Spot Instances, which can take a few minutes. If a Spot Instance is hibernated or stopped when interrupted, it remains in this state until the request can be fulfilled again or the request is canceled.

Status code	Request state	Instance state
fulfilled	active	pending → running
fulfilled	active	stopped → running

If you stop a Spot Instance, your Spot request goes into the `marked-for-stop` or `instance-stopped-by-user` state until the Spot Instance can be started again or the request is cancelled.

Status code	Request state	Instance state
<code>marked-for-stop</code>	active	stopping
<code>instance-stopped-by-user</code> *	disabled or cancelled**	stopped

* A Spot Instance goes into the `instance-stopped-by-user` state if you stop the instance or run the shutdown command from the instance. After you've stopped the instance, you can start it again. On restart, the Spot Instance request returns to the pending-evaluation state and then Amazon EC2 launches a new Spot Instance when the constraints are met.

** The Spot request state is `disabled` if you stop the Spot Instance but do not cancel the request. The request state is `cancelled` if your spot instance is stopped and the request expires.

Fulfilled-terminal

Your Spot Instances continue to run as long as your maximum price is at or above the Spot price, there is available capacity for your instance type, and you don't terminate the instance. If a change in the Spot price or available capacity requires Amazon EC2 to terminate your Spot Instances, the Spot request goes into a terminal state. A request also goes into the terminal state if you cancel the Spot request or terminate the Spot Instances.

Status code	Request state	Instance state
<code>request-canceled-and-instance-running</code>	<code>cancelled</code>	<code>running</code>
<code>marked-for-stop</code>	<code>active</code>	<code>running</code>
<code>marked-for-termination</code>	<code>active</code>	<code>running</code>
<code>instance-stopped-by-price</code>	<code>disabled</code>	<code>stopped</code>
<code>instance-stopped-by-user</code>	<code>disabled</code>	<code>stopped</code>
<code>instance-stopped-no-capacity</code>	<code>disabled</code>	<code>stopped</code>
<code>instance-terminated-by-price</code>	<code>closed (one-time), open(persistent)</code>	<code>terminated</code>
<code>instance-terminated-by-schedule</code>	<code>closed</code>	<code>terminated</code>
<code>instance-terminated-by-service</code>	<code>cancelled</code>	<code>terminated</code>

Status code	Request state	Instance state
instance-terminated-by-user	closed or cancelled *	terminated
instance-terminated-no-capacity	closed (one-time), open (persistent)	terminated
instance-terminated-launch-group-constraint	closed (one-time), open (persistent)	terminated

* The request state is `closed` if you terminate the instance but do not cancel the request. The request state is `cancelled` if you terminate the instance and cancel the request. Even if you terminate a Spot Instance before you cancel its request, there might be a delay before Amazon EC2 detects that your Spot Instance was terminated. In this case, the request state can either be `closed` or `cancelled`.

Persistent requests

When your Spot Instances are terminated (either by you or Amazon EC2), if the Spot request is a persistent request, it returns to the `pending-evaluation` state and then Amazon EC2 can launch a new Spot Instance when the constraints are met.

Getting request status information

You can get request status information using the AWS Management Console or a command line tool.

To get request status information (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests** and select the Spot request.
3. To check the status, on the **Description** tab, check the **Status** field.

To get request status information using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `describe-spot-instance-requests` (AWS CLI)
- `Get-EC2SpotInstanceRequest` (AWS Tools for Windows PowerShell)

Spot request status codes

Spot request status information is composed of a status code, the update time, and a status message. Together, these help you determine the disposition of your Spot request.

The following are the Spot request status codes:

`az-group-constraint`

Amazon EC2 cannot launch all the instances you requested in the same Availability Zone.

`bad-parameters`

One or more parameters for your Spot request are not valid (for example, the AMI you specified does not exist). The status message indicates which parameter is not valid.

canceled-before-fulfillment

The user canceled the Spot request before it was fulfilled.

capacity-not-available

There is not enough capacity available for the instances that you requested.

constraint-not-fulfillable

The Spot request can't be fulfilled because one or more constraints are not valid (for example, the Availability Zone does not exist). The status message indicates which constraint is not valid.

fulfilled

The Spot request is active, and Amazon EC2 is launching your Spot Instances.

instance-stopped-by-price

Your instance was stopped because the Spot price exceeded your maximum price.

instance-stopped-by-user

Your instance was stopped because a user stopped the instance or ran the shutdown command from the instance.

instance-stopped-no-capacity

Your instance was stopped because there was no longer enough Spot capacity available for the instance.

instance-terminated-by-price

Your instance was terminated because the Spot price exceeded your maximum price. If your request is persistent, the process restarts, so your request is pending evaluation.

instance-terminated-by-schedule

Your Spot Instance was terminated at the end of its scheduled duration.

instance-terminated-by-service

Your instance was terminated from a stopped state.

instance-terminated-by-user or **spot-instance-terminated-by-user**

You terminated a Spot Instance that had been fulfilled, so the request state is closed (unless it's a persistent request) and the instance state is terminated.

instance-terminated-launch-group-constraint

One or more of the instances in your launch group was terminated, so the launch group constraint is no longer fulfilled.

instance-terminated-no-capacity

Your instance was terminated because there is no longer enough Spot capacity available for the instance.

launch-group-constraint

Amazon EC2 cannot launch all the instances that you requested at the same time. All instances in a launch group are started and terminated together.

limit-exceeded

The limit on the number of EBS volumes or total volume storage was exceeded. For more information about these limits and how to request an increase, see [Amazon EBS Limits](#) in the [Amazon Web Services General Reference](#).

`marked-for-stop`

The Spot Instance is marked for stopping.

`marked-for-termination`

The Spot Instance is marked for termination.

`not-scheduled-yet`

The Spot request is not evaluated until the scheduled date.

`pending-evaluation`

After you make a Spot Instance request, it goes into the pending-evaluation state while the system evaluates the parameters of your request.

`pending-fulfillment`

Amazon EC2 is trying to provision your Spot Instances.

`placement-group-constraint`

The Spot request can't be fulfilled yet because a Spot Instance can't be added to the placement group at this time.

`price-too-low`

The request can't be fulfilled yet because your maximum price is below the Spot price. In this case, no instance is launched and your request remains open.

`request-canceled-and-instance-running`

You canceled the Spot request while the Spot Instances are still running. The request is cancelled, but the instances remain running.

`schedule-expired`

The Spot request expired because it was not fulfilled before the specified date.

`system-error`

There was an unexpected system error. If this is a recurring issue, please contact AWS Support for assistance.

Spot Instance interruptions

You can launch Spot Instances on spare EC2 capacity for steep discounts in exchange for returning them when Amazon EC2 needs the capacity back. When Amazon EC2 reclaims a Spot Instances, we call this event a *Spot Instance interruption*.

Demand for Spot Instances can vary significantly from moment to moment, and the availability of Spot Instances can also vary significantly depending on how many unused EC2 instances are available. It is always possible that your Spot Instance might be interrupted. Therefore, you must ensure that your application is prepared for a Spot Instance interruption.

An On-Demand Instance specified in an EC2 Fleet or Spot Fleet cannot be interrupted.

Contents

- [Reasons for interruption \(p. 427\)](#)
- [Interruption behaviors \(p. 427\)](#)
- [Specifying the interruption behavior \(p. 429\)](#)

- [Preparing for interruptions \(p. 430\)](#)
- [Preparing for instance hibernation \(p. 430\)](#)
- [Spot Instance interruption notices \(p. 431\)](#)
- [Finding interrupted Spot Instances \(p. 433\)](#)
- [Determining whether Amazon EC2 interrupted a Spot Instance \(p. 433\)](#)
- [Billing for interrupted Spot Instances \(p. 433\)](#)

Reasons for interruption

The following are the possible reasons that Amazon EC2 might interrupt your Spot Instances:

- Price – The Spot price is greater than your maximum price.
- Capacity – If there are not enough unused EC2 instances to meet the demand for On-Demand Instances, Amazon EC2 interrupts Spot Instances. The order in which the instances are interrupted is determined by Amazon EC2.
- Constraints – If your request includes a constraint such as a launch group or an Availability Zone group, these Spot Instances are terminated as a group when the constraint can no longer be met.

Interruption behaviors

You can specify that Amazon EC2 should do one of the following when it interrupts a Spot Instance:

- Stop the Spot Instance
- Hibernate the Spot Instance
- Terminate the Spot Instance

The default is to terminate Spot Instances when they are interrupted. To change the interruption behavior, see [Specifying the interruption behavior \(p. 429\)](#).

Stopping interrupted Spot Instances

You can specify the interruption behavior so that Amazon EC2 stops Spot Instances when they are interrupted if the following requirements are met.

Requirements

- For a Spot Instance request, the type must be `persistent`. You cannot specify a launch group in the Spot Instance request.
- For an EC2 Fleet or Spot Fleet request, the type must be `maintain`.
- The root volume must be an EBS volume, not an instance store volume.

After a Spot Instance is stopped by the Spot service, only the Spot service can restart the Spot Instance, and the same launch specification must be used.

For a Spot Instance launched by a `persistent` Spot Instance request, the Spot service restarts the stopped instance when capacity is available in the same Availability Zone and for the same instance type as the stopped instance.

If instances in an EC2 Fleet or Spot Fleet are stopped and the fleet is of type `maintain`, the Spot service launches replacement instances to maintain the target capacity. The Spot service finds the best pools based on the specified allocation strategy (`lowestPrice`, `diversified`, or

`InstancePoolsToUseCount`); it does not prioritize the pool with the earlier stopped instances. Later, if the allocation strategy leads to a pool containing the earlier stopped instances, the Spot service restarts the stopped instances to meet the target capacity.

For example, consider a Spot Fleet with the `lowestPrice` allocation strategy. At initial launch, a `c3.large` pool meets the `lowestPrice` criteria for the launch specification. Later, when the `c3.large` instances are interrupted, the Spot service stops the instances and replenishes capacity from another pool that fits the `lowestPrice` strategy. This time, the pool happens to be a `c4.large` pool and the Spot service launches `c4.large` instances to meet the target capacity. Similarly, Spot Fleet could move to a `c5.large` pool the next time. In each of these transitions, the Spot service does not prioritize pools with earlier stopped instances, but rather prioritizes purely on the specified allocation strategy. The `lowestPrice` strategy can lead back to pools with earlier stopped instances. For example, if instances are interrupted in the `c5.large` pool and the `lowestPrice` strategy leads it back to the `c3.large` or `c4.large` pools, the earlier stopped instances are restarted to fulfill target capacity.

While a Spot Instance is stopped, you can modify some of its instance attributes, but not the instance type. If you detach or delete an EBS volume, it is not attached when the Spot Instance is started. If you detach the root volume and the Spot service attempts to start the Spot Instance, instance start fails and the Spot service terminates the stopped instance.

You can terminate a Spot Instance while it is stopped. If you cancel a Spot request, an EC2 Fleet, or a Spot Fleet, the Spot service terminates any associated Spot Instances that are stopped.

While a Spot Instance is stopped, you are charged only for the EBS volumes, which are preserved. With EC2 Fleet and Spot Fleet, if you have many stopped instances, you can exceed the limit on the number of EBS volumes for your account.

Hibernating interrupted Spot Instances

You can specify the interruption behavior so that Amazon EC2 hibernates Spot Instances when they are interrupted if the following requirements are met.

Requirements

- For a Spot Instance request, the type must be `persistent`. You cannot specify a launch group in the Spot Instance request.
- For an EC2 Fleet or Spot Fleet request, the type must be `maintain`.
- The root volume must be an EBS volume, not an instance store volume, and it must be large enough to store the instance memory (RAM) during hibernation.
- The following instances are supported: C3, C4, C5, M4, M5, R3, and R4, with less than 100 GB of memory.
- The following operating systems are supported: Amazon Linux 2, Amazon Linux AMI, Ubuntu with an AWS-tuned Ubuntu kernel (`linux-aws`) greater than 4.4.0-1041, and Windows Server 2008 R2 and later.
- Install the hibernation agent on a supported operating system, or use one of the following AMIs, which already include the agent:
 - Amazon Linux 2
 - Amazon Linux AMI 2017.09.1 or later
 - Ubuntu Xenial 16.04 20171121 or later
 - Windows Server 2008 R2 AMI 2017.11.19 or later
 - Windows Server 2012 or Windows Server 2012 R2 AMI 2017.11.19 or later
 - Windows Server 2016 AMI 2017.11.19 or later
 - Windows Server 2019
- Start the agent. We recommend that you use user data to start the agent on instance startup. Alternatively, you could start the agent manually.

Recommendation

- We strongly recommend that you use an encrypted Amazon EBS volume as the root volume, because instance memory is stored on the root volume during hibernation. This ensures that the contents of memory (RAM) are encrypted when the data is at rest on the volume and when data is moving between the instance and volume. Use one of the following three options to ensure that the root volume is an encrypted Amazon EBS volume:
 - EBS “single-step” encryption: In a single run-instances API call, you can launch encrypted EBS-backed EC2 instances from an unencrypted AMI. For more information, see [Using encryption with EBS-backed AMIs \(p. 157\)](#).
 - EBS encryption by default: You can enable EBS encryption by default to ensure all new EBS volumes created in your AWS account are encrypted. For more information, see [Encryption by default \(p. 1124\)](#).
 - Encrypted AMI: You can enable EBS encryption by using an encrypted AMI to launch your instance. If your AMI does not have an encrypted root snapshot, you can copy it to a new AMI and request encryption. For more information, see [Encrypt an unencrypted image during copy \(p. 162\)](#) and [Copying an AMI \(p. 167\)](#).

When a Spot Instance is hibernated by the Spot service, the EBS volumes are preserved and instance memory (RAM) is preserved on the root volume. The private IP addresses of the instance are also preserved. Instance storage volumes and public IP addresses, other than Elastic IP addresses, are not preserved. While the instance is hibernating, you are charged only for the EBS volumes. With EC2 Fleet and Spot Fleet, if you have many hibernated instances, you can exceed the limit on the number of EBS volumes for your account.

The agent prompts the operating system to hibernate when the instance receives a signal from the Spot service. If the agent is not installed, the underlying operating system doesn't support hibernation, or there isn't enough volume space to save the instance memory, hibernation fails and the Spot service stops the instance instead.

When the Spot service hibernates a Spot Instance, you receive an interruption notice, but you do not have two minutes before the Spot Instance is interrupted. Hibernation begins immediately. While the instance is in the process of hibernating, instance health checks might fail. When the hibernation process completes, the state of the instance is stopped.

Resuming a hibernated Spot Instance

After a Spot Instance is hibernated by the Spot service, it can only be resumed by the Spot service. The Spot service resumes the instance when capacity becomes available with a Spot price that is less than your specified maximum price.

For more information, see [Preparing for instance hibernation \(p. 430\)](#).

For information about hibernating On-Demand Instances, see [Hibernate your Linux instance \(p. 591\)](#).

Specifying the interruption behavior

If you do not specify an interruption behavior, the default is to terminate Spot Instances when they are interrupted. You can specify the interruption behavior when you create a Spot request. The way in which you specify the interruption behavior is different depending on how you request Spot Instances.

If you request Spot Instances using the [launch instance wizard \(p. 499\)](#), you can specify the interruption behavior as follows: Select the **Persistent request** check box and then, from **Interruption behavior**, choose an interruption behavior.

If you request Spot Instances using the [Spot console \(p. 391\)](#), you can specify the interruption behavior as follows: Select the **Maintain target capacity** check box and then, from **Interruption behavior**, choose an interruption behavior.

If you configure Spot Instances in a [launch template \(p. 506\)](#), you can specify the interruption behavior as follows: In the launch template, expand **Advanced details** and select the **Request Spot Instances** checkbox. Choose **Customize** and then, from **Interruption behavior**, choose an interruption behavior.

If you configure Spot Instances in a launch configuration when using the [request-spot-fleet](#) CLI, you can specify the interruption behavior as follows: For `InstanceInterruptionBehavior`, specify an interruption behavior.

If you configure Spot Instances using the [request-spot-instances](#) CLI, you can specify the interruption behavior as follows: For `--instance-interruption-behavior`, specify an interruption behavior.

Preparing for interruptions

Here are some best practices to follow when you use Spot Instances:

- Use the default maximum price, which is the On-Demand price.
- Ensure that your instance is ready to go as soon as the request is fulfilled by using an Amazon Machine Image (AMI) that contains the required software configuration. You can also use user data to run commands at start-up.
- Store important data regularly in a place that isn't affected when the Spot Instance terminates. For example, you can use Amazon S3, Amazon EBS, or DynamoDB.
- Divide the work into small tasks (using a Grid, Hadoop, or queue-based architecture) or use checkpoints so that you can save your work frequently.
- Use Spot Instance interruption notices to monitor the status of your Spot Instances.
- While we make every effort to provide this warning as soon as possible, it is possible that your Spot Instance is terminated before the warning can be made available. Test your application to ensure that it handles an unexpected instance termination gracefully, even if you are testing for interruption notices. You can do so by running the application using an On-Demand Instance and then terminating the On-Demand Instance yourself.

Preparing for instance hibernation

You must install a hibernation agent on your instance, unless you used an AMI that already includes the agent. You must run the agent on instance startup, whether the agent was included in your AMI or you installed it yourself.

The following procedures help you prepare a Linux instance. For directions to prepare a Windows instance, see [Preparing for instance hibernation](#) in the *Amazon EC2 User Guide for Windows Instances*.

To prepare an Amazon Linux instance

1. Verify that your kernel supports hibernation and update the kernel if necessary.
2. If your AMI doesn't include the agent, install the agent using the following command.

```
sudo yum update; sudo yum install hibagent
```

3. Add the following to the user data:

```
#!/bin/bash
/usr/bin/enable-ec2-spot-hibernation
```

To prepare an Ubuntu instance

1. If your AMI doesn't include the agent, install the agent using the following command. The hibernation agent is only available on Ubuntu 16.04 or later.

```
sudo apt-get install hibagent
```

2. Add the following to the user data.

```
#!/bin/bash
/usr/bin/enable-ec2-spot-hibernation
```

Spot Instance interruption notices

The best way for you to gracefully handle Spot Instance interruptions is to architect your application to be fault-tolerant. To accomplish this, you can take advantage of *Spot Instance interruption notices*. A Spot Instance interruption notice is a warning that is issued two minutes before Amazon EC2 stops or terminates your Spot Instance. If you specify hibernation as the interruption behavior, you receive an interruption notice, but you do not receive a two-minute warning because the hibernation process begins immediately.

We recommend that you check for these interruption notices every 5 seconds.

The interruption notice is made available as a CloudWatch event and as an item in the [instance metadata \(p. 658\)](#) on the Spot Instance.

EC2 Spot Instance interruption notice

When Amazon EC2 is going to interrupt your Spot Instance, it emits an event two minutes prior to the actual interruption (except for hibernation, which gets the interruption notice, but not two minutes in advance, because hibernation begins immediately). This event can be detected by Amazon CloudWatch Events. For more information about CloudWatch events, see the [Amazon CloudWatch Events User Guide](#). For a detailed example that walks you through how to create and use event rules, see [Taking Advantage of Amazon EC2 Spot Instance Interruption Notices](#).

The following is an example of the event for Spot Instance interruption. The possible values for `instance-action` are `hibernate`, `stop`, and `terminate`.

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Spot Instance Interruption Warning",
    "source": "aws.ec2",
    "account": "123456789012",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-1234567890abcdef0"],
    "detail": {
        "instance-id": "i-1234567890abcdef0",
        "instance-action": "action"
    }
}
```

instance-action

If your Spot Instance is marked to be stopped or terminated by the Spot service, the `instance-action` item is present in your [instance metadata \(p. 658\)](#). Otherwise, it is not present. You can retrieve `instance-action` as follows.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
```

```
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/spot/instance-action
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/spot/instance-action
```

The `instance-action` item specifies the action and the approximate time, in UTC, when the action will occur.

The following example indicates the time at which this instance will be stopped.

```
{"action": "stop", "time": "2017-09-18T08:22:00Z"}
```

The following example indicates the time at which this instance will be terminated.

```
{"action": "terminate", "time": "2017-09-18T08:22:00Z"}
```

If Amazon EC2 is not preparing to stop or terminate the instance, or if you terminated the instance yourself, `instance-action` is not present and you receive an HTTP 404 error when you try to retrieve it.

termination-time

This item is maintained for backward compatibility; you should use `instance-action` instead.

If your Spot Instance is marked for termination by the Spot service, the `termination-time` item is present in your instance metadata. Otherwise, it is not present. You can retrieve `termination-time` as follows.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
[ec2-user ~]$ if curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/spot/termination-time | grep -q .*T.*Z; then echo terminated; fi
```

IMDSv1

```
[ec2-user ~]$ if curl -s http://169.254.169.254/latest/meta-data/spot/termination-time | grep -q .*T.*Z; then echo terminated; fi
```

The `termination-time` item specifies the approximate time in UTC when the instance receives the shutdown signal. For example:

```
2015-01-05T18:02:00Z
```

If Amazon EC2 is not preparing to terminate the instance, or if you terminated the Spot Instance yourself, the `termination-time` item is either not present (so you receive an HTTP 404 error) or contains a value that is not a time value.

If Amazon EC2 fails to terminate the instance, the request status is set to `fulfilled`. The `termination-time` value remains in the instance metadata with the original approximate time, which is now in the past.

Finding interrupted Spot Instances

In the console, the **Instances** pane displays all instances, including Spot Instances. You can identify a Spot Instance from the **spot** value in the **Instance lifecycle** column. The **Instance state** column indicates whether the instance is pending, running, stopping, stopped, shutting-down, or terminated. For a hibernated Spot Instance, the instance state is stopped.

To find an interrupted Spot Instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**. In the top right corner, choose the settings icon (), and under **Attribute columns**, select **Instance lifecycle**. For Spot Instances, **Instance lifecycle** is **spot**.

Alternatively, in the navigation pane, choose **Spot Requests**. You can see both Spot Instance requests and Spot Fleet requests. To view the IDs of the instances, select a Spot Instance request or a Spot Fleet request and choose the **Instances** tab. Choose an instance ID to display the instance in the **Instances** pane.

3. For each Spot Instance, you can view its state in the **Instance State** column.

To find interrupted Spot Instances (AWS CLI)

You can list your interrupted Spot Instances using the `describe-instances` command with the `--filters` parameter. To list only the instance IDs in the output, add the `--query` parameter.

```
aws ec2 describe-instances \
  --filters Name=instance-lifecycle,Values=spot Name=instance-state-
  name,Values=terminated,stopped \
  --query Reservations[*].Instances[*].InstanceId
```

Determining whether Amazon EC2 interrupted a Spot Instance

If a Spot Instance is stopped, hibernated, or terminated, you can use CloudTrail to see whether Amazon EC2 interrupted the Spot Instance. In CloudTrail, the event name `BidEvictedEvent` indicates that Amazon EC2 interrupted the Spot Instance. For more information about using CloudTrail, see [Logging Amazon EC2 and Amazon EBS API calls with AWS CloudTrail \(p. 755\)](#).

Billing for interrupted Spot Instances

When a Spot Instance (*not* in a Spot block) is interrupted, you're charged as follows.

Who interrupts the Spot Instance	Operating system	Interrupted in the first hour	Interrupted in any hour after the first hour
If you stop or terminate the Spot Instance	Linux (excluding RHEL and SUSE)	Charged for the seconds used	Charged for the seconds used
	Windows, RHEL, SUSE	Charged for the full hour even if you used a partial hour	Charged for the full hours used, and charged a full hour for the interrupted partial hour
If Amazon EC2 interrupts the Spot Instance	Linux (excluding RHEL and SUSE)	No charge	Charged for the seconds used

Who interrupts the Spot Instance	Operating system	Interrupted in the first hour	Interrupted in any hour after the first hour
	Windows, RHEL, SUSE	No charge	Charged for the full hours used, but no charge for the interrupted partial hour

When a Spot Instance *in a Spot block* is interrupted, you're charged as follows.

Who interrupts the Spot Instance	Operating system	Interrupted in the first hour	Interrupted in any hour after the first hour
If you stop or terminate the Spot Instance	Linux (excluding RHEL and SUSE)	Charged for the seconds used	Charged for the seconds used
	Windows, RHEL, SUSE	Charged for the full hour even if you used a partial hour	Charged for the full hours used, and charged a full hour for the interrupted partial hour
If Amazon EC2 interrupts the Spot Instance	Linux (excluding RHEL and SUSE)	No charge	No charge
	Windows, RHEL, SUSE	No charge	No charge

Spot Instance data feed

To help you understand the charges for your Spot Instances, Amazon EC2 provides a data feed that describes your Spot Instance usage and pricing. This data feed is sent to an Amazon S3 bucket that you specify when you subscribe to the data feed.

Data feed files arrive in your bucket typically once an hour, and each hour of usage is typically covered in a single data file. These files are compressed (gzip) before they are delivered to your bucket. Amazon EC2 can write multiple files for a given hour of usage where files are large (for example, when file contents for the hour exceed 50 MB before compression).

Note

If you don't have a Spot Instance running during a certain hour, you don't receive a data feed file for that hour.

Contents

- [Data feed file name and format \(p. 434\)](#)
- [Amazon S3 bucket requirements \(p. 435\)](#)
- [Subscribing to your Spot Instance data feed \(p. 436\)](#)
- [Deleting your Spot Instance data feed \(p. 436\)](#)

Data feed file name and format

The Spot Instance data feed file name uses the following format (with the date and hour in UTC):

`bucket-name.s3.amazonaws.com/optional-prefix/aws-account-id.YYYY-MM-DD-HH.n.unique-id.gz`

For example, if your bucket name is **my-bucket-name** and your prefix is **my-prefix**, your file names are similar to the following:

`my-bucket-name.s3.amazonaws.com/my-prefix/111122223333.2019-03-17-20.001.pwBdGTJG.gz`

For more information about bucket names, see [Rules for bucket naming in the Amazon Simple Storage Service Developer Guide](#).

The Spot Instance data feed files are tab-delimited. Each line in the data file corresponds to one instance hour and contains the fields listed in the following table.

Field	Description
Timestamp	The timestamp used to determine the price charged for this instance usage.
UsageType	The type of usage and instance type being charged for. For m1.small Spot Instances, this field is set to SpotUsage. For all other instance types, this field is set to SpotUsage:{instance-type}. For example, SpotUsage:c1.medium.
Operation	The product being charged for. For Linux Spot Instances, this field is set to RunInstances. For Windows Spot Instances, this field is set to RunInstances:0002. Spot usage is grouped according to Availability Zone.
InstanceID	The ID of the Spot Instance that generated this instance usage.
MyBidID	The ID for the Spot Instance request that generated this instance usage.
MyMaxPrice	The maximum price specified for this Spot Instance request.
MarketPrice	The Spot price at the time specified in the <code>Timestamp</code> field.
Charge	The price charged for this instance usage.
Version	The version included in the data feed file name for this record.

Amazon S3 bucket requirements

When you subscribe to the data feed, you must specify an Amazon S3 bucket to store the data feed files. Before you choose an Amazon S3 bucket for the data feed, consider the following:

- You must have `FULL_CONTROL` permission to the bucket, which includes permission for the `s3:GetBucketAcl` and `s3:PutBucketAcl` actions.

If you're the bucket owner, you have this permission by default. Otherwise, the bucket owner must grant your AWS account this permission.

- When you subscribe to a data feed, these permissions are used to update the bucket ACL to give the AWS data feed account `FULL_CONTROL` permission. The AWS data feed account writes data feed files to the bucket. If your account doesn't have the required permissions, the data feed files cannot be written to the bucket.

Note

If you update the ACL and remove the permissions for the AWS data feed account, the data feed files cannot be written to the bucket. You must resubscribe to the data feed to receive the data feed files.

- Each data feed file has its own ACL (separate from the ACL for the bucket). The bucket owner has `FULL_CONTROL` permission to the data files. The AWS data feed account has read and write permissions.

- If you delete your data feed subscription, Amazon EC2 doesn't remove the read and write permissions for the AWS data feed account on either the bucket or the data files. You must remove these permissions yourself.

Subscribing to your Spot Instance data feed

To subscribe to your data feed, use the [create-spot-datafeed-subscription](#) command.

```
aws ec2 create-spot-datafeed-subscription \
  --bucket my-bucket-name \
  [--prefix my-prefix]
```

The following is example output:

```
{  
    "SpotDatafeedSubscription": {  
        "OwnerId": "111122223333",  
        "Bucket": "my-bucket-name",  
        "Prefix": "my-prefix",  
        "State": "Active"  
    }  
}
```

Deleting your Spot Instance data feed

To delete your data feed, use the [delete-spot-datafeed-subscription](#) command.

```
aws ec2 delete-spot-datafeed-subscription
```

Spot Instance limits

There is a limit on the number of running and requested Spot Instances per AWS account per Region. Spot Instance limits are managed in terms of the *number of virtual central processing units (vCPUs)* that your running Spot Instances are either using or will use pending the fulfillment of open Spot Instance requests. If you terminate your Spot Instances but do not cancel the Spot Instance requests, the requests count against your Spot Instance vCPU limit until Amazon EC2 detects the Spot Instance terminations and closes the requests.

Topics

- [Spot Instance limits \(p. 436\)](#)
- [Requesting a Spot Instance limit increase \(p. 437\)](#)
- [Monitoring Spot Instance limits and usage \(p. 437\)](#)
- [Spot Fleet limits \(p. 438\)](#)

Spot Instance limits

There are six Spot Instance limits, listed in the following table. Each limit specifies the vCPU limit for one or more instance families. For information about the different instance families, generations, and sizes, see [Amazon EC2 Instance Types](#).

Spot Instance limit name	Default vCPU limit
All Standard (A, C, D, H, I, M, R, T, Z) Spot Instance Requests	1440 vCPUs

Spot Instance limit name	Default vCPU limit
All F Spot Instance Requests	11 vCPUs
All G Spot Instance Requests	11 vCPUs
All Inf Spot Instance Requests	64 vCPUs
All P Spot Instance Requests	16 vCPUs
All X Spot Instance Requests	21 vCPUs

Note

New AWS accounts might start with limits that are lower than the limits described here. These limits can increase over time.

With vCPU limits, you can use your limit in terms of the number of vCPUs that are required to launch any combination of instance types that meet your changing application needs. For example, with an All Standard Spot Instance Requests limit of 256 vCPUs, you could request 32 m5.2xlarge Spot Instances (32 x 8 vCPUs) or 16 c5.4xlarge Spot Instances (16 x 16 vCPUs), or a combination of any Standard Spot Instance types and sizes that total 256 vCPUs.

Requesting a Spot Instance limit increase

Even though Amazon EC2 automatically increases your Spot Instance limits based on your usage, you can request a limit increase if necessary. For example, if you intend to launch more Spot Instances than your current limit allows, you can request a limit increase. You can also request a limit increase if you submit a Spot Instance request and you receive the error `Max spot instance count exceeded`.

To request a Spot Instance limit increase

1. Open the **Create case, Service limit increase** form in the Support Center console at <https://console.aws.amazon.com/support/home#/case/create>.
2. For **Limit type**, choose **EC2 Spot Instances**.
3. For **Region**, select the required Region.
4. For **Primary instance type**, select the Spot Instance limit for which you want to request a limit increase.
5. For **New limit value**, enter the total number of vCPUs that you want to run concurrently. To determine the total number of vCPUs that you need, see [Amazon EC2 Instance Types](#) to find the number of vCPUs of each instance type.
6. (Conditional) You must create a separate limit request for each Spot Instance limit. To request an increase for another Spot Instance limit, choose **Add another request** and repeat steps 4 and 5 in this procedure.
7. For **Use case description**, enter your use case, and then choose **Submit**.

For more information about viewing limits and requesting a limit increase, see [Amazon EC2 service quotas \(p. 1241\)](#).

Monitoring Spot Instance limits and usage

You can view and manage your Spot Instance limits using the following:

- The [Limits page](#) in the Amazon EC2 console
- The Amazon EC2 [Services quotas page](#) in the Service Quotas console

- The [get-service-quota](#) AWS CLI

For more information, see [Amazon EC2 service quotas \(p. 1241\)](#) in the *Amazon EC2 User Guide for Linux Instances* and [Viewing a Service Quota](#) in the *Service Quotas User Guide*.

With Amazon CloudWatch metrics integration, you can monitor EC2 usage against limits. You can also configure alarms to warn about approaching limits. For more information, see [Using Amazon CloudWatch Alarms](#) in the *Service Quotas User Guide*.

Spot Fleet limits

The usual Amazon EC2 limits apply to instances launched by a Spot Fleet or an EC2 Fleet, such as Spot Instance limits and volume limits. In addition, the following limits apply:

- The number of active Spot Fleets and EC2 Fleets per Region: 1,000*
- The number of Spot Instance pools (unique combination of instance type and subnet): 300*
- The size of the user data in a launch specification: 16 KB*
- The target capacity per Spot Fleet or EC2 Fleet: 10,000
- The target capacity across all Spot Fleets and EC2 Fleets in a Region: 100,000
- A Spot Fleet request or an EC2 Fleet request can't span Regions.
- A Spot Fleet request or an EC2 Fleet request can't span different subnets from the same Availability Zone.

If you need more than the default limits for target capacity, complete the AWS Support Center [Create case](#) form to request a limit increase. For **Limit type**, choose **EC2 Fleet**, choose a Region, and then choose **Target Fleet Capacity per Fleet (in units)** or **Target Fleet Capacity per Region (in units)**, or both.

* These are hard limits. You cannot request a limit increase for these limits.

Burstable performance instances

If you launch your Spot Instances using a [burstable performance instance type \(p. 219\)](#), and if you plan to use your burstable performance Spot Instances immediately and for a short duration, with no idle time for accruing CPU credits, we recommend that you launch them in [Standard mode \(p. 231\)](#) to avoid paying higher costs. If you launch burstable performance Spot Instances in [Unlimited mode \(p. 223\)](#) and burst CPU immediately, you'll spend surplus credits for bursting. If you use the instance for a short duration, the instance doesn't have time to accrue CPU credits to pay down the surplus credits, and you are charged for the surplus credits when you terminate the instance.

Unlimited mode is suitable for burstable performance Spot Instances only if the instance runs long enough to accrue CPU credits for bursting. Otherwise, paying for surplus credits makes burstable performance Spot Instances more expensive than using other instances. For more information, see [When to use unlimited mode versus fixed CPU \(p. 224\)](#).

Launch credits are meant to provide a productive initial launch experience for T2 instances by providing sufficient compute resources to configure the instance. Repeated launches of T2 instances to access new launch credits is not permitted. If you require sustained CPU, you can earn credits (by idling over some period), use [Unlimited mode \(p. 223\)](#) for T2 Spot Instances, or use an instance type with dedicated CPU.

Dedicated Hosts

An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated to your use. Dedicated Hosts allow you to use your existing per-socket, per-core, or per-VM software licenses, including Windows Server, Microsoft SQL Server, SUSE, and Linux Enterprise Server.

For information about the configurations supported on Dedicated Hosts, see the [Dedicated Hosts Configuration Table](#).

Contents

- Differences between Dedicated Hosts and Dedicated Instances (p. 439)
- Bring your own license (p. 439)
- Dedicated Host instance capacity (p. 440)
- Dedicated Hosts restrictions (p. 440)
- Pricing and billing (p. 441)
- Working with Dedicated Hosts (p. 442)
- Working with shared Dedicated Hosts (p. 459)
- Host recovery (p. 464)
- Tracking configuration changes (p. 468)

Differences between Dedicated Hosts and Dedicated Instances

Dedicated Hosts and Dedicated Instances can both be used to launch Amazon EC2 instances onto physical servers that are dedicated for your use.

There are no performance, security, or physical differences between Dedicated Instances and instances on Dedicated Hosts. However, there are some differences between the two. The following table highlights some of the key differences between Dedicated Hosts and Dedicated Instances:

	Dedicated Host	Dedicated Instance
Billing	Per-host billing	Per-instance billing
Visibility of sockets, cores, and host ID	Provides visibility of the number of sockets and physical cores	No visibility
Host and instance affinity	Allows you to consistently deploy your instances to the same physical server over time	Not supported
Targeted instance placement	Provides additional visibility and control over how instances are placed on a physical server	Not supported
Automatic instance recovery	Supported. For more information, see Host recovery (p. 464) .	Supported
Bring Your Own License (BYOL)	Supported	Not supported

Bring your own license

Dedicated Hosts allow you to use your existing per-socket, per-core, or per-VM software licenses. When you bring your own license, you are responsible for managing your own licenses. However, Amazon EC2 has features that help you maintain license compliance, such as instance affinity and targeted placement.

These are the general steps to follow in order to bring your own volume licensed machine image into Amazon EC2.

1. Verify that the license terms controlling the use of your machine images allow usage in a virtualized cloud environment.
2. After you have verified that your machine image can be used within Amazon EC2, import it using VM Import/Export. For information about how to import your machine image, see the [VM Import/Export User Guide](#).
3. After you import your machine image, you can launch instances from it onto active Dedicated Hosts in your account.
4. When you run these instances, depending on the operating system, you might be required to activate these instances against your own KMS server.

Note

To track how your images are used in AWS, enable host recording in AWS Config. You can use AWS Config to record configuration changes to a Dedicated Host and use the output as a data source for license reporting. For more information, see [Tracking configuration changes \(p. 468\)](#).

Dedicated Host instance capacity

Support for multiple instance types on the same Dedicated Host is available for the following instance families: c5, m5, r5, c5n, r5n, and m5n. For example, when you allocate an r5 Dedicated Host, you can use a host with 2 sockets and 48 physical cores on which you can run different instance types, such as r5.2xlarge and r5.4xlarge. You can run any number of instances up to the core capacity associated with the host. For example, the table below shows the different instance type combinations you can run on a Dedicated Host.

Instance family	Example instance type combinations
R5	<ul style="list-style-type: none">• Example 1: 4 x r5.4xlarge + 4 x r5.2xlarge• Example 2: 1 x r5.12xlarge + 1 x r5.4xlarge + 1 x r5.2xlarge + 5 x r5.xlarge + 2 x r5.large
C5	<ul style="list-style-type: none">• Example 1: 1 x c5.9xlarge + 2 x c5.4xlarge + 1 x c5.xlarge• Example 2: 4 x c5.4xlarge + 1 x c5.xlarge + 2 x c5.large
M5	<ul style="list-style-type: none">• Example 1: 4 x m5.4xlarge + 4 x m5.2xlarge• Example 2: 1 x m5.12xlarge + 1 x m5.4xlarge + 1 x m5.2xlarge + 5 x m5.xlarge + 2 x m5.large

Other instance families support only a single instance type on the same Dedicated Host. For more information about the instance families and instance type configurations supported on Dedicated Hosts see [Amazon EC2 Dedicated Host Pricing](#).

Dedicated Hosts restrictions

Before you allocate Dedicated Hosts, take note of the following limitations and restrictions:

- To run RHEL, SUSE Linux, and SQL Server on Dedicated Hosts, you must bring your own AMIs. RHEL, SUSE Linux, and SQL Server AMIs that are offered by AWS or that are available on AWS Marketplace

can't be used with Dedicated Hosts. For more information on how to create your own AMI, see [Bring your own license \(p. 439\)](#).

- Up to two On-Demand Dedicated Hosts per instance family, per Region can be allocated. It is possible to request a limit increase: [Request to Raise Allocation Limit on Amazon EC2 Dedicated Hosts](#).
- The instances that run on a Dedicated Host can only be launched in a VPC.
- Auto Scaling groups are supported when using a launch template that specifies a host resource group. For more information, see [Creating a Launch Template for an Auto Scaling Group](#) in the *Amazon EC2 Auto Scaling User Guide*.
- Amazon RDS instances are not supported.
- The AWS Free Usage tier is not available for Dedicated Hosts.
- Instance placement control refers to managing instance launches onto Dedicated Hosts. You cannot launch Dedicated Hosts into placement groups.

Pricing and billing

The price for a Dedicated Host varies by payment option.

Payment Options

- [On-Demand Dedicated Hosts \(p. 441\)](#)
- [Dedicated Host Reservations \(p. 441\)](#)
- [Savings Plans \(p. 442\)](#)
- [Pricing for Windows Server on Dedicated Hosts \(p. 442\)](#)

On-Demand Dedicated Hosts

On-Demand billing is automatically activated when you allocate a Dedicated Host to your account.

The On-Demand price for a Dedicated Host varies by instance family and Region. You pay per second (with a minimum of 60 seconds) for active Dedicated Host, regardless of the quantity or the size of instances that you choose to launch on it. For more information about On-Demand pricing, see [Amazon EC2 Dedicated Hosts On-Demand Pricing](#).

You can release an On-Demand Dedicated Host at any time to stop accruing charges for it. For information about releasing a Dedicated Host, see [Releasing Dedicated Hosts \(p. 455\)](#).

Dedicated Host Reservations

Dedicated Host Reservations provide a billing discount compared to running On-Demand Dedicated Hosts. Reservations are available in three payment options:

- **No Upfront**—No Upfront Reservations provide you with a discount on your Dedicated Host usage over a term and do not require an upfront payment. Available for a one-year term only.
- **Partial Upfront**—A portion of the reservation must be paid upfront and the remaining hours in the term are billed at a discounted rate. Available in one-year and three-year terms.
- **All Upfront**—Provides the lowest effective price. Available in one-year and three-year terms and covers the entire cost of the term upfront, with no additional future charges.

You must have active Dedicated Hosts in your account before you can purchase reservations. Each reservation covers a single, specific Dedicated Host in your account. Reservations are applied to the instance family on the host, not the instance size. If you have three Dedicated Hosts with different instance sizes (`m4.xlarge`, `m4.medium`, and `m4.large`) you can associate a single `m4` reservation with all those Dedicated Hosts. The instance family and Region of the reservation must match that of the Dedicated Hosts you want to associate it with.

When a reservation is associated with a Dedicated Host, the Dedicated Host can't be released until the reservation's term is over.

For more information about reservation pricing, see [Amazon EC2 Dedicated Hosts Pricing](#).

Savings Plans

Savings Plans are a flexible pricing model that offers significant savings over On-Demand Instances. With Savings Plans, you make a commitment to a consistent amount of usage, in USD per hour, for a term of one or three years. This provides you with the flexibility to use the Dedicated Hosts that best meet your needs and continue to save money, instead of making a commitment to a specific Dedicated Host. For more information, see the [AWS Savings Plans User Guide](#).

Pricing for Windows Server on Dedicated Hosts

Subject to Microsoft licensing terms, you can bring your existing Windows Server and SQL Server licenses to Dedicated Hosts. There is no additional charge for software usage if you choose to bring your own licenses.

In addition, you can also use Windows Server AMIs provided by Amazon to run the latest versions of Windows Server on Dedicated Hosts. This is common for scenarios where you have existing SQL Server licenses eligible to run on Dedicated Hosts, but need Windows Server to run the SQL Server workload. Windows Server AMIs provided by Amazon are supported on [current generation instance types \(p. 201\)](#) only. For more information, see [Amazon EC2 Dedicated Hosts Pricing](#).

Working with Dedicated Hosts

To use a Dedicated Host, you first allocate hosts for use in your account. You then launch instances onto the hosts by specifying *host tenancy* for the instance. You must select a specific host for the instance to launch on to, or you can allow it to launch on to any host that has auto-placement enabled and matches its instance type. When an instance is stopped and restarted, the *Host affinity* setting determines whether it's restarted on the same, or a different, host.

If you no longer need an On-Demand host, you can stop the instances running on the host, direct them to launch on a different host, and then *release* the host.

Dedicated Hosts are also integrated with AWS License Manager. With License Manager, you can create a host resource group, which is a collection of Dedicated Hosts that are managed as a single entity. When creating a host resource group, you specify the host management preferences, such as auto-allocate and auto-release, for the Dedicated Hosts. This allows you to launch instances onto Dedicated Hosts without manually allocating and managing those hosts. For more information, see [Host Resource Groups](#) in the [AWS License Manager User Guide](#).

Contents

- [Allocating Dedicated Hosts \(p. 443\)](#)
- [Launching instances onto a Dedicated Host \(p. 445\)](#)
- [Launching instances into a host resource group \(p. 447\)](#)
- [Understanding auto-placement and affinity \(p. 448\)](#)
- [Modifying Dedicated Host auto-placement \(p. 448\)](#)
- [Modifying the supported instance types \(p. 449\)](#)
- [Modifying instance tenancy and affinity \(p. 451\)](#)
- [Viewing Dedicated Hosts \(p. 452\)](#)
- [Tagging Dedicated Hosts \(p. 453\)](#)
- [Monitoring Dedicated Hosts \(p. 454\)](#)

- [Releasing Dedicated Hosts \(p. 455\)](#)
- [Purchasing Dedicated Host Reservations \(p. 456\)](#)
- [Viewing Dedicated Host reservations \(p. 458\)](#)
- [Tagging Dedicated Host Reservations \(p. 459\)](#)

Allocating Dedicated Hosts

To begin using Dedicated Hosts, you must allocate Dedicated Hosts in your account using the Amazon EC2 console or the command line tools. After you allocate the Dedicated Host, the Dedicated Host capacity is made available in your account immediately and you can start launching instances onto the Dedicated Host.

Support for multiple instance types on the same Dedicated Host is available for the following instance families: c5, m5, r5, c5n, r5n, and m5n. Other instance families support only a single instance type on the same Dedicated Host.

You can allocate a Dedicated Host using the following methods.

New console

To allocate a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts** and then choose **Allocate Dedicated Host**.
3. For **Instance family**, choose the instance family for the Dedicated Host.
4. Specify whether the Dedicated Host supports multiple instance types within the selected instance family, or a specific instance type only. Do one of the following.
 - To configure the Dedicated Host to support multiple instance types in the selected instance family, for **Support multiple instance types**, choose **Enable**. Enabling this allows you to launch different instance types from the same instance family onto the Dedicated Host. For example, if you choose the m5 instance family and choose this option, you can launch m5.xlarge and m5.4xlarge instances onto the Dedicated Host.
 - To configure the Dedicated Host to support a single instance type within the selected instance family, clear **Support multiple instance types**, and then for **Instance type**, choose the instance type to support. This allows you to launch a single instance type on the Dedicated Host. For example, if you choose this option and specify m5.4xlarge as the supported instance type, you can launch only m5.4xlarge instances onto the Dedicated Host.
5. For **Availability Zone**, choose the Availability Zone in which to allocate the Dedicated Host.
6. To allow the Dedicated Host to accept untargeted instance launches that match its instance type, for **Instance auto-placement**, choose **Enable**. For more information about auto-placement, see [Understanding auto-placement and affinity \(p. 448\)](#).
7. To enable host recovery for the Dedicated Host, for **Host recovery**, choose **Enable**. For more information, see [Host recovery \(p. 464\)](#).
8. For **Quantity**, enter the number of Dedicated Hosts to allocate.
9. (Optional) Choose **Add Tag** and enter a tag key and a tag value.
10. Choose **Allocate**.

Old console

To allocate a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Dedicated Hosts, Allocate Dedicated Host**.
3. For **Instance family**, choose the instance family for the Dedicated Host.
4. Specify whether the Dedicated Host supports multiple instance types within the selected instance family, or a specific instance type only. Do one of the following.
 - To configure the Dedicated Host to support multiple instance types in the selected instance family, select **Support multiple instance types**. Enabling this allows you to launch different instance types from the same instance family onto the Dedicated Host. For example, if you choose the `m5` instance family and choose this option, you can launch `m5.xlarge` and `m5.4xlarge` instances onto the Dedicated Host. The instance family must be powered by the Nitro System.
 - To configure the Dedicated Host to support a single instance type within the selected instance family, clear **Support multiple instance types**, and then for **Instance type**, choose the instance type to support. This allows you to launch a single instance type on the Dedicated Host. For example, if you choose this option and specify `m5.4xlarge` as the supported instance type, you can launch only `m5.4xlarge` instances onto the Dedicated Host.
5. For **Availability Zone**, choose the Availability Zone in which to allocate the Dedicated Host.
6. To allow the Dedicated Host to accept untargeted instance launches that match its instance type, for **Instance auto-placement**, choose **Enable**. For more information about auto-placement, see [Understanding auto-placement and affinity \(p. 448\)](#).
7. To enable host recovery for the Dedicated Host, for **Host recovery** choose **Enable**. For more information, see [Host recovery \(p. 464\)](#).
8. For **Quantity**, enter the number of Dedicated Hosts to allocate.
9. (Optional) Choose **Add Tag** and enter a tag key and a tag value.
10. Choose **Allocate host**.

AWS CLI

To allocate a Dedicated Host

Use the [allocate-hosts](#) AWS CLI command. The following command allocates a Dedicated Host that supports multiple instance types from the `m5` instance family in `us-east-1a` Availability Zone. The host also has host recovery enabled and it has auto-placement disabled.

```
aws ec2 allocate-hosts --instance-family "m5" --availability-zone "us-east-1a" --auto-placement "off" --host-recovery "on" --quantity 1
```

The following command allocates a Dedicated Host that supports *untargeted* `m4.large` instance launches in the `eu-west-1a` Availability Zone, enables host recovery, and applies a tag with a key of `purpose` and a value of `production`.

```
aws ec2 allocate-hosts --instance-type "m4.large" --availability-zone "eu-west-1a" --auto-placement "on" --host-recovery "on" --quantity 1 --tag-specifications 'ResourceType=dedicated-host,Tags=[{Key=purpose,Value=production}]'
```

PowerShell

To allocate a Dedicated Host

Use the [New-EC2Host](#) AWS Tools for Windows PowerShell command. The following command allocates a Dedicated Host that supports multiple instance types from the `m5` instance family in `us-east-1a` Availability Zone. The host also has host recovery enabled and it has auto-placement disabled.

```
PS C:\> New-EC2Host -InstanceFamily m5 -AvailabilityZone us-east-1a -AutoPlacement Off  
-HostRecovery On -Quantity 1
```

The following commands allocate a Dedicated Host that supports *untargeted* m4.large instance launches in the eu-west-1a Availability Zone, enable host recovery, and apply a tag with a key of purpose and a value of production.

The TagSpecification parameter used to tag a Dedicated Host on creation requires an object that specifies the type of resource to be tagged, the tag key, and the tag value. The following commands create the required object.

```
PS C:\> $tag = @{ Key="purpose"; Value="production" }  
PS C:\> $tagspec = new-object Amazon.EC2.Model.TagSpecification  
PS C:\> $tagspec.ResourceType = "dedicated-host"  
PS C:\> $tagspec.Tags.Add($tag)
```

The following command allocates the Dedicated Host and applies the tag specified in the \$tagspec object.

```
PS C:\> New-EC2Host -InstanceType m4.large -AvailabilityZone eu-west-1a -  
AutoPlacement On -HostRecovery On -Quantity 1 -TagSpecification $tagspec
```

Launching instances onto a Dedicated Host

After you have allocated a Dedicated Host, you can launch instances onto it. You can't launch instances with host tenancy if you do not have active Dedicated Hosts with enough available capacity for the instance type that you are launching.

Note

The instances launched onto Dedicated Hosts can only be launched in a VPC. For more information, see [Introduction to VPC](#).

Before you launch your instances, take note of the limitations. For more information, see [Dedicated Hosts restrictions \(p. 440\)](#).

You can launch an instance onto a Dedicated Host using the following methods.

Console

To launch an instance onto a specific Dedicated Host from the Dedicated Hosts page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Dedicated Hosts** in the navigation pane.
3. On the **Dedicated Hosts** page, select a host and choose **Actions, Launch Instance(s) onto Host**.
4. Select an AMI from the list. SQL Server, SUSE, and RHEL AMIs provided by Amazon EC2 can't be used with Dedicated Hosts.
5. On the **Choose an Instance Type** page, select the instance type to launch and then choose **Next: Configure Instance Details**.

If the Dedicated Host supports a single instance type only, the supported instance type is selected by default and can't be changed.

If the Dedicated Host supports multiple instance types, you must select an instance type within the supported instance family based on the available instance capacity of the Dedicated Host. We recommend that you launch the larger instance sizes first, and then fill the remaining instance capacity with the smaller instance sizes as needed.

6. On the **Configure Instance Details** page, configure the instance settings to suit your needs, and then for **Affinity**, choose one of the following options:
 - **Off**—The instance launches onto the specified host, but it is not guaranteed to restart on the same Dedicated Host if stopped.
 - **Host**—If stopped, the instance always restarts on this specific host.

For more information about Affinity, see [Understanding auto-placement and affinity \(p. 448\)](#).

The **Tenancy** and **Host** options are pre-configured based on the host that you selected.

7. Choose **Review and Launch**.
8. On the **Review Instance Launch** page, choose **Launch**.
9. When prompted, select an existing key pair or create a new one, and then choose **Launch Instances**.

To launch an instance onto a Dedicated Host using the Launch Instance wizard

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances, Launch Instance**.
3. Select an AMI from the list. SQL Server, SUSE, and RHEL AMIs provided by Amazon EC2 can't be used with Dedicated Hosts.
4. Select the type of instance to launch and choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, configure the instance settings to suit your needs, and then configure the following settings, which are specific to a Dedicated Host:
 - Tenancy—Choose **Dedicated Host - Launch this instance on a Dedicated Host**.
 - Host—Choose either **Use auto-placement** to launch the instance on any Dedicated Host that has auto-placement enabled, or select a specific Dedicated Host in the list. The list displays only Dedicated Hosts that support the selected instance type.
 - Affinity—Choose one of the following options:
 - **Off**—The instance launches onto the specified host, but it is not guaranteed to restart on it if stopped.
 - **Host**—If stopped, the instance always restarts on the specified host.

For more information, see [Understanding auto-placement and affinity \(p. 448\)](#).

If you are unable to see these settings, check that you have selected a VPC in the **Network** menu.

6. Choose **Review and Launch**.
7. On the **Review Instance Launch** page, choose **Launch**.
8. When prompted, select an existing key pair or create a new one, and then choose **Launch Instances**.

AWS CLI

To launch an instance onto a Dedicated Host

Use the [run-instances](#) AWS CLI command and specify the instance affinity, tenancy, and host in the `Placement request` parameter.

PowerShell

To launch an instance onto a Dedicated Host

Use the [New-EC2Instance](#) AWS Tools for Windows PowerShell command and specify the instance affinity, tenancy, and host in the `Placement` request parameter.

Launching instances into a host resource group

When you launch an instance into a host resource group that has a Dedicated Host with available instance capacity, Amazon EC2 launches the instance onto that host. If the host resource group does not have a host with available instance capacity, Amazon EC2 automatically allocates a new host in the host resource group, and then launches the instance onto that host. For more information, see [Host Resource Groups](#) in the *AWS License Manager User Guide*.

Requirements and limits

- You must associate a core- or socket-based license configuration with the AMI.
- You can't use SQL Server, SUSE, or RHEL AMIs provided by Amazon EC2 with Dedicated Hosts.
- You can't target a specific host by choosing a host ID, and you can't enable instance affinity when launching an instance into a host resource group.

You can launch an instance into a host resource group using the following methods.

Console

To launch an instance into a host resource group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, **Launch Instance**.
3. Select an AMI.
4. Select the type of instance to launch and choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, configure the instance settings to suit your needs, and then do the following:
 - a. For **Tenancy**, choose **Dedicated Host**.
 - b. For **Host resource group**, choose **Launch instance into a host resource group**.
 - c. For **Host resource group name**, choose the host resource group in which to launch the instance.
6. Choose **Review and Launch**.
7. On the **Review Instance Launch** page, choose **Launch**.
8. When prompted, select an existing key pair or create a new one, and then choose **Launch Instances**.

AWS CLI

To launch an instance into a host resource group

Use the [run-instances](#) AWS CLI command, and in the `Placement` request parameter, omit the `Tenancy` option and specify the host resource group ARN.

PowerShell

To launch an instance into a host resource group

Use the [New-EC2Instance](#) AWS Tools for Windows PowerShell command, and in the `Placement` request parameter, omit the `Tenancy` option and specify the host resource group ARN.

Understanding auto-placement and affinity

Placement control for Dedicated Hosts happens on both the instance level and host level.

Auto-placement

Auto-placement is configured at the host level. It allows you to manage whether instances that you launch are launched onto a specific host, or onto any available host that has matching configurations.

When the auto-placement of a Dedicated Host is *disabled*, it only accepts *Host tenancy* instance launches that specify its unique host ID. This is the default setting for new Dedicated Hosts.

When the auto-placement of a Dedicated Host is *enabled*, it accepts any untargeted instance launches that match its instance type configuration.

When launching an instance, you need to configure its tenancy. Launching an instance onto a Dedicated Host without providing a specific `HostId` enables it to launch on any Dedicated Host that has auto-placement *enabled* and that matches its instance type.

Host affinity

Host affinity is configured at the instance level. It establishes a launch relationship between an instance and a Dedicated Host.

When affinity is set to `Host`, an instance launched onto a specific host always restarts on the same host if stopped. This applies to both targeted and untargeted launches.

When affinity is set to `Off`, and you stop and restart the instance, it can be restarted on any available host. However, it tries to launch back onto the last Dedicated Host on which it ran (on a best-effort basis).

Modifying Dedicated Host auto-placement

You can modify the auto-placement settings of a Dedicated Host after you have allocated it to your AWS account, using one of the following methods.

New console

To modify the auto-placement of a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Select a host and choose **Actions, Modify host**.
4. For **Instance auto-placement**, choose **Enable** to enable auto-placement, or clear **Enable** to disable auto-placement. For more information, see [Understanding auto-placement and affinity \(p. 448\)](#).
5. Choose **Save**.

Old console

To modify the auto-placement of a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Dedicated Hosts** in the navigation pane.
3. On the **Dedicated Hosts** page, select a host and choose **Actions, Modify Auto-Placement**.
4. On the Modify Auto-placement window, for **Allow instance auto-placement**, choose **Yes** to enable auto-placement, or choose **No** to disable auto-placement. For more information, see [Understanding auto-placement and affinity \(p. 448\)](#).

5. Choose **Save**.

AWS CLI

To modify the auto-placement of a Dedicated Host

Use the [modify-hosts](#) AWS CLI command. The following example enables auto-placement for the specified Dedicated Host.

```
aws ec2 modify-hosts --auto-placement on --host-ids h-012a3456b7890cdef
```

PowerShell

To modify the auto-placement of a Dedicated Host

Use the [Edit-EC2Host](#) AWS Tools for Windows PowerShell command. The following example enables auto-placement for the specified Dedicated Host.

```
PS C:\> Edit-EC2Host --AutoPlacement 1 --HostId h-012a3456b7890cdef
```

Modifying the supported instance types

Support for multiple instance types on the same Dedicated Host is available for the following instance families: `c5`, `m5`, `r5`, `c5n`, `r5n`, and `m5n`. Other instance families support only a single instance type on the same Dedicated Host.

You can allocate a Dedicated Host using the following methods.

You can modify a Dedicated Host to change the instance types that it supports. If it currently supports a single instance type, you can modify it to support multiple instance types within that instance family. Similarly, if it currently supports multiple instance types, you can modify it to support a specific instance type only.

To modify a Dedicated Host to support multiple instance types, you must first stop all running instances on the host. The modification takes approximately 10 minutes to complete. The Dedicated Host transitions to the pending state while the modification is in progress. You can't start stopped instances or launch new instances on the Dedicated Host while it is in the pending state.

To modify a Dedicated Host that supports multiple instance types to support only a single instance type, the host must either have no running instances, or the running instances must be of the instance type that you want the host to support. For example, to modify a host that supports multiple instance types in the `m5` instance family to support only `m5.1.large` instances, the Dedicated Host must either have no running instances, or it must have only `m5.1.large` instances running on it.

You can modify the supported instance types using one of the following methods.

New console

To modify the supported instance types for a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the Navigation pane, choose **Dedicated Host**.
3. Select the Dedicated Host to modify and choose **Actions, Modify host**.
4. Do one of the following, depending on the current configuration of the Dedicated Host:
 - If the Dedicated Host currently supports a specific instance type, **Support multiple instance types** is not enabled, and **Instance type** lists the supported instance type. To modify the host

to support multiple types in the current instance family, for **Support multiple instance types**, choose **Enable**.

You must first stop all instances running on the host before modifying it to support multiple instance types.

- If the Dedicated Host currently supports multiple instance types in an instance family, **Enabled** is selected for **Support multiple instance types**. To modify the host to support a specific instance type, for **Support multiple instance types**, clear **Enable**, and then for **Instance type**, select the specific instance type to support.

You can't change the instance family supported by the Dedicated Host.

5. Choose **Save**.

Old console

To modify the supported instance types for a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 2. In the Navigation pane, choose **Dedicated Host**.
 3. Select the Dedicated Host to modify and choose **Actions, Modify Supported Instance Types**.
 4. Do one of the following, depending on the current configuration of the Dedicated Host:
 - If the Dedicated Host currently supports a specific instance type, **No** is selected for **Support multiple instance types**. To modify the host to support multiple types in the current instance family, for **Support multiple instance types**, select **Yes**.
- You must first stop all instances running on the host before modifying it to support multiple instance types.
- If the Dedicated Host currently supports multiple instance types in an instance family, **Yes** is selected for **Support multiple instance types**, and **Instance family** displays the supported instance family. To modify the host to support a specific instance type, for **Support multiple instance types**, select **No**, and then for **Instance type**, select the specific instance type to support.
- You can't change the instance family supported by the Dedicated Host.

5. Choose **Save**.

AWS CLI

To modify the supported instance types for a Dedicated Host

Use the [modify-hosts](#) AWS CLI command.

The following command modifies a Dedicated Host to support multiple instance types within the **m5** instance family.

```
aws ec2 modify-hosts --instance-family m5 --host-ids h-012a3456b7890cdef
```

The following command modifies a Dedicated Host to support **m5.xlarge** instances only.

```
aws ec2 modify-hosts --instance-type m5.xlarge --instance-family --host-ids h-012a3456b7890cdef
```

PowerShell

To modify the supported instance types for a Dedicated Host

Use the [Edit-EC2Host](#) AWS Tools for Windows PowerShell command.

The following command modifies a Dedicated Host to support multiple instance types within the `m5` instance family.

```
PS C:\> Edit-EC2Host --InstanceFamily m5 --HostId h-012a3456b7890cdef
```

The following command modifies a Dedicated Host to support `m5.xlarge` instances only.

```
PS C:\> Edit-EC2Host --InstanceType m5.xlarge --HostId h-012a3456b7890cdef
```

Modifying instance tenancy and affinity

You can change the tenancy of an instance from dedicated to host, or from host to dedicated, after you have launched it. You can also modify the affinity between the instance and the host. To modify either instance tenancy or affinity, the instance must be in the stopped state.

You can modify an instance's tenancy and affinity using the following methods.

Console

To modify instance tenancy or affinity

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Instances**, and select the instance to modify.
3. Choose **Actions**, **Instance State**, and **Stop**.
4. Open the context (right-click) menu on the instance and choose **Instance Settings**, **Modify Instance Placement**.
5. On the **Modify Instance Placement** page, configure the following:
 - **Tenancy**—Choose one of the following:
 - Run a dedicated hardware instance—Launches the instance as a Dedicated Instance. For more information, see [Dedicated Instances \(p. 469\)](#).
 - Launch the instance on a Dedicated Host—Launches the instance onto a Dedicated Host with configurable affinity.
 - **Affinity**—Choose one of the following:
 - This instance can run on any one of my hosts—The instance launches onto any available Dedicated Host in your account that supports its instance type.
 - This instance can only run on the selected host—The instance is only able to run on the Dedicated Host selected for **Target Host**.
 - **Target Host**—Select the Dedicated Host that the instance must run on. If no target host is listed, you might not have available, compatible Dedicated Hosts in your account.

For more information, see [Understanding auto-placement and affinity \(p. 448\)](#).

6. Choose **Save**.

AWS CLI

To modify instance tenancy or affinity

Use the [modify-instance-placement](#) AWS CLI command. The following example changes the specified instance's affinity from default to host, and specifies the Dedicated Host that the instance has affinity with.

```
aws ec2 modify-instance-placement --instance-id i-1234567890abcdef0 --affinity host --  
host-id h-012a3456b7890cdef
```

PowerShell

To modify instance tenancy or affinity

Use the [Edit-EC2InstancePlacement](#) AWS Tools for Windows PowerShell command. The following example changes the specified instance's affinity from default to host, and specifies the Dedicated Host that the instance has affinity with.

```
PS C:\> Edit-EC2InstancePlacement -InstanceId i-1234567890abcdef0 -Affinity host -  
HostId h-012a3456b7890cdef
```

Viewing Dedicated Hosts

You can view details about a Dedicated Host and the individual instances on it using the following methods.

New console

To view the details of a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. On the **Dedicated Hosts** page, select a host.
4. For information about the host, choose **Details**.

Available vCPUs indicates the vCPUs that are available on the Dedicated Host for new instance launches. For example, a Dedicated Host that supports multiple instance types within the c5 instance family, and that has no instances running on it, has 72 available vCPUs. This means that you can launch different combinations of instance types onto the Dedicated Host to consume the 72 available vCPUs.

For information about instances running on the host, choose **Running instances**.

Old console

To view the details of a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. On the **Dedicated Hosts** page, select a host.
4. For information about the host, choose **Description**. **Available vCPUs** indicates the vCPUs that are available on the Dedicated Host for new instance launches. For example, a Dedicated Host that supports multiple instance types within the c5 instance family, and that has no instances running on it, has 72 available vCPUs. This means that you can launch different combinations of instance types onto the Dedicated Host to consume the 72 available vCPUs.

For information about instances running on the host, choose **Instances**.

AWS CLI

To view the capacity of a Dedicated Host

Use the [describe-hosts](#) AWS CLI command.

The following example uses the [describe-hosts](#) (AWS CLI) command to view the available instance capacity for a Dedicated Host that supports multiple instance types within the c5 instance family. The Dedicated Host already has two c5.4xlarge instances and four c5.2xlarge instances running on it.

```
$ aws ec2 describe-hosts --host-id h-012a3456b7890cdef
```

```
"AvailableInstanceCapacity": [  
    { "AvailableCapacity": 2,  
      "InstanceType": "c5.xlarge",  
      "TotalCapacity": 18 },  
    { "AvailableCapacity": 4,  
      "InstanceType": "c5.large",  
      "TotalCapacity": 36 }  
],  
"AvailableVCpus": 8
```

PowerShell

To view the instance capacity of a Dedicated Host

Use the [Get-EC2Host](#) AWS Tools for Windows PowerShell command.

```
PS C:\> Get-EC2Host -HostId h-012a3456b7890cdef
```

Tagging Dedicated Hosts

You can assign custom tags to your existing Dedicated Hosts to categorize them in different ways, for example, by purpose, owner, or environment. This helps you to quickly find a specific Dedicated Host based on the custom tags that you assigned. Dedicated Host tags can also be used for cost allocation tracking.

You can also apply tags to Dedicated Hosts at the time of creation. For more information, see [Allocating Dedicated Hosts \(p. 443\)](#).

You can tag a Dedicated Host using the following methods.

New console

To tag a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host to tag, and then choose **Actions, Manage tags**.
4. In the **Manage tags** screen, choose **Add tag**, and then specify the key and value for the tag.
5. (Optional) Choose **Add tag** to add additional tags to the Dedicated Host.
6. Choose **Save changes**.

Old console

To tag a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host to tag, and then choose **Tags**.
4. Choose **Add/Edit Tags**.
5. In the **Add/Edit Tags** dialog box, choose **Create Tag**, and then specify the key and value for the tag.
6. (Optional) Choose **Create Tag** to add additional tags to the Dedicated Host.
7. Choose **Save**.

AWS CLI

To tag a Dedicated Host

Use the [create-tags](#) AWS CLI command.

The following command tags the specified Dedicated Host with `Owner=TeamA`.

```
aws ec2 create-tags --resources h-abc12345678909876 --tags Key=Owner,Value=TeamA
```

PowerShell

To tag a Dedicated Host

Use the [New-EC2Tag](#) AWS Tools for Windows PowerShell command.

The `New-EC2Tag` command needs a `Tag` object, which specifies the key and value pair to be used for the Dedicated Host tag. The following commands create a `Tag` object named `$tag`, with a key and value pair of `Owner` and `TeamA` respectively.

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag
PS C:\> $tag.Key = "Owner"
PS C:\> $tag.Value = "TeamA"
```

The following command tags the specified Dedicated Host with the `$tag` object.

```
PS C:\> New-EC2Tag -Resource h-abc12345678909876 -Tag $tag
```

Monitoring Dedicated Hosts

Amazon EC2 constantly monitors the state of your Dedicated Hosts. Updates are communicated on the Amazon EC2 console. You can view information about a Dedicated Host using the following methods.

Console

To view the state of a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Locate the Dedicated Host in the list and review the value in the **State** column.

AWS CLI

To view the state of a Dedicated Host

Use the [describe-hosts](#) AWS CLI command and then review the state property in the hostSet response element.

```
aws ec2 describe-hosts --host-id h-012a3456b7890cdef
```

PowerShell

To view the state of a Dedicated Host

Use the [Get-EC2Host](#) AWS Tools for Windows PowerShell command and then review the state property in the hostSet response element.

```
PS C:\> Get-EC2Host -HostId h-012a3456b7890cdef
```

The following table explains the possible Dedicated Host states.

State	Description
available	AWS hasn't detected an issue with the Dedicated Host. No maintenance or repairs are scheduled. Instances can be launched onto this Dedicated Host.
released	The Dedicated Host has been released. The host ID is no longer in use. Released hosts can't be reused.
under-assessment	AWS is exploring a possible issue with the Dedicated Host. If action must be taken, you are notified via the AWS Management Console or email. Instances can't be launched onto a Dedicated Host in this state.
pending	The Dedicated Host cannot be used for new instance launches. It is either being modified to support multiple instance types (p. 449) , or a host recovery (p. 464) is in progress.
permanent-failure	An unrecoverable failure has been detected. You receive an eviction notice through your instances and by email. Your instances might continue to run. If you stop or terminate all instances on a Dedicated Host with this state, AWS retires the host. AWS does not restart instances in this state. Instances can't be launched onto Dedicated Hosts in this state.
released-permanent-failure	AWS permanently releases Dedicated Hosts that have failed and no longer have running instances on them. The Dedicated Host ID is no longer available for use.

Releasing Dedicated Hosts

Any running instances on the Dedicated Host must be stopped before you can release the host. These instances can be migrated to other Dedicated Hosts in your account so that you can continue to use them. These steps apply only to On-Demand Dedicated Hosts.

You can release a Dedicated Host using the following methods.

New console

To release a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Dedicated Hosts**.
3. On the **Dedicated Hosts** page, select the Dedicated Host to release.
4. Choose **Actions, Release host**.
5. To confirm, choose **Release**.

Old console

To release a Dedicated Host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Dedicated Hosts** in the navigation pane.
3. On the **Dedicated Hosts** page, select the Dedicated Host to release.
4. Choose **Actions, Release Hosts**.
5. Choose **Release** to confirm.

AWS CLI

To release a Dedicated Host

Use the [release-hosts](#) AWS CLI command.

```
aws ec2 release-hosts --host-ids h-012a3456b7890cdef
```

PowerShell

To release a Dedicated Host

Use the [Remove-EC2Hosts](#) AWS Tools for Windows PowerShell command.

```
PS C:\> Remove-EC2Hosts -HostId h-012a3456b7890cdef
```

After you release a Dedicated Host, you can't reuse the same host or host ID again, and you are no longer charged On-Demand billing rates for it. The state of the Dedicated Host is changed to `released`, and you are not able to launch any instances onto that host.

Note

If you have recently released Dedicated Hosts, it can take some time for them to stop counting towards your limit. During this time, you might experience `LimitExceeded` errors when trying to allocate new Dedicated Hosts. If this is the case, try allocating new hosts again after a few minutes.

The instances that were stopped are still available for use and are listed on the **Instances** page. They retain their host tenancy setting.

Purchasing Dedicated Host Reservations

You can purchase reservations using the following methods:

Console

To purchase reservations

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. Choose **Dedicated Hosts, Dedicated Host Reservations, Purchase Dedicated Host Reservation**.
3. On the **Purchase Dedicated Host Reservation** screen, you can search for available offerings using the default settings, or you can specify custom values for the following:
 - **Host instance family**—The options listed correspond with the Dedicated Hosts in your account that are not already assigned to a reservation.
 - **Availability Zone**—The Availability Zone of the Dedicated Hosts in your account that aren't already assigned to a reservation.
 - **Payment option**—The payment option for the offering.
 - **Term**—The term of the reservation, which can be one or three years.
4. Choose **Find offering** and select an offering that matches your requirements.
5. Choose the Dedicated Hosts to associate with the reservation, and then choose **Review**.
6. Review your order and choose **Order**.

AWS CLI

To purchase reservations

1. Use the [describe-host-reservation-offerings](#) AWS CLI command to list the available offerings that match your needs. The following example lists the offerings that support instances in the `m4` instance family and have a one-year term.

Note

The term is specified in seconds. A one-year term includes 31,536,000 seconds, and a three-year term includes 94,608,000 seconds.

```
aws ec2 describe-host-reservation-offerings --filter Name=instance-family,Values=m4  
--max-duration 31536000
```

The command returns a list of offerings that match your criteria. Note the `offeringId` of the offering to purchase.

2. Use the [purchase-host-reservation](#) AWS CLI command to purchase the offering and provide the `offeringId` noted in the previous step. The following example purchases the specified reservation and associates it with a specific Dedicated Host that is already allocated in the AWS account, and it applies a tag with a key of `purpose` and a value of `production`.

```
aws ec2 purchase-host-reservation --offering-id hro-03f707bf363b6b324 --  
host-id-set h-013abcd2a00cbd123 --tag-specifications 'ResourceType=host-  
reservation,Tags=[{Key=purpose,Value=production}]'
```

PowerShell

To purchase reservations

1. Use the [Get-EC2HostReservationOffering](#) AWS Tools for Windows PowerShell command to list the available offerings that match your needs. The following examples list the offerings that support instances in the `m4` instance family and have a one-year term.

Note

The term is specified in seconds. A one-year term includes 31,536,000 seconds, and a three-year term includes 94,608,000 seconds.

```
PS C:\> $filter = @{Name="instance-family"; Value="m4"}
```

```
PS C:\> Get-EC2HostReservationOffering -filter $filter -MaxDuration 31536000
```

The command returns a list of offerings that match your criteria. Note the `offeringId` of the offering to purchase.

2. Use the `New-EC2HostReservation` AWS Tools for Windows PowerShell command to purchase the offering and provide the `offeringId` noted in the previous step. The following example purchases the specified reservation and associates it with a specific Dedicated Host that is already allocated in the AWS account.

```
PS C:\> New-EC2HostReservation -OfferingId hro-03f707bf363b6b324 -  
HostIdSet h-013abcd2a00cb123
```

Viewing Dedicated Host reservations

You can view information about the Dedicated Hosts that are associated with your reservation, including:

- The term of the reservation
- The payment option
- The start and end dates

You can view details of your Dedicated Host reservations using the following methods.

Console

To view the details of a Dedicated Host reservation

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Dedicated Hosts** in the navigation pane.
3. On the **Dedicated Hosts** page, choose **Dedicated Host Reservations**, and then select the reservation from the list provided.
4. Choose **Details** for information about the reservation.
5. Choose **Hosts** for information about the Dedicated Hosts with which the reservation is associated.

AWS CLI

To view the details of a Dedicated Host reservation

Use the `describe-host-reservations` AWS CLI command.

```
aws ec2 describe-host-reservations
```

PowerShell

To view the details of a Dedicated Host reservation

Use the `Get-EC2HostReservation` AWS Tools for Windows PowerShell command.

```
PS C:\> Get-EC2HostReservation
```

Tagging Dedicated Host Reservations

You can assign custom tags to your Dedicated Host Reservations to categorize them in different ways, for example, by purpose, owner, or environment. This helps you to quickly find a specific Dedicated Host Reservation based on the custom tags that you assigned.

You can tag a Dedicated Host Reservation using the command line tools only.

AWS CLI

To tag a Dedicated Host Reservation

Use the [create-tags](#) AWS CLI command.

```
aws ec2 create-tags --resources hr-1234563a4ffc669ae --tags Key=Owner,Value=TeamA
```

PowerShell

To tag a Dedicated Host Reservation

Use the [New-EC2Tag](#) AWS Tools for Windows PowerShell command.

The New-EC2Tag command needs a Tag parameter, which specifies the key and value pair to be used for the Dedicated Host Reservation tag. The following commands create the Tag parameter.

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag
PS C:\> $tag.Key = "Owner"
PS C:\> $tag.Value = "TeamA"
```

```
PS C:\> New-EC2Tag -Resource hr-1234563a4ffc669ae -Tag $tag
```

Working with shared Dedicated Hosts

Dedicated Host sharing enables Dedicated Host owners to share their Dedicated Hosts with other AWS accounts or within an AWS organization. This enables you to create and manage Dedicated Hosts centrally, and share the Dedicated Host across multiple AWS accounts or within your AWS organization.

In this model, the AWS account that owns the Dedicated Host (*owner*) shares it with other AWS accounts (*consumers*). Consumers can launch instances onto Dedicated Hosts that are shared with them in the same way that they would launch instances onto Dedicated Hosts that they allocate in their own account. The owner is responsible for managing the Dedicated Host and the instances that they launch onto it. Owners can't modify instances that consumers launch onto shared Dedicated Hosts. Consumers are responsible for managing the instances that they launch onto Dedicated Hosts shared with them. Consumers can't view or modify instances owned by other consumers or by the Dedicated Host owner, and they can't modify Dedicated Hosts that are shared with them.

A Dedicated Host owner can share a Dedicated Host with:

- Specific AWS accounts inside or outside of its AWS organization
- An organizational unit inside its AWS organization
- Its entire AWS organization

Contents

- [Prerequisites for sharing Dedicated Hosts \(p. 460\)](#)
- [Limitations for sharing Dedicated Hosts \(p. 460\)](#)

- [Related services \(p. 460\)](#)
- [Sharing across Availability Zones \(p. 460\)](#)
- [Sharing a Dedicated Host \(p. 461\)](#)
- [Unsharing a shared Dedicated Host \(p. 461\)](#)
- [Identifying a shared Dedicated Host \(p. 462\)](#)
- [Viewing instances running on a shared Dedicated Host \(p. 462\)](#)
- [Shared Dedicated Host permissions \(p. 463\)](#)
- [Billing and metering \(p. 463\)](#)
- [Dedicated Host limits \(p. 463\)](#)
- [Host recovery and Dedicated Host sharing \(p. 463\)](#)

Prerequisites for sharing Dedicated Hosts

- To share a Dedicated Host, you must own it in your AWS account. You can't share a Dedicated Host that has been shared with you.
- To share a Dedicated Host with your AWS organization or an organizational unit in your AWS organization, you must enable sharing with AWS Organizations. For more information, see [Enable Sharing with AWS Organizations](#) in the *AWS RAM User Guide*.

Limitations for sharing Dedicated Hosts

You can't share Dedicated Hosts that have been allocated for the following instance types: `u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal`, and `u-24tb1.metal`.

Related services

AWS Resource Access Manager

Dedicated Host sharing integrates with AWS Resource Access Manager (AWS RAM). AWS RAM is a service that enables you to share your AWS resources with any AWS account or through AWS Organizations. With AWS RAM, you share resources that you own by creating a *resource share*. A resource share specifies the resources to share, and the consumers with whom to share them. Consumers can be individual AWS accounts, or organizational units or an entire organization from AWS Organizations.

For more information about AWS RAM, see the [AWS RAM User Guide](#).

Sharing across Availability Zones

To ensure that resources are distributed across the Availability Zones for a Region, we independently map Availability Zones to names for each account. This could lead to Availability Zone naming differences across accounts. For example, the Availability Zone `us-east-1a` for your AWS account might not have the same location as `us-east-1a` for another AWS account.

To identify the location of your Dedicated Hosts relative to your accounts, you must use the *Availability Zone ID* (AZ ID). The Availability Zone ID is a unique and consistent identifier for an Availability Zone across all AWS accounts. For example, `use1-az1` is an Availability Zone ID for the `us-east-1` Region and it is the same location in every AWS account.

To view the Availability Zone IDs for the Availability Zones in your account

1. Open the AWS RAM console at <https://console.aws.amazon.com/ram>.
2. The Availability Zone IDs for the current Region are displayed in the **Your AZ ID** panel on the right-hand side of the screen.

Sharing a Dedicated Host

When an owner shares a Dedicated Host, it enables consumers to launch instances on the host. Consumers can launch as many instances onto the shared host as its available capacity allows.

Important

Note that you are responsible for ensuring that you have appropriate license rights to share any BYOL licenses on your Dedicated Hosts.

If you share a Dedicated Host with auto-placement enabled, keep the following in mind as it could lead to unintended Dedicated Host usage:

- If consumers launch instances with Dedicated Host tenancy and they do not have capacity on a Dedicated Host that they own in their account, the instance is automatically launched onto the shared Dedicated Host.

To share a Dedicated Host, you must add it to a resource share. A resource share is an AWS RAM resource that lets you share your resources across AWS accounts. A resource share specifies the resources to share, and the consumers with whom they are shared. You can add the Dedicated Host to an existing resource, or you can add it to a new resource share.

If you are part of an organization in AWS Organizations and sharing within your organization is enabled, consumers in your organization are automatically granted access to the shared Dedicated Host. Otherwise, consumers receive an invitation to join the resource share and are granted access to the shared Dedicated Host after accepting the invitation.

Note

After you share a Dedicated Host, it could take a few minutes for consumers to have access to it.

You can share a Dedicated Host that you own by using one of the following methods.

Amazon EC2 console

To share a Dedicated Host that you own using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Choose the Dedicated Host to share and choose **Actions, Share host**.
4. Select the resource share to which to add the Dedicated Host and choose **Share host**.

It could take a few minutes for consumers to get access to the shared host.

AWS RAM console

To share a Dedicated Host that you own using the AWS RAM console

See [Creating a Resource Share](#) in the *AWS RAM User Guide*.

AWS CLI

To share a Dedicated Host that you own using the AWS CLI

Use the [create-resource-share](#) command.

Unsharing a shared Dedicated Host

The Dedicated Host owner can unshare a shared Dedicated Host at any time. When you unshare a shared Dedicated Host, the following rules apply:

- Consumers with whom the Dedicated Host was shared can no longer launch new instances onto it.

- Instances owned by consumers that were running on the Dedicated Host at the time of unsharing continue to run but are scheduled for [retirement](#). Consumers receive retirement notifications for the instances and they have two weeks to take action on the notifications. However, if the Dedicated Host is reshared with the consumer within the retirement notice period, the instance retirements are cancelled.

To unshare a shared Dedicated Host that you own, you must remove it from the resource share. You can do this by using one of the following methods.

Amazon EC2 console

To unshare a shared Dedicated Host that you own using the Amazon EC2 console

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, choose **Dedicated Hosts**.
- Choose the Dedicated Host to unshare and choose the **Sharing** tab.
- The **Sharing** tab lists the resource shares to which the Dedicated Host has been added. Select the resource share from which to remove the Dedicated Host and choose **Remove host from resource share**.

AWS RAM console

To unshare a shared Dedicated Host that you own using the AWS RAM console

See [Updating a Resource Share](#) in the *AWS RAM User Guide*.

Command line

To unshare a shared Dedicated Host that you own using the AWS CLI

Use the [disassociate-resource-share](#) command.

Identifying a shared Dedicated Host

Owners and consumers can identify shared Dedicated Hosts using one of the following methods.

Amazon EC2 console

To identify a shared Dedicated Host using the Amazon EC2 console

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, choose **Dedicated Hosts**. The screen lists Dedicated Hosts that you own and Dedicated Hosts that are shared with you. The **Owner** column shows the AWS account ID of the Dedicated Host owner.

Command line

To identify a shared Dedicated Host using the AWS CLI

Use the [describe-hosts](#) command. The command returns the Dedicated Hosts that you own and Dedicated Hosts that are shared with you.

Viewing instances running on a shared Dedicated Host

Owners and consumers can view the instances running on a shared Dedicated Host at any time using one of the following methods.

Amazon EC2 console

To view the instances running on a shared Dedicated Host using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host for which to view the instances and choose **Instances**. The tab lists the instances that are running on the host. Owners see all of the instances running on the host, including instances launched by consumers. Consumers only see running instances that they launched onto the host. The **Owner** column shows the AWS account ID of the account that launched the instance.

Command line

To view the instances running on a shared Dedicated Host using the AWS CLI

Use the `describe-hosts` command. The command returns the instances running on each Dedicated Host. Owners see all of the instances running on the host. Consumers only see running instances that they launched on the shared hosts. `InstanceOwnerId` shows the AWS account ID of the instance owner.

Shared Dedicated Host permissions

Permissions for owners

Owners are responsible for managing their shared Dedicated Hosts and the instances that they launch onto them. Owners can view all instances running on the shared Dedicated Host, including those launched by consumers. However, owners can't take any action on running instances that were launched by consumers.

Permissions for consumers

Consumers are responsible for managing the instances that they launch onto a shared Dedicated Host. Consumers can't modify the shared Dedicated Host in any way, and they can't view or modify instances that were launched by other consumers or the Dedicated Host owner.

Billing and metering

There are no additional charges for sharing Dedicated Hosts.

Owners are billed for Dedicated Hosts that they share. Consumers are not billed for instances that they launch onto shared Dedicated Hosts.

Dedicated Host Reservations continue to provide billing discounts for shared Dedicated Hosts. Only Dedicated Host owners can purchase Dedicated Host Reservations for shared Dedicated Hosts that they own.

Dedicated Host limits

Shared Dedicated Hosts count towards the owner's Dedicated Hosts limits only. Consumer's Dedicated Hosts limits are not affected by Dedicated Hosts that have been shared with them. Similarly, instances that consumers launch onto shared Dedicated Hosts do not count towards their instance limits.

Host recovery and Dedicated Host sharing

Host recovery recovers instances launched by the Dedicated Host owner and the consumers with whom it has been shared. The replacement Dedicated Host is allocated to the owner's account. It is added to the same resource shares as the original Dedicated Host, and it is shared with the same consumers.

For more information, see [Host recovery \(p. 464\)](#).

Host recovery

Host recovery automatically restarts your instances onto a new replacement host if failures are detected on your Dedicated Host. Host recovery reduces the need for manual intervention and lowers the operational burden if there is an unexpected Dedicated Host failure.

Additionally, built-in integration with AWS License Manager automates the tracking and management of your licenses if a host recovery occurs.

Note

AWS License Manager integration is supported only in Regions in which AWS License Manager is available.

Contents

- [Host recovery basics \(p. 464\)](#)
- [Supported instance types \(p. 465\)](#)
- [Configuring host recovery \(p. 465\)](#)
- [Host recovery states \(p. 466\)](#)
- [Manually recovering unsupported instances \(p. 467\)](#)
- [Related services \(p. 467\)](#)
- [Pricing \(p. 467\)](#)

Host recovery basics

Host recovery uses host-level health checks to assess Dedicated Host availability and to detect underlying system failures. Examples of problems that can cause host-level health checks to fail include:

- Loss of network connectivity
- Loss of system power
- Hardware or software issues on the physical host

When a system failure is detected on your Dedicated Host, host recovery is initiated and Amazon EC2 **automatically allocates a replacement Dedicated Host**. The replacement Dedicated Host receives a new host ID, but retains the same attributes as the original Dedicated Host, including:

- Availability Zone
- Instance type
- Tags
- Auto placement settings

After the replacement Dedicated Host is allocated, the **instances are recovered on to the replacement Dedicated Host**. The recovered instances retain the same attributes as the original instances, including:

- Instance ID
- Private IP addresses
- Elastic IP addresses
- EBS volume attachments
- All instance metadata

If instances have a host affinity relationship with the impaired Dedicated Host, the recovered instances establish host affinity with the replacement Dedicated Host.

When all of the instances have been recovered on to the replacement Dedicated Host, **the impaired Dedicated Host is released**, and the replacement Dedicated Host becomes available for use.

When host recovery is initiated, the AWS account owner is notified by email and by an AWS Personal Health Dashboard event. A second notification is sent after the host recovery has been successfully completed.

Stopped instances are not recovered on to the replacement Dedicated Host. If you attempt to start a stopped instance that targets the impaired Dedicated Host, the instance start fails. We recommend that you modify the stopped instance to either target a different Dedicated Host, or to launch on any available Dedicated Host with matching configurations and auto-placement enabled.

Instances with instance storage are not recovered on to the replacement Dedicated Host. As a remedial measure, the impaired Dedicated Host is marked for retirement and you receive a retirement notification after the host recovery is complete. Follow the remedial steps described in the retirement notification within the specified time period to manually recover the remaining instances on the impaired Dedicated Host.

If you are using AWS License Manager to track your licenses, AWS License Manager allocates new licenses for the replacement Dedicated Host based on the license configuration limits. If the license configuration has hard limits that will be breached as a result of the host recovery, the recovery process is not allowed and you are notified of the host recovery failure through an Amazon SNS notification. If the license configuration has soft limits that will be breached as a result of the host recovery, the recovery is allowed to continue and you are notified of the limit breach through an Amazon SNS notification. For more information, see [Using License Configurations](#) in the *AWS License Manager User Guide*.

Supported instance types

Host recovery is supported for the following instance families: A1, C3, C4, C5, C5n, C6g, Inf1, M3, M4, M5, M5n, M6g, P3, R3, R4, R5, R5n, R6g, X1, X1e, u-6tb1, u-9tb1, u-12tb1, u-18tb1, and u-24tb1.

To recover instances that are not supported, see [Manually recovering unsupported instances \(p. 467\)](#).

Configuring host recovery

You can configure host recovery at the time of Dedicated Host allocation, or after allocation using the Amazon EC2 console or AWS Command Line Interface (CLI).

Contents

- [Enabling host recovery \(p. 465\)](#)
- [Disabling host recovery \(p. 466\)](#)
- [Viewing the host recovery configuration \(p. 466\)](#)

Enabling host recovery

You can enable host recovery at the time of Dedicated Host allocation or after allocation.

For more information about enabling host recovery at the time of Dedicated Host allocation, see [Allocating Dedicated Hosts \(p. 443\)](#).

To enable host recovery after allocation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host for which to enable host recovery, and then choose **Actions, Modify Host Recovery**.
4. For **Host recovery**, choose **Enable**, and then choose **Save**.

To enable host recovery after allocation using the AWS CLI

Use the [modify-hosts](#) command and specify the `host-recovery` parameter.

```
$ aws ec2 modify-hosts --host-recovery on --host-ids h-012a3456b7890cdef
```

Disabling host recovery

You can disable host recovery at any time after the Dedicated Host has been allocated.

To disable host recovery after allocation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host for which to disable host recovery, and then choose **Actions, Modify Host Recovery**.
4. For **Host recovery**, choose **Disable**, and then choose **Save**.

To disable host recovery after allocation using the AWS CLI

Use the [modify-hosts](#) command and specify the `host-recovery` parameter.

```
$ aws ec2 modify-hosts --host-recovery off --host-ids h-012a3456b7890cdef
```

Viewing the host recovery configuration

You can view the host recovery configuration for a Dedicated Host at any time.

To view the host recovery configuration for a Dedicated Host using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host, and in the **Description** tab, review the **Host Recovery** field.

To view the host recovery configuration for a Dedicated Host using the AWS CLI

Use the [describe-hosts](#) command.

```
$ aws ec2 describe-hosts --host-ids h-012a3456b7890cdef
```

The `HostRecovery` response element indicates whether host recovery is enabled or disabled.

Host recovery states

When a Dedicated Host failure is detected, the impaired Dedicated Host enters the `under-assessment` state, and all of the instances enter the `impaired` state. You can't launch instances on to the impaired Dedicated Host while it is in the `under-assessment` state.

After the replacement Dedicated Host is allocated, it enters the `pending` state. It remains in this state until the host recovery process is complete. You can't launch instances on to the replacement Dedicated Host while it is in the `pending` state. Recovered instances on the replacement Dedicated Host remain in the `impaired` state during the recovery process.

After the host recovery is complete, the replacement Dedicated Host enters the `available` state, and the recovered instances return to the `running` state. You can launch instances on to the replacement Dedicated Host after it enters the `available` state. The original impaired Dedicated Host is permanently released and it enters the `released-permanent-failure` state.

If the impaired Dedicated Host has instances that do not support host recovery, such as instances with instance store-backed volumes, the Dedicated Host is not released. Instead, it is marked for retirement and enters the `permanent-failure` state.

Manually recovering unsupported instances

Host recovery does not support recovering instances that use instance store volumes. Follow the instructions below to manually recover any of your instances that could not be automatically recovered.

Warning

Data on instance store volumes is lost when an instance is stopped or terminated. This includes instance store volumes that are attached to an instance that has an EBS volume as the root device. To protect data from instance store volumes, back it up to persistent storage before the instance is stopped or terminated.

Manually recovering EBS-backed instances

For EBS-backed instances that could not be automatically recovered, we recommend that you manually stop and start the instances to recover them onto a new Dedicated Host. For more information about stopping your instance, and about the changes that occur in your instance configuration when it's stopped, see [Stop and start your instance \(p. 587\)](#).

Manually recovering instance store-backed instances

For instance store-backed instances that could not be automatically recovered, we recommend that you do the following:

1. Launch a replacement instance on a new Dedicated Host from your most recent AMI.
2. Migrate all of the necessary data to the replacement instance.
3. Terminate the original instance on the impaired Dedicated Host.

Related services

Dedicated Host integrates with the following AWS services:

- **AWS License Manager**—Tracks licenses across your Amazon EC2 Dedicated Hosts (supported only in Regions in which AWS License Manager is available). For more information, see the [AWS License Manager User Guide](#).

Pricing

There are no additional charges for using host recovery, but the usual Dedicated Host charges apply. For more information, see [Amazon EC2 Dedicated Hosts Pricing](#).

As soon as host recovery is initiated, you are no longer billed for the impaired Dedicated Host. Billing for the replacement Dedicated Host begins only after it enters the `available` state.

If the impaired Dedicated Host was billed using the On-Demand rate, the replacement Dedicated Host is also billed using the On-Demand rate. If the impaired Dedicated Host had an active Dedicated Host Reservation, it is transferred to the replacement Dedicated Host.

Tracking configuration changes

You can use AWS Config to record configuration changes for Dedicated Hosts, and for instances that are launched, stopped, or terminated on them. You can then use the information captured by AWS Config as a data source for license reporting.

AWS Config records configuration information for Dedicated Hosts and instances individually, and pairs this information through relationships. There are three reporting conditions:

- **AWS Config recording status**—When **On**, AWS Config is recording one or more AWS resource types, which can include Dedicated Hosts and Dedicated Instances. To capture the information required for license reporting, verify that hosts and instances are being recorded with the following fields.
- **Host recording status**—When **Enabled**, the configuration information for Dedicated Hosts is recorded.
- **Instance recording status**—When **Enabled**, the configuration information for Dedicated Instances is recorded.

If any of these three conditions are disabled, the icon in the **Edit Config Recording** button is red. To derive the full benefit of this tool, ensure that all three recording methods are enabled. When all three are enabled, the icon is green. To edit the settings, choose **Edit Config Recording**. You are directed to the **Set up AWS Config** page in the AWS Config console, where you can set up AWS Config and start recording for your hosts, instances, and other supported resource types. For more information, see [Setting up AWS Config using the Console](#) in the *AWS Config Developer Guide*.

Note

AWS Config records your resources after it discovers them, which might take several minutes.

After AWS Config starts recording configuration changes to your hosts and instances, you can get the configuration history of any host that you have allocated or released and any instance that you have launched, stopped, or terminated. For example, at any point in the configuration history of a Dedicated Host, you can look up how many instances are launched on that host, along with the number of sockets and cores on the host. For any of those instances, you can also look up the ID of its Amazon Machine Image (AMI). You can use this information to report on licensing for your own server-bound software that is licensed per-socket or per-core.

You can view configuration histories in any of the following ways:

- By using the AWS Config console. For each recorded resource, you can view a timeline page, which provides a history of configuration details. To view this page, choose the gray icon in the **Config Timeline** column of the **Dedicated Hosts** page. For more information, see [Viewing Configuration Details in the AWS Config Console](#) in the *AWS Config Developer Guide*.
- By running AWS CLI commands. First, you can use the `list-discovered-resources` command to get a list of all hosts and instances. Then, you can use the `get-resource-config-history` command to get the configuration details of a host or instance for a specific time interval. For more information, see [View Configuration Details Using the CLI](#) in the *AWS Config Developer Guide*.
- By using the AWS Config API in your applications. First, you can use the `ListDiscoveredResources` action to get a list of all hosts and instances. Then, you can use the `GetResourceConfigHistory` action to get the configuration details of a host or instance for a specific time interval.

For example, to get a list of all of your Dedicated Hosts from AWS Config, run a CLI command such as the following.

```
aws configservice list-discovered-resources --resource-type AWS::EC2::Host
```

To obtain the configuration history of a Dedicated Host from AWS Config, run a CLI command such as the following.

```
aws configservice get-resource-config-history --resource-type AWS::EC2::Instance --  
resource-id i-1234567890abcdef0
```

To manage AWS Config settings using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the **Dedicated Hosts** page, choose **Edit Config Recording**.
3. In the AWS Config console, follow the steps provided to turn on recording. For more information, see [Setting up AWS Config using the Console](#).

For more information, see [Viewing Configuration Details in the AWS Config Console](#).

To activate AWS Config using the command line or API

- AWS CLI: [Viewing Configuration Details \(AWS CLI\)](#) in the *AWS Config Developer Guide*.
- Amazon EC2 API: [GetResourceConfigHistory](#).

Dedicated Instances

Dedicated Instances are Amazon EC2 instances that run in a virtual private cloud (VPC) on hardware that's dedicated to a single customer. Dedicated Instances that belong to different AWS accounts are physically isolated at a hardware level, even if those accounts are linked to a single payer account. However, Dedicated Instances may share hardware with other instances from the same AWS account that are not Dedicated Instances.

Note

A *Dedicated Host* is also a physical server that's dedicated for your use. With a Dedicated Host, you have visibility and control over how instances are placed on the server. For more information, see [Dedicated Hosts \(p. 438\)](#).

Dedicated Instance Basics

Each instance that you launch into a VPC has a tenancy attribute. This attribute has the following values.

Tenancy Value	Description
default	Your instance runs on shared hardware.
dedicated	Your instance runs on single-tenant hardware.
host	Your instance runs on a Dedicated Host, which is an isolated server with configurations that you can control.

After you launch an instance, there are some limitations to changing its tenancy.

- You cannot change the tenancy of an instance from `default` to `dedicated` or `host` after you've launched it.
- You cannot change the tenancy of an instance from `dedicated` or `host` to `default` after you've launched it.

You can change the tenancy of an instance from dedicated to host, or from host to dedicated after you've launched it. For more information, see [Changing the Tenancy of an Instance \(p. 473\)](#).

Each VPC has a related instance tenancy attribute. This attribute has the following values.

Tenancy Value	Description
default	An instance launched into the VPC runs on shared hardware by default, unless you explicitly specify a different tenancy during instance launch.
dedicated	An instance launched into the VPC is a Dedicated Instance by default, unless you explicitly specify a tenancy of host during instance launch. You cannot specify a tenancy of default during instance launch.

You can change the instance tenancy of a VPC from dedicated to default after you create it. You cannot change the instance tenancy of a VPC from default to dedicated after it is created.

To create Dedicated Instances, you can do the following:

- Create the VPC with the instance tenancy set to dedicated (all instances launched into this VPC are Dedicated Instances).
- Create the VPC with the instance tenancy set to default, and specify a tenancy of dedicated for any instances when you launch them.

Dedicated Instances Limitations

Some AWS services or their features won't work with a VPC with the instance tenancy set to dedicated. Check the service's documentation to confirm if there are any limitations.

Some instance types cannot be launched into a VPC with the instance tenancy set to dedicated. For more information about supported instances types, see [Amazon EC2 Dedicated Instances](#).

Amazon EBS with Dedicated Instances

When you launch an Amazon EBS-backed Dedicated Instance, the EBS volume doesn't run on single-tenant hardware.

Reserved Instances with Dedicated Tenancy

To guarantee that sufficient capacity is available to launch Dedicated Instances, you can purchase Dedicated Reserved Instances. For more information, see [Reserved Instances \(p. 308\)](#).

When you purchase a Dedicated Reserved Instance, you are purchasing the capacity to launch a Dedicated Instance into a VPC at a much reduced usage fee; the price break in the usage charge applies only if you launch an instance with dedicated tenancy. When you purchase a Reserved Instance with default tenancy, it applies only to a running instance with default tenancy; it would not apply to a running instance with dedicated tenancy.

You can't use the modification process to change the tenancy of a Reserved Instance after you've purchased it. However, you can exchange a Convertible Reserved Instance for a new Convertible Reserved Instance with a different tenancy.

Automatic Scaling of Dedicated Instances

You can use Amazon EC2 Auto Scaling to launch Dedicated Instances. For more information, see [Launching Auto Scaling Instances in a VPC](#) in the *Amazon EC2 Auto Scaling User Guide*.

Automatic Recovery of Dedicated Instances

You can configure automatic recovery for a Dedicated Instances if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. For more information, see [Recover your instance \(p. 612\)](#).

Dedicated Spot Instances

You can run a Dedicated Spot Instance by specifying a tenancy of dedicated when you create a Spot Instance request. For more information, see [Specifying a tenancy for your Spot Instances \(p. 369\)](#).

Pricing for Dedicated Instances

Pricing for Dedicated Instances is different to pricing for On-Demand Instances. For more information, see the [Amazon EC2 Dedicated Instances product page](#).

Burstable Performance Instances with Dedicated Instances

You can leverage the benefits of running on dedicated tenancy hardware with [the section called "Burstable performance instances" \(p. 219\)](#). T3 Dedicated Instances launch in unlimited mode by default, and they provide a baseline level of CPU performance with the ability to burst to a higher CPU level when required by your workload. The T3 baseline performance and ability to burst are governed by CPU credits. Because of the burstable nature of the T3 instance types, we recommend that you monitor how your T3 instances use the CPU resources of the dedicated hardware for the best performance. T3 Dedicated Instances are intended for customers with diverse workloads that display random CPU behavior, but that ideally have average CPU usage at or below the baseline usages. For more information, see [the section called "CPU credits and baseline utilization" \(p. 220\)](#).

Amazon EC2 has systems in place to identify and correct variability in performance. However, it is still possible to experience short term variability if you launch multiple T3 Dedicated Instances that have correlated CPU usage patterns. For these more demanding or correlated workloads, we recommend using M5 or M5a Dedicated Instances rather than T3 Dedicated Instances.

Working with Dedicated Instances

You can create a VPC with an instance tenancy of dedicated to ensure that all instances launched into the VPC are Dedicated Instances. Alternatively, you can specify the tenancy of the instance during launch.

Topics

- [Creating a VPC with an Instance Tenancy of Dedicated \(p. 471\)](#)
- [Launching Dedicated Instances into a VPC \(p. 472\)](#)
- [Displaying Tenancy Information \(p. 472\)](#)
- [Changing the Tenancy of an Instance \(p. 473\)](#)
- [Changing the Tenancy of a VPC \(p. 474\)](#)

Creating a VPC with an Instance Tenancy of Dedicated

When you create a VPC, you have the option of specifying its instance tenancy. If you're using the Amazon VPC console, you can create a VPC using the VPC wizard or the [Your VPCs](#) page.

To create a VPC with an instance tenancy of dedicated (VPC Wizard)

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. From the dashboard, choose **Start VPC Wizard**.
3. Select a VPC configuration, and then choose **Select**.

4. On the next page of the wizard, choose **Dedicated** from the **Hardware tenancy** list.
5. Choose **Create VPC**.

To create a VPC with an instance tenancy of dedicated (Create VPC dialog box)

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**, and then **Create VPC**.
3. For **Tenancy**, choose **Dedicated**. Specify the CIDR block, and choose **Yes, Create**.

To set the tenancy option when you create a VPC using the command line

- [create-vpc](#) (AWS CLI)
- [New-EC2Vpc](#) (AWS Tools for Windows PowerShell)

If you launch an instance into a VPC that has an instance tenancy of dedicated, your instance is automatically a Dedicated Instance, regardless of the tenancy of the instance.

Launching Dedicated Instances into a VPC

You can launch a Dedicated Instance using the Amazon EC2 launch instance wizard.

To launch a Dedicated Instance into a default tenancy VPC using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, select an AMI and choose **Select**.
4. On the **Choose an Instance Type** page, select the instance type and choose **Next: Configure Instance Details**.

Note

Ensure that you choose an instance type that's supported as a Dedicated Instance. For more information, see [Amazon EC2 Dedicated Instances](#).

5. On the **Configure Instance Details** page, select a VPC and subnet. Choose **Dedicated - Run a dedicated instance** from the **Tenancy** list, and then **Next: Add Storage**.
6. Continue as prompted by the wizard. When you've finished reviewing your options on the **Review Instance Launch** page, choose **Launch** to choose a key pair and launch the Dedicated Instance.

For more information about launching an instance with a tenancy of host, see [Launching instances onto a Dedicated Host \(p. 445\)](#).

To set the tenancy option for an instance during launch using the command line

- [run-instances](#) (AWS CLI)
- [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

Displaying Tenancy Information

To display tenancy information for your VPC using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Check the instance tenancy of your VPC in the **Tenancy** column.

4. If the **Tenancy** column is not displayed, choose **Edit Table Columns** (the gear-shaped icon), **Tenancy** in the **Show/Hide Columns** dialog box, and then **Close**.

To display tenancy information for your instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Check the tenancy of your instance in the **Tenancy** column.
4. If the **Tenancy** column is not displayed, do one of the following:
 - Choose **Show/Hide Columns** (the gear-shaped icon), **Tenancy** in the **Show/Hide Columns** dialog box, and then **Close**.
 - Select the instance. The **Description** tab in the details pane displays information about the instance, including its tenancy.

To describe the tenancy of your VPC using the command line

- [describe-vpcs](#) (AWS CLI)
- [Get-EC2Vpc](#) (AWS Tools for Windows PowerShell)

To describe the tenancy of your instance using the command line

- [describe-instances](#) (AWS CLI)
- [Get-EC2Instance](#) (AWS Tools for Windows PowerShell)

To describe the tenancy value of a Reserved Instance using the command line

- [describe-reserved-instances](#) (AWS CLI)
- [Get-EC2ReservedInstance](#) (AWS Tools for Windows PowerShell)

To describe the tenancy value of a Reserved Instance offering using the command line

- [describe-reserved-instances-offerings](#) (AWS CLI)
- [Get-EC2ReservedInstancesOffering](#) (AWS Tools for Windows PowerShell)

Changing the Tenancy of an Instance

Depending on your instance type and platform, you can change the tenancy of a stopped Dedicated Instance to host after launching it. The next time the instance starts, it's started on a Dedicated Host that's allocated to your account. For more information about allocating and working with Dedicated Hosts, and the instance types that can be used with Dedicated Hosts, see [Working with Dedicated Hosts \(p. 442\)](#). Similarly, you can change the tenancy of a stopped Dedicated Host instance to dedicated after launching it. The next time the instance starts, it's started on single-tenant hardware that we control.

To change the tenancy of an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select your instance.
3. Choose **Actions, Instance State, Stop**.
4. Choose **Actions, Instance Settings, Modify Instance Placement**.

5. In the **Tenancy** list, choose whether to run your instance on dedicated hardware or on a Dedicated Host. Choose **Save**.

To modify the tenancy value of an instance using the command line

- [modify-instance-placement](#) (AWS CLI)
- [Edit-EC2InstancePlacement](#) (AWS Tools for Windows PowerShell)

Changing the Tenancy of a VPC

You can change the instance tenancy attribute of a VPC from dedicated to default. Modifying the instance tenancy of the VPC does not affect the tenancy of any existing instances in the VPC. The next time you launch an instance in the VPC, it has a tenancy of default, unless you specify otherwise during launch.

You cannot change the tenancy attribute of a VPC from default to dedicated after it is created.

You can modify the instance tenancy attribute of a VPC using the AWS CLI, an AWS SDK, or the Amazon EC2 API only.

To modify the instance tenancy attribute of a VPC using the AWS CLI

- Use the [modify-vpc-tenancy](#) command to specify the ID of the VPC and instance tenancy value. The only supported value is default.

```
aws ec2 modify-vpc-tenancy --vpc-id vpc-1a2b3c4d --instance-tenancy default
```

On-Demand Capacity Reservations

On-Demand Capacity Reservations enable you to reserve capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. This gives you the ability to create and manage Capacity Reservations independently from the billing discounts offered by Savings Plans or regional Reserved Instances. By creating Capacity Reservations, you ensure that you always have access to EC2 capacity when you need it, for as long as you need it. You can create Capacity Reservations at any time, without entering into a one-year or three-year term commitment, and the capacity is available immediately. When you no longer need it, cancel the Capacity Reservation to stop incurring charges.

When you create a Capacity Reservation, you specify:

- The Availability Zone in which to reserve the capacity
- The number of instances for which to reserve capacity
- The instance attributes, including the instance type, tenancy, and platform/OS

Capacity Reservations can only be used by instances that match their attributes. By default, they are automatically used by running instances that match the attributes. If you don't have any running instances that match the attributes of the Capacity Reservation, it remains unused until you launch an instance with matching attributes.

In addition, you can use Savings Plans and regional Reserved Instances with your Capacity Reservations to benefit from billing discounts. AWS automatically applies your discount when the attributes of a Capacity Reservation match the attributes of a Savings Plan or regional Reserved Instance. For more information, see [Billing discounts \(p. 477\)](#).

Contents

- Differences between Capacity Reservations, Reserved Instances, and Savings Plans (p. 475)
- Supported platforms (p. 476)
- Capacity Reservation limits (p. 476)
- Capacity Reservation limitations and restrictions (p. 476)
- Capacity Reservation pricing and billing (p. 476)
- Working with Capacity Reservations (p. 478)
- Capacity Reservations in Local Zones (p. 486)
- Working with shared Capacity Reservations (p. 487)
- CloudWatch metrics for On-Demand Capacity Reservations (p. 491)

Differences between Capacity Reservations, Reserved Instances, and Savings Plans

The following table highlights key differences between Capacity Reservations, Reserved Instances, and Savings Plans:

	Capacity Reservations	Zonal Reserved Instances	Regional Reserved Instances	Savings Plans
Term	No commitment required. Can be created and canceled as needed.	Require fixed one-year or three-year commitment		
Capacity benefit	Capacity reserved in a specific Availability Zone.		Do not reserve capacity in an Availability Zone.	
Billing discount	No billing discount. Instances launched into a Capacity Reservation are charged at their standard On-Demand rates. However, you can use Savings Plans or regional Reserved Instances with Capacity Reservations to get a billing discount. Zonal Reserved Instances do not apply to Capacity Reservations.		Provide billing discounts	
Instance Limits	Limited to your On-Demand Instance limits per Region.	Limited to 20 per Availability Zone. A limit increase can be requested.	Limited to 20 per Region. A limit increase can be requested.	No limits.

For more information, see the following:

- [Reserved Instances \(p. 308\)](#)
- [Savings Plans User Guide](#)

Supported platforms

You must create the Capacity Reservation with the correct platform to ensure that it properly matches with your instances. Capacity Reservations support the following platforms:

- Linux/UNIX
- Linux with SQL Server Standard
- Linux with SQL Server Web
- Linux with SQL Server Enterprise
- Red Hat Enterprise Linux
- SUSE Linux

For more information about the supported Windows platforms, see [Supported platforms](#) in the *Amazon EC2 User Guide for Windows Instances*.

Capacity Reservation limits

The number of instances for which you are allowed to reserve capacity is based on your account's On-Demand Instance limit. You can reserve capacity for as many instances as that limit allows, minus the number of instances that are already running.

Capacity Reservation limitations and restrictions

Before you create Capacity Reservations, take note of the following limitations and restrictions.

- Active and unused Capacity Reservations count toward your On-Demand Instance limits
- Capacity Reservations are not transferable from one AWS account to another. However, you can share Capacity Reservations with other AWS accounts. For more information, see [Working with shared Capacity Reservations \(p. 487\)](#).
- Zonal Reserved Instance billing discounts do not apply to Capacity Reservations
- Capacity Reservations can't be created in placement groups
- Capacity Reservations can't be used with Dedicated Hosts

Capacity Reservation pricing and billing

The price for a Capacity Reservation varies by payment option.

Pricing

When the Capacity Reservation is active, you are charged the equivalent On-Demand rate whether you run the instances or not. If you do not use the reservation, this shows up as unused reservation on your EC2 bill. When you run an instance that matches the attributes of a reservation, you just pay for the instance and nothing for the reservation. There are no upfront or additional charges.

For example, if you create a Capacity Reservation for 20 m4.1large Linux instances and run 15 m4.1large Linux instances in the same Availability Zone, you will be charged for 15 active instances and for 5 unused instances in the reservation.

Billing discounts for Savings Plans and regional Reserved Instances apply to Capacity Reservations. For more information, see [Billing discounts \(p. 477\)](#).

For more information, see [Amazon EC2 Pricing](#).

Billing

Capacity Reservations are billed at per-second granularity. This means that you are charged for partial hours. For example, if a reservation remains active in your account for 24 hours and 15 minutes, you will be billed for 24.25 reservation hours.

The following example shows how a Capacity Reservation is billed. The Capacity Reservation is created for one `m4.1large` Linux instance, which has an On-Demand rate of \$0.10 per usage hour. In this example, the Capacity Reservation is active in the account for five hours. The Capacity Reservation is unused for the first hour, so it is billed for one unused hour at the `m4.1large` instance type's standard On-Demand rate. In hours two through five, the Capacity Reservation is occupied by an `m4.1large` instance. During this time, the Capacity Reservation accrues no charges, and the account is instead billed for the `m4.1large` instance occupying it. In the sixth hour, the Capacity Reservation is canceled and the `m4.1large` instance runs normally outside of the reserved capacity. For that hour, it is charged at the On-Demand rate of the `m4.1large` instance type.

Hour	1	2	3	
Unused Capacity Reservation	\$0.10	\$0.00	\$0.00	\$
On-demand Instance Usage	\$0.00	\$0.10	\$0.10	\$
Hourly cost	\$0.10	\$0.10	\$0.10	\$

Billing discounts

Billing discounts for Savings Plans and regional Reserved Instances apply to Capacity Reservations. AWS automatically applies these discounts to Capacity Reservations that have matching attributes. When a Capacity Reservation is used by an instance, the discount is applied to the instance. Discounts are preferentially applied to instance usage before covering unused Capacity Reservations.

Billing discounts for zonal Reserved Instances do not apply to Capacity Reservations.

For more information, see the following:

- [Reserved Instances \(p. 308\)](#)
- [Savings Plans User Guide](#)

Viewing your bill

You can review the charges and fees to your account on the AWS Billing and Cost Management console.

- The **Dashboard** displays a spend summary for your account.
- On the **Bills** page, under **Details**, expand the **Elastic Compute Cloud** section and the Region to get billing information about your Capacity Reservations.

You can view the charges online, or you can download a CSV file. For more information, see [Capacity Reservation Line Items](#) in the *AWS Billing and Cost Management User Guide*.

Working with Capacity Reservations

To start using Capacity Reservations, you create the capacity reservation in the required Availability Zone. Then, you can launch instances into the reserved capacity, view its capacity utilization in real time, and increase or decrease its capacity as needed.

By default, Capacity Reservations automatically match new instances and running instances that have matching attributes (instance type, platform, and Availability Zone). This means that any instance with matching attributes automatically runs in the Capacity Reservation. However, you can also target a Capacity Reservation for specific workloads. This enables you to explicitly control which instances are allowed to run in that reserved capacity.

You can specify how the reservation ends. You can choose to manually cancel the Capacity Reservation or end it automatically at a specified time. If you specify an end time, the Capacity Reservation is canceled within an hour of the specified time. For example, if you specify 5/31/2019, 13:30:55, the Capacity Reservation is guaranteed to end between 13:30:55 and 14:30:55 on 5/31/2019. After a reservation ends, you can no longer target instances to the Capacity Reservation. Instances running in the reserved capacity continue to run uninterrupted. If instances targeting a Capacity Reservation are stopped, you cannot restart them until you remove their Capacity Reservation targeting preference or configure them to target a different Capacity Reservation.

Contents

- [Creating a Capacity Reservation \(p. 478\)](#)
- [Working with Capacity Reservation groups \(p. 479\)](#)
- [Launching instances into an existing Capacity Reservation \(p. 483\)](#)
- [Modifying a Capacity Reservation \(p. 484\)](#)
- [Modifying an instance's Capacity Reservation settings \(p. 484\)](#)
- [Viewing a Capacity Reservation \(p. 485\)](#)
- [Canceling a Capacity Reservation \(p. 486\)](#)

Creating a Capacity Reservation

After you create the Capacity Reservation, the capacity is available immediately. The capacity remains reserved for your use as long as the Capacity Reservation is active, and you can launch instances into it at any time. If the Capacity Reservation is open, new instances and existing instances that have matching attributes automatically run in the capacity of the Capacity Reservation. If the Capacity Reservation is targeted, instances must specifically target it to run in the reserved capacity.

Your request to create a Capacity Reservation could fail if one of the following is true:

- Amazon EC2 does not have sufficient capacity to fulfill the request. Either try again at a later time, try a different Availability Zone, or try a smaller capacity. If your application is flexible across instance types and sizes, try different instance attributes.
- The requested quantity exceeds your On-Demand Instance limit for the selected instance family. Increase your On-Demand Instance limit for the instance family and try again. For more information, see [On-Demand Instance limits \(p. 306\)](#).

To create a Capacity Reservation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Capacity Reservations**, and then choose **Create Capacity Reservation**.
3. On the Create a Capacity Reservation page, configure the following settings in the **Instance details** section. The instance type, platform, and Availability Zone of the instances that you launch must match the instance type, platform, and Availability Zone that you specify here or the Capacity

Reservation is not applied. For example, if an open Capacity Reservation doesn't match, an instance launch that targets that Capacity Reservation explicitly will fail.

- a. **Instance Type**—The type of instance to launch into the reserved capacity.
 - b. **Launch EBS-optimized instances**—Specify whether to reserve the capacity for EBS-optimized instances. This option is selected by default for some instance types. For more information about EBS-optimized instances, see [Amazon Elastic Block Store \(p. 1019\)](#).
 - c. **Attach instance store at launch**—Specify whether instances launched into the Capacity Reservation use temporary block-level storage. The data on an instance store volume persists only during the life of the associated instance.
 - d. **Platform**—The operating system for your instances. For more information, see [Supported platforms \(p. 476\)](#). For more information about the supported Windows platforms, see [Supported platforms](#) in the *Amazon EC2 User Guide for Windows Instances*.
 - e. **Availability Zone**—The Availability Zone in which to reserve the capacity.
 - f. **Tenancy**—Specify whether to run on shared hardware (default) or a dedicated instance.
 - g. **Quantity**—The number of instances for which to reserve capacity. If you specify a quantity that exceeds your remaining On-Demand Instance limit for the selected instance type, the request is denied.
4. Configure the following settings in the **Reservation details** section:
 - a. **Reservation Ends**—Choose one of the following options:
 - **Manually**—Reserve the capacity until you explicitly cancel it.
 - **Specific time**—Cancel the capacity reservation automatically at the specified date and time.
 - b. **Instance eligibility**—Choose one of the following options:
 - **open**—(Default) The Capacity Reservation matches any instance that has matching attributes (instance type, platform, and Availability Zone). If you launch an instance with matching attributes, it is placed into the reserved capacity automatically.
 - **targeted**—The Capacity Reservation only accepts instances that have matching attributes (instance type, platform, and Availability Zone), and that explicitly target the reservation.
 5. Choose **Request reservation**.

To create a Capacity Reservation using the AWS CLI

Use the `create-capacity-reservation` command. For more information, see [Supported platforms \(p. 476\)](#). For more information about the supported Windows platforms, see [Supported platforms](#) in the *Amazon EC2 User Guide for Windows Instances*.

For example, the following command creates a Capacity Reservation that reserves capacity for three `m5.2xlarge` instances running Red Hat Enterprise Linux AMIs in the `us-east-1a` Availability Zone.

```
aws ec2 create-capacity-reservation --instance-type m5.2xlarge --instance-platform Red Hat Enterprise Linux --availability-zone us-east-1a --instance-count 3
```

Working with Capacity Reservation groups

You can use AWS Resource Groups to create logical collections of Capacity Reservations, called *resource groups*. A resource group is a logical grouping of AWS resources that are all in the same AWS Region. You can include multiple Capacity Reservations that have different attributes (instance type, platform, and Availability Zone) in a single resource group.

When you create resource groups for your Capacity Reservations, you can target instances to a group of Capacity Reservations instead of an individual Capacity Reservation. Instances that target a group of

Capacity Reservations match with any Capacity Reservation in the group that has matching attributes (instance type, platform, and Availability Zone) and available capacity. If the group does not have a Capacity Reservation with matching attributes and available capacity, the instances run using On-Demand capacity. If a matching Capacity Reservation is added to the targeted group at a later stage, the instance is automatically matched with and moved into its reserved capacity.

To prevent unintended use of Capacity Reservations in a group, configure the Capacity Reservations in the group to accept only instances that explicitly target the capacity reservation. To do this, set **Instance eligibility to targeted** (old console) or **Only instances that specify this reservation** (new console) when creating the Capacity Reservation using the Amazon EC2 console. When using the AWS CLI, specify `--instance-match-criteria targeted` when creating the Capacity Reservation. Doing this ensures that only instances that explicitly target the group, or a Capacity Reservation in the group, can run in the group.

If a Capacity Reservation in a group is canceled or expires while it has running instances, the instances are automatically moved to another Capacity Reservation in the group that has matching attributes and available capacity. If there are no remaining Capacity Reservations in the group that have matching attributes and available capacity, the instances run in On-Demand capacity. If a matching Capacity Reservation is added to the targeted group at a later stage, the instance is automatically moved into its reserved capacity.

To create a group for your Capacity Reservations

Use the [create-group](#) AWS CLI command. For name, provide a descriptive name for the group, and for configuration, specify two Type request parameters:

- `AWS::EC2::CapacityReservationPool` to ensure that the resource group can be targeted for instance launches
- `AWS::ResourceGroups::Generic` with `allowed-resource-types` set to `AWS::EC2::CapacityReservation` to ensure that the resource group accepts Capacity Reservations only

For example, the following command creates a group named `MyCRGroup`.

```
$ aws resource-groups create-group --name MyCRGroup --configuration
'{"Type":"AWS::EC2::CapacityReservationPool"}' '{"Type":"AWS::ResourceGroups::Generic",
"Parameters": [{"Name": "allowed-resource-types", "Values":
["AWS::EC2::CapacityReservation"]}]}'
```

The following shows example output.

```
{
    "GroupConfiguration": {
        "Status": "UPDATE_COMPLETE",
        "Configuration": [
            {
                "Type": "AWS::EC2::CapacityReservationPool"
            },
            {
                "Type": "AWS::ResourceGroups::Generic",
                "Parameters": [
                    {
                        "Values": [
                            "AWS::EC2::CapacityReservation"
                        ],
                        "Name": "allowed-resource-types"
                    }
                ]
            }
        ]
    }
}
```

```
        }
    ],
    "Group": {
        "GroupArn": "arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup",
        "Name": "MyCRGroup"
    }
}
```

To add a Capacity Reservation to a group

Use the [group-resources](#) AWS CLI command. For `group`, specify the name of the group to which to add the Capacity Reservations, and for `resources`, specify ARNs of the Capacity Reservations to add. To add multiple Capacity Reservations, separate the ARNs with a space.

For example, the following command adds two Capacity Reservations to a group named `MyCRGroup`.

```
$ aws resource-groups group-resources --group MyCRGroup --resource-arns arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1 arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

The following shows example output.

```
{
    "Failed": [],
    "Succeeded": [
        "arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1",
        "arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890"
    ]
}
```

To view the Capacity Reservations in a specific group

Use the [list-group-resources](#) AWS CLI command. For `group`, specify the name of the group.

For example, the following command lists the Capacity Reservations in a group named `MyCRGroup`.

```
$ aws resource-groups list-group-resources --group MyCRGroup
```

The following shows example output.

```
{
    "QueryErrors": [],
    "ResourceIdentifiers": [
        {
            "ResourceType": "AWS::EC2::CapacityReservation",
            "ResourceArn": "arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1"
        },
        {
            "ResourceType": "AWS::EC2::CapacityReservation",
            "ResourceArn": "arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890"
        }
    ]
}
```

To view the groups to which a specific Capacity Reservation has been added (AWS CLI)

Use the [get-groups-for-capacity-reservation](#) AWS CLI command.

For example, the following command lists the groups to which Capacity Reservation `cr-1234567890abcdef1` has been added.

```
$ aws ec2 get-groups-for-capacity-reservation --capacity-reservation-id cr-1234567890abcdef1
```

The following shows example output.

```
{  
    "CapacityReservationGroups": [  
        {  
            "OwnerId": "123456789012",  
            "GroupArn": "arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup"  
        }  
    ]  
}
```

To view the groups to which a specific Capacity Reservation has been added (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Capacity Reservations**, select the Capacity Reservation to view, and then choose **View**.

The groups to which the Capacity Reservation has been added are listed in the **Groups** card.

To remove a Capacity Reservation from a group

Use the [ungroup-resources](#) AWS CLI command. For `group`, specify the ARN of the group from which to remove the Capacity Reservation, and for `resources` specify the ARNs of the Capacity Reservations to remove. To remove multiple Capacity Reservations, separate the ARNs with a space.

The following example removes two Capacity Reservations from a group named `MyCRGroup`.

```
$ aws resource-groups ungroup-resources --group MyCRGroup --resource-arns arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-0e154d26a16094dd arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890
```

The following shows example output.

```
{  
    "Failed": [],  
    "Succeeded": [  
        "arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-0e154d26a16094dd",  
        "arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-54321abcdef567890"  
    ]  
}
```

To delete a group

Use the [delete-group](#) AWS CLI command. For `group`, provide the name of the group to delete.

For example, the following command deletes a group named `MyCRGroup`.

```
$ aws resource-groups delete-group --group MyCRGroup
```

The following shows example output.

```
{  
    "Group": {  
        "GroupArn": "arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup",  
        "Name": "MyCRGroup"  
    }  
}
```

Launching instances into an existing Capacity Reservation

When you launch an instance, you can specify whether to launch the instance into any open Capacity Reservation, into a specific Capacity Reservation, or into a group of Capacity Reservations. You can only launch an instance into a Capacity Reservation that has matching attributes (instance type, platform, and Availability Zone) and sufficient capacity. Alternatively, you can configure the instance to avoid running in a Capacity Reservation, even if you have an open Capacity Reservation that has matching attributes and available capacity.

Launching an instance into a Capacity Reservation reduces its available capacity by the number of instances launched. For example, if you launch three instances, the available capacity of the Capacity Reservation is reduced by three.

To launch instances into an existing Capacity Reservation using the console

1. Open the Launch Instance wizard by choosing **Launch Instances** from **Dashboard** or **Instances**.
2. Select an Amazon Machine Image (AMI) and an instance type.
3. Complete the **Configure Instance Details** page. For **Capacity Reservation**, choose one of the following options:
 - **None** — Prevents the instances from launching into a Capacity Reservation. The instances run in On-Demand capacity.
 - **Open** — Launches the instances into any Capacity Reservation that has matching attributes and sufficient capacity for the number of instances you selected. If there is no matching Capacity Reservation with sufficient capacity, the instance uses On-Demand capacity.
 - **Target by ID** — Launches the instances into the selected Capacity Reservation. If the selected Capacity Reservation does not have sufficient capacity for the number of instances you selected, the instance launch fails.
 - **Target by group** — Launches the instances into any Capacity Reservation with matching attributes and available capacity in the selected Capacity Reservation group. If the selected group does not have a Capacity Reservation with matching attributes and available capacity, the instances launch into On-Demand capacity.
4. Complete the remaining steps to launch the instances.

To launch an instance into an existing Capacity Reservation using the AWS CLI

Use the `run-instances` command and specify the `--capacity-reservation-specification` parameter.

The following example launches a `t2.micro` instance into any open Capacity Reservation that has matching attributes and available capacity:

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --key-name MyKeyPair --subnet-id subnet-1234567890abcdef1 --capacity-reservation-specification CapacityReservationPreference=open
```

The following example launches a `t2.micro` instance into a targeted Capacity Reservation:

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --key-name MyKeyPair --subnet-id subnet-1234567890abcdef1 --capacity-reservation-specification CapacityReservationTarget={CapacityReservationId=cr-a1234567}
```

The following example launches a `t2.micro` instance into a Capacity Reservation group:

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --key-name MyKeyPair --subnet-id subnet-1234567890abcdef1 --capacity-reservation-specification CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-groups:us-west-1:123456789012:group/my-cr-group}
```

Modifying a Capacity Reservation

You can change the attributes of an active Capacity Reservation after you have created it. You cannot modify a Capacity Reservation after it has expired or after you have explicitly canceled it.

When modifying a Capacity Reservation, you can only increase or decrease the quantity and change the way in which it is released. You cannot change the instance type, EBS optimization, instance store settings, platform, Availability Zone, or instance eligibility of a Capacity Reservation. If you need to modify any of these attributes, we recommend that you cancel the reservation, and then create a new one with the required attributes.

If you specify a new quantity that exceeds your remaining On-Demand Instance limit for the selected instance type, the update fails.

To modify a Capacity Reservation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Capacity Reservations**, select the Capacity Reservation to modify, and then choose **Edit**.
3. Modify the **Quantity** or **Reservation ends** options as needed, and choose **Save changes**.

To modify a Capacity Reservation using the AWS CLI

Use the `modify-capacity-reservations` command:

For example, the following command modifies a Capacity Reservation to reserve capacity for eight instances.

```
aws ec2 modify-capacity-reservation --capacity-reservation-id cr-1234567890abcdef0 --instance-count 8
```

Modifying an instance's Capacity Reservation settings

You can modify the following Capacity Reservation settings for a stopped instance at any time:

- Start in any Capacity Reservation that has matching attributes (instance type, platform, and Availability Zone) and available capacity.
- Start the instance in a specific Capacity Reservation.
- Start the instance in any Capacity Reservation that has matching attributes and available capacity in a Capacity Reservation group
- Prevent the instance from starting in a Capacity Reservation.

To modify an instance's Capacity Reservation settings using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Instances** and select the instance to modify. Stop the instance if it is not already stopped.
3. Choose **Actions, Modify Capacity Reservation Settings**.
4. For **Capacity Reservation**, choose one of the following options:
 - **Open** — Launches the instances into any Capacity Reservation that has matching attributes and sufficient capacity for the number of instances you selected. If there is no matching Capacity Reservation with sufficient capacity, the instance uses On-Demand capacity.
 - **None** — Prevents the instances from launching into a Capacity Reservation. The instances run in On-Demand capacity.
 - **Specify Capacity Reservation** — Launches the instances into the selected Capacity Reservation. If the selected Capacity Reservation does not have sufficient capacity for the number of instances you selected, the instance launch fails.
 - **Specify Capacity Reservation group** — Launches the instances into any Capacity Reservation with matching attributes and available capacity in the selected Capacity Reservation group. If the selected group does not have a Capacity Reservation with matching attributes and available capacity, the instances launch into On-Demand capacity.

To modify an instance's Capacity Reservation settings using the AWS CLI

Use the [modify-instance-capacity-reservation-attributes](#) command.

For example, the following command changes an instance's Capacity Reservation setting to open or none.

```
aws ec2 modify-instance-capacity-reservation-attributes --instance-id i-1234567890abcdef0  
--capacity-reservation-specification CapacityReservationPreference=none|open
```

For example, the following command modifies an instance to target a specific Capacity Reservation.

```
aws ec2 modify-instance-capacity-reservation-attributes --instance-  
id i-1234567890abcdef0 --capacity-reservation-specification  
CapacityReservationTarget={CapacityReservationId=cr-1234567890abcdef0}
```

For example, the following command modifies an instance to target a specific Capacity Reservation group.

```
aws ec2 modify-instance-capacity-reservation-attributes --instance-  
id i-1234567890abcdef0 --capacity-reservation-specification  
CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-groups:us-  
west-1:123456789012:group/my-cr-group}
```

Viewing a Capacity Reservation

Capacity Reservations have the following possible states:

- **active**—The capacity is available for use.
- **expired**—The Capacity Reservation expired automatically at the date and time specified in your reservation request. The reserved capacity is no longer available for your use.
- **cancelled**—The Capacity Reservation was manually canceled. The reserved capacity is no longer available for your use.
- **pending**—The Capacity Reservation request was successful but the capacity provisioning is still pending.

- **failed**—The Capacity Reservation request has failed. A request can fail due to invalid request parameters, capacity constraints, or instance limit constraints. You can view a failed request for 60 minutes.

To view your Capacity Reservations using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Capacity Reservations** and select a Capacity Reservation to view.
3. Choose **View launched instances for this reservation**.

To view your Capacity Reservations using the AWS CLI

Use the [describe-capacity-reservations](#) command:

For example, the following command describes all Capacity Reservations.

```
aws ec2 describe-capacity-reservations
```

Canceling a Capacity Reservation

You can cancel a Capacity Reservation at any time if you no longer need the reserved capacity. When you cancel a Capacity Reservation, the capacity is released immediately, and it is no longer reserved for your use.

You can cancel empty Capacity Reservations and Capacity Reservations that have running instances. If you cancel a Capacity Reservation that has running instances, the instances continue to run normally outside of the capacity reservation at standard On-Demand Instance rates or at a discounted rate if you have a matching Savings Plan or regional Reserved Instance.

After you cancel a Capacity Reservation, instances that target it can no longer launch. Modify these instances so that they either target a different Capacity Reservation, launch into any open Capacity Reservation with matching attributes and sufficient capacity, or avoid launching into a Capacity Reservation. For more information, see [Modifying an instance's Capacity Reservation settings \(p. 484\)](#).

To cancel a Capacity Reservation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Capacity Reservations** and select the Capacity Reservation to cancel.
3. Choose **Cancel reservation, Cancel reservation**.

To cancel a Capacity Reservation using the AWS CLI

Use the [cancel-capacity-reservation](#) command:

For example, the following command cancels a Capacity Reservation with an ID of `cr-1234567890abcdef0`.

```
aws ec2 cancel-capacity-reservation --capacity-reservation-id cr-1234567890abcdef0
```

Capacity Reservations in Local Zones

A Local Zone is an extension of an AWS Region that is geographically close to your users. Resources created in a Local Zone can serve local users with very low-latency communications. For more information, see [AWS Local Zones](#).

You can extend a VPC from its parent AWS Region into a Local Zone by creating a new subnet in that Local Zone. When you create a subnet in a Local Zone, your VPC is extended to that Local Zone. The subnet in the Local Zone operates the same as the other subnets in your VPC.

By using Local Zones, you can place Capacity Reservations in multiple locations that are closer to your users. You create and use Capacity Reservations in Local Zones in the same way that you create and use Capacity Reservations in regular Availability Zones. The same features and instance matching behavior apply. For more information about the pricing models that are supported in Local Zones, see [AWS Local Zones FAQs](#).

Limitations

- You can't share Capacity Reservations that are created in a Local Zone.
- You can't use Capacity Reservation groups in a Local Zone.

To use a Capacity Reservation in a Local Zone

1. Enable the Local Zone for use in your AWS account. For more information, see [Enable Local Zones in the Amazon EC2 User Guide for Linux Instances](#).
2. Create a Capacity Reservation in the Local Zone. For **Availability Zone**, choose the Local Zone. The Local Zone is represented by an AWS Region code followed by an identifier that indicates the location, for example us-west-2-lax-1a. For more information, see [Creating a Capacity Reservation \(p. 478\)](#).
3. Create a subnet in the Local Zone. For **Availability Zone**, choose the Local Zone. For more information, see [Creating a subnet in your VPC in the Amazon VPC User Guide](#).
4. Launch an instance. For **Subnet**, choose the subnet in the Local Zone (for example subnet-123abc | us-west-2-lax-1a), and for **Capacity Reservation**, choose the specification (either open or target it by ID) that's required for the Capacity Reservation that you created in the Local Zone. For more information, see [Launching instances into an existing Capacity Reservation \(p. 483\)](#).

Working with shared Capacity Reservations

Capacity Reservation sharing enables Capacity Reservation owners to share their reserved capacity with other AWS accounts or within an AWS organization. This enables you to create and manage Capacity Reservations centrally, and share the reserved capacity across multiple AWS accounts or within your AWS organization.

In this model, the AWS account that owns the Capacity Reservation (owner) shares it with other AWS accounts (consumers). Consumers can launch instances into Capacity Reservations that are shared with them in the same way that they launch instances into Capacity Reservations that they own in their own account. The Capacity Reservation owner is responsible for managing the Capacity Reservation and the instances that they launch into it. Owners cannot modify instances that consumers launch into Capacity Reservations that they have shared. Consumers are responsible for managing the instances that they launch into Capacity Reservations shared with them. Consumers cannot view or modify instances owned by other consumers or by the Capacity Reservation owner.

A Capacity Reservation owner can share a Capacity Reservation with:

- Specific AWS accounts inside or outside of its AWS organization
- An organizational unit inside its AWS organization
- Its entire AWS organization

Contents

- [Prerequisites for sharing Capacity Reservations \(p. 488\)](#)
- [Related services \(p. 488\)](#)
- [Sharing across Availability Zones \(p. 488\)](#)
- [Sharing a Capacity Reservation \(p. 489\)](#)
- [Stop sharing a Capacity Reservation \(p. 489\)](#)
- [Identifying a shared Capacity Reservation \(p. 490\)](#)
- [Viewing shared Capacity Reservation usage \(p. 490\)](#)
- [Shared Capacity Reservation permissions \(p. 491\)](#)
- [Billing and metering \(p. 491\)](#)
- [Instance limits \(p. 491\)](#)

Prerequisites for sharing Capacity Reservations

- To share a Capacity Reservation, you must own it in your AWS account. You cannot share a Capacity Reservation that has been shared with you.
- You can only share Capacity Reservations for shared tenancy instances. You cannot share Capacity Reservations for dedicated tenancy instances.
- Capacity Reservation sharing is not available to new AWS accounts or AWS accounts that have a limited billing history. New accounts that are linked to a qualified payer account or are linked through an AWS organization are exempt from this restriction.
- To share a Capacity Reservation with your AWS organization or an organizational unit in your AWS organization, you must enable sharing with AWS Organizations. For more information, see [Enable Sharing with AWS Organizations](#) in the *AWS RAM User Guide*.

Related services

Capacity Reservation sharing integrates with AWS Resource Access Manager (AWS RAM). AWS RAM is a service that enables you to share your AWS resources with any AWS account or through AWS Organizations. With AWS RAM, you share resources that you own by creating a *resource share*. A resource share specifies the resources to share, and the consumers with whom to share them. Consumers can be individual AWS accounts, or organizational units or an entire organization from AWS Organizations.

For more information about AWS RAM, see the [AWS RAM User Guide](#).

Sharing across Availability Zones

To ensure that resources are distributed across the Availability Zones for a Region, we independently map Availability Zones to names for each account. This could lead to Availability Zone naming differences across accounts. For example, the Availability Zone `us-east-1a` for your AWS account might not have the same location as `us-east-1a` for another AWS account.

To identify the location of your Capacity Reservations relative to your accounts, you must use the *Availability Zone ID* (AZ ID). The AZ ID is a unique and consistent identifier for an Availability Zone across all AWS accounts. For example, `use1-az1` is an AZ ID for the `us-east-1` Region and it is the same location in every AWS account.

To view the AZ IDs for the Availability Zones in your account

1. Open the AWS RAM console at <https://console.aws.amazon.com/ram>.
2. The AZ IDs for the current Region are displayed in the **Your AZ ID** panel on the right-hand side of the screen.

Sharing a Capacity Reservation

When you share a Capacity Reservation that you own with other AWS accounts, you enable them to launch instances into your reserved capacity. If you share an open Capacity Reservation, keep the following in mind as it could lead to unintended Capacity Reservation usage:

- If consumers have running instances that match the attributes of the Capacity Reservation, have the `CapacityReservationPreference` parameter set to `open`, and are not yet running in reserved capacity, they automatically use the shared Capacity Reservation.
- If consumers launch instances that have matching attributes (instance type, platform, and Availability Zone) and have the `CapacityReservationPreference` parameter set to `open`, they automatically launch into the shared Capacity Reservation.

To share a Capacity Reservation, you must add it to a resource share. A resource share is an AWS RAM resource that lets you share your resources across AWS accounts. A resource share specifies the resources to share, and the consumers with whom they are shared. When you share a Capacity Reservation using the Amazon EC2 console, you add it to an existing resource share. To add the Capacity Reservation to a new resource share, you must create the resource share using the [AWS RAM console](#).

If you are part of an organization in AWS Organizations and sharing within your organization is enabled, consumers in your organization are automatically granted access to the shared Capacity Reservation. Otherwise, consumers receive an invitation to join the resource share and are granted access to the shared Capacity Reservation after accepting the invitation.

You can share a Capacity Reservation that you own using the Amazon EC2 console, AWS RAM console, or the AWS CLI.

To share a Capacity Reservation that you own using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Capacity Reservations**.
3. Choose the Capacity Reservation to share and choose **Actions, Share reservation**.
4. Select the resource share to which to add the Capacity Reservation and choose **Share Capacity Reservation**.

It could take a few minutes for consumers to get access to the shared Capacity Reservation.

To share a Capacity Reservation that you own using the AWS RAM console

See [Creating a Resource Share](#) in the *AWS RAM User Guide*.

To share a Capacity Reservation that you own using the AWS CLI

Use the `create-resource-share` command.

Stop sharing a Capacity Reservation

The Capacity Reservation owner can stop sharing a Capacity Reservation at any time. The following rules apply:

- Instances owned by consumers that were running in the shared capacity at the time sharing stops continue to run normally outside of the reserved capacity, and the capacity is restored to the Capacity Reservation subject to Amazon EC2 capacity availability.
- Consumers with whom the Capacity Reservation was shared can no longer launch new instances into the reserved capacity.

To stop sharing a Capacity Reservation that you own, you must remove it from the resource share. You can do this using the Amazon EC2 console, AWS RAM console, or the AWS CLI.

To stop sharing a Capacity Reservation that you own using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Capacity Reservations**.
3. Select the Capacity Reservation and choose the **Sharing** tab.
4. The **Sharing** tab lists the resource shares to which the Capacity Reservation has been added. Select the resource share from which to remove the Capacity Reservation and choose **Remove from resource share**.

To stop sharing a Capacity Reservation that you own using the AWS RAM console

See [Updating a Resource Share](#) in the *AWS RAM User Guide*.

To stop sharing a Capacity Reservation that you own using the AWS CLI

Use the [disassociate-resource-share](#) command.

Identifying a shared Capacity Reservation

Owners and consumers can identify shared Capacity Reservations using the Amazon EC2 console and AWS CLI

To identify a shared Capacity Reservation using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Capacity Reservations**. The screen lists Capacity Reservations that you own and Capacity Reservations that are shared with you. The **Owner** column shows the AWS account ID of the Capacity Reservation owner. (me) next to the AWS account ID indicates that you are the owner.

To identify a shared Capacity Reservation using the AWS CLI

Use the [describe-capacity-reservations](#) command. The command returns the Capacity Reservations that you own and Capacity Reservations that are shared with you. **OwnerId** shows the AWS account ID of the Capacity Reservation owner.

Viewing shared Capacity Reservation usage

The owner of a shared Capacity Reservation can view its usage at any time using the Amazon EC2 console and the AWS CLI.

To view Capacity Reservation usage using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Capacity Reservations**.
3. Select the Capacity Reservation for which to view the usage and choose the **Usage** tab.

The **AWS account ID** column shows the account IDs of the consumers currently using the Capacity Reservation. The **Launched instances** column shows the number of instances each consumer currently has running in the reserved capacity.

To view Capacity Reservation usage using the AWS CLI

Use the [get-capacity-reservation-usage](#) command. AccountId shows the account ID of the account using the Capacity Reservation. UsedInstanceCount shows the number of instances the consumer currently has running in the reserved capacity.

Shared Capacity Reservation permissions

Permissions for owners

Owners are responsible for managing and canceling their shared Capacity Reservations. Owners cannot modify instances running in the shared Capacity Reservation that are owned by other accounts. Owners remain responsible for managing instances that they launch into the shared Capacity Reservation.

Permissions for consumers

Consumers are responsible for managing their instances that are running the shared Capacity Reservation. Consumers cannot modify the shared Capacity Reservation in any way, and they cannot view or modify instances that are owned by other consumers or the Capacity Reservation owner.

Billing and metering

There are no additional charges for sharing Capacity Reservations.

The Capacity Reservation owner is billed for instances that they run inside the Capacity Reservation and for unused reserved capacity. Consumers are billed for the instances that they run inside the shared Capacity Reservation.

Instance limits

All Capacity Reservation usage counts toward the Capacity Reservation owner's On-Demand Instance limits. This includes:

- Unused reserved capacity
- Usage by instances owned by the Capacity Reservation owner
- Usage by instances owned by consumers

Instances launched into the shared capacity by consumers count towards the Capacity Reservation owner's On-Demand Instance limit. Consumers' instance limits are a sum of their own On-Demand Instance limits and the capacity available in the shared Capacity Reservations to which they have access.

CloudWatch metrics for On-Demand Capacity Reservations

With CloudWatch metrics, you can efficiently monitor your Capacity Reservations and identify unused capacity by setting CloudWatch alarms to notify you when usage thresholds are met. This can help you maintain a constant Capacity Reservation volume and achieve a higher level of utilization.

On-Demand Capacity Reservations send metric data to CloudWatch every five minutes. Metrics are not supported for Capacity Reservations that are active for less than five minutes.

For more information about viewing metrics in the CloudWatch console, see [Using Amazon CloudWatch Metrics](#). For more information about creating alarms, see [Creating Amazon CloudWatch Alarms](#).

Contents

- [Capacity Reservation usage metrics \(p. 492\)](#)
- [Capacity Reservation metric dimensions \(p. 492\)](#)
- [Viewing CloudWatch metrics for Capacity Reservations \(p. 492\)](#)

Capacity Reservation usage metrics

The AWS/EC2CapacityReservations namespace includes the following usage metrics you can use to monitor and maintain on-demand capacity within thresholds you specify for your reservation.

Metric	Description
UsedInstanceCount	The number of instances that are currently in use. Unit: Count
AvailableInstanceCount	The number of instances that are available. Unit: Count
TotalInstanceCount	The total number of instances you have reserved. Unit: Count
InstanceUtilization	The percentage of reserved capacity instances that are currently in use. Unit: Percent

Capacity Reservation metric dimensions

You can use the following dimensions to refine the metrics listed in the previous table.

Dimension	Description
CapacityReservationId	This globally unique dimension filters the data you request for the identified capacity reservation only.

Viewing CloudWatch metrics for Capacity Reservations

Metrics are grouped first by the service namespace, and then by the supported dimensions. You can use the following procedures to view the metrics for your Capacity Reservations.

To view Capacity Reservation metrics using the CloudWatch console

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, change the Region. From the navigation bar, select the Region where your Capacity Reservation resides. For more information, see [Regions and Endpoints](#).
3. In the navigation pane, choose **Metrics**.
4. For **All metrics**, choose **EC2 Capacity Reservations**.
5. Choose the metric dimension **By Capacity Reservation**. Metrics will be grouped by CapacityReservationId.
6. To sort the metrics, use the column heading. To graph a metric, select the check box next to the metric.

To view Capacity Reservation metrics (AWS CLI)

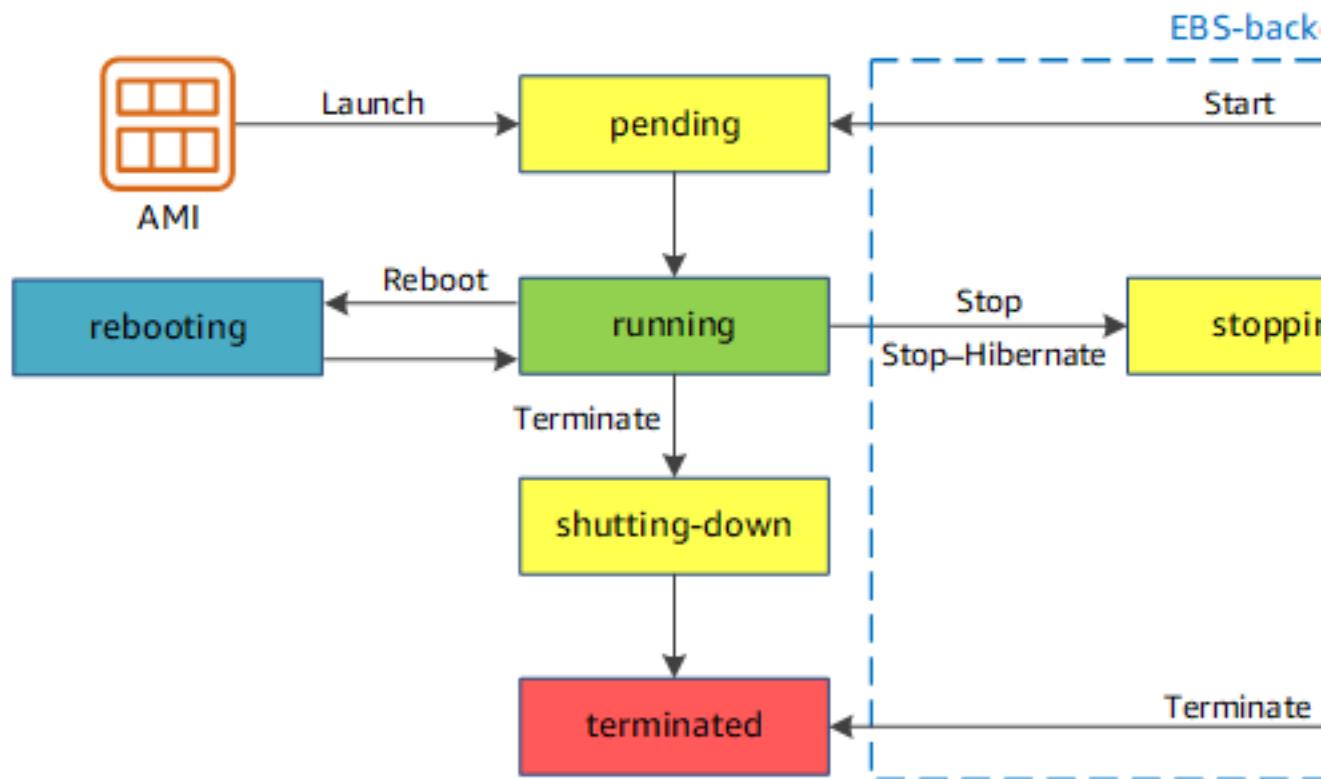
Use the following [list-metrics](#) command:

```
aws cloudwatch list-metrics --namespace "AWS/EC2CapacityReservations"
```

Instance lifecycle

An Amazon EC2 instance transitions through different states from the moment you launch it through to its termination.

The following illustration represents the transitions between instance states. Notice that you can't stop and start an instance store-backed instance. For more information about instance store-backed instances, see [Storage for the root device \(p. 100\)](#).



The following table provides a brief description of each instance state and indicates whether it is billed or not.

Note

The table indicates billing for instance usage only. Some AWS resources, such as Amazon EBS volumes and Elastic IP addresses, incur charges regardless of the instance's state. For more information, see [Avoiding Unexpected Charges](#) in the *AWS Billing and Cost Management User Guide*.

Instance state	Description	Instance usage billing
pending	The instance is preparing to enter the running state. An instance enters the pending state when it launches for	Not billed

Instance state	Description	Instance usage billing
	the first time, or when it is started after being in the stopped state.	
running	The instance is running and ready for use.	Billed
stopping	The instance is preparing to be stopped or stop-hibernated.	Not billed if preparing to stop Billed if preparing to hibernate
stopped	The instance is shut down and cannot be used. The instance can be started at any time.	Not billed
shutting down	The instance is preparing to be terminated.	Not billed
terminated	The instance has been permanently deleted and cannot be started.	<p>Not billed</p> <p>Note Reserved Instances that applied to terminated instances are billed until the end of their term according to their payment option. For more information, see Reserved Instances (p. 308)</p>

Note

Rebooting an instance doesn't start a new instance billing period because the instance stays in the `running` state.

Instance launch

When you launch an instance, it enters the `pending` state. The instance type that you specified at launch determines the hardware of the host computer for your instance. We use the Amazon Machine Image (AMI) you specified at launch to boot the instance. After the instance is ready for you, it enters the `running` state. You can connect to your running instance and use it the way that you'd use a computer sitting in front of you.

As soon as your instance transitions to the `running` state, you're billed for each second, with a one-minute minimum, that you keep the instance running, even if the instance remains idle and you don't connect to it.

For more information, see [Launch your instance \(p. 497\)](#) and [Connect to your Linux instance \(p. 563\)](#).

Instance stop and start (Amazon EBS-backed instances only)

If your instance fails a status check or is not running your applications as expected, and if the root volume of your instance is an Amazon EBS volume, you can stop and start your instance to try to fix the problem.

When you stop your instance, it enters the `stopping` state, and then the `stopped` state. We don't charge usage or data transfer fees for your instance after you stop it, but we do charge for the storage

for any Amazon EBS volumes. While your instance is in the `stopped` state, you can modify certain attributes of the instance, including the instance type.

When you start your instance, it enters the `pending` state, and we move the instance to a new host computer (though in some cases, it remains on the current host). When you stop and start your instance, you lose any data on the instance store volumes on the previous host computer.

Your instance retains its private IPv4 address, which means that an Elastic IP address associated with the private IPv4 address or network interface is still associated with your instance. If your instance has an IPv6 address, it retains its IPv6 address.

Each time you transition an instance from `stopped` to `running`, we charge per second when the instance is running, with a minimum of one minute every time you start your instance.

For more information, see [Stop and start your instance \(p. 587\)](#).

Instance hibernate (Amazon EBS-backed instances only)

When you hibernate an instance, we signal the operating system to perform hibernation (suspend-to-disk), which saves the contents from the instance memory (RAM) to your Amazon EBS root volume. We persist the instance's Amazon EBS root volume and any attached Amazon EBS data volumes. When you start your instance, the Amazon EBS root volume is restored to its previous state and the RAM contents are reloaded. Previously attached data volumes are reattached and the instance retains its instance ID.

When you hibernate your instance, it enters the `stopping` state, and then the `stopped` state. We don't charge usage for a hibernated instance when it is in the `stopped` state, but we do charge while it is in the `stopping` state, unlike when you [stop an instance \(p. 494\)](#) without hibernating it. We don't charge usage for data transfer fees, but we do charge for the storage for any Amazon EBS volumes, including storage for the RAM data.

When you start your hibernated instance, it enters the `pending` state, and we move the instance to a new host computer (though in some cases, it remains on the current host).

Your instance retains its private IPv4 address, which means that an Elastic IP address associated with the private IPv4 address or network interface is still associated with your instance. If your instance has an IPv6 address, it retains its IPv6 address.

For more information, see [Hibernate your Linux instance \(p. 591\)](#).

Instance reboot

You can reboot your instance using the Amazon EC2 console, a command line tool, and the Amazon EC2 API. We recommend that you use Amazon EC2 to reboot your instance instead of running the operating system reboot command from your instance.

Rebooting an instance is equivalent to rebooting an operating system. The instance remains on the same host computer and maintains its public DNS name, private IP address, and any data on its instance store volumes. It typically takes a few minutes for the reboot to complete, but the time it takes to reboot depends on the instance configuration.

Rebooting an instance doesn't start a new instance billing period; per second billing continues without a further one-minute minimum charge.

For more information, see [Reboot your instance \(p. 603\)](#).

Instance retirement

An instance is scheduled to be retired when AWS detects the irreparable failure of the underlying hardware hosting the instance. When an instance reaches its scheduled retirement date, it is stopped or terminated by AWS. If your instance root device is an Amazon EBS volume, the instance is stopped, and you can start it again at any time. If your instance root device is an instance store volume, the instance is terminated, and cannot be used again.

For more information, see [Instance retirement \(p. 604\)](#).

Instance termination

When you've decided that you no longer need an instance, you can terminate it. As soon as the status of an instance changes to **shutting-down** or **terminated**, you stop incurring charges for that instance.

If you enable termination protection, you can't terminate the instance using the console, CLI, or API.

After you terminate an instance, it remains visible in the console for a short while, and then the entry is automatically deleted. You can also describe a terminated instance using the CLI and API. Resources (such as tags) are gradually disassociated from the terminated instance, therefore may no longer be visible on the terminated instance after a short while. You can't connect to or recover a terminated instance.

Each Amazon EBS-backed instance supports the `InstanceInitiatedShutdownBehavior` attribute, which controls whether the instance stops or terminates when you initiate shutdown from within the instance itself (for example, by using the `shutdown` command on Linux). The default behavior is to stop the instance. You can modify the setting of this attribute while the instance is running or stopped.

Each Amazon EBS volume supports the `DeleteOnTermination` attribute, which controls whether the volume is deleted or preserved when you terminate the instance it is attached to. The default is to delete the root device volume and preserve any other EBS volumes.

For more information, see [Terminate your instance \(p. 607\)](#).

Differences between reboot, stop, hibernate, and terminate

The following table summarizes the key differences between rebooting, stopping, hibernating, and terminating your instance.

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Hibernate (Amazon EBS-backed instances only)	Terminate
Host computer	The instance stays on the same host computer	We move the instance to a new host computer (though in some cases, it remains on the current host).	We move the instance to a new host computer (though in some cases, it remains on the current host).	None
Private and public IPv4 addresses	These addresses stay the same	The instance keeps its private IPv4 address. The instance gets	The instance keeps its private IPv4 address. The instance gets	None

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Hibernate (Amazon EBS-backed instances only)	Terminate
		a new public IPv4 address, unless it has an Elastic IP address, which doesn't change during a stop/start.	a new public IPv4 address, unless it has an Elastic IP address, which doesn't change during a stop/start.	
Elastic IP addresses (IPv4)	The Elastic IP address remains associated with the instance	The Elastic IP address remains associated with the instance	The Elastic IP address remains associated with the instance	The Elastic IP address is disassociated from the instance
IPv6 address	The address stays the same	The instance keeps its IPv6 address	The instance keeps its IPv6 address	None
Instance store volumes	The data is preserved	The data is erased	The data is erased	The data is erased
Root device volume	The volume is preserved	The volume is preserved	The volume is preserved	The volume is deleted by default
RAM (contents of memory)	The RAM is erased	The RAM is erased	The RAM is saved to a file on the root volume	The RAM is erased
Billing	The instance billing hour doesn't change.	You stop incurring charges for an instance as soon as its state changes to stopping. Each time an instance transitions from stopped to running, we start a new instance billing period, billing a minimum of one minute every time you start your instance.	You incur charges while the instance is in the stopping state, but stop incurring charges when the instance is in the stopped state. Each time an instance transitions from stopped to running, we start a new instance billing period, billing a minimum of one minute every time you start your instance.	You stop incurring charges for an instance as soon as its state changes to shutting-down.

Operating system shutdown commands always terminate an instance store-backed instance. You can control whether operating system shutdown commands stop or terminate an Amazon EBS-backed instance. For more information, see [Changing the instance initiated shutdown behavior \(p. 610\)](#).

Launch your instance

An instance is a virtual server in the AWS Cloud. You launch an instance from an Amazon Machine Image (AMI). The AMI provides the operating system, application server, and applications for your instance.

When you sign up for AWS, you can get started with Amazon EC2 for free using the [AWS Free Tier](#). You can use the free tier to launch and use a `t2.micro` instance for free for 12 months (in Regions where

`t2.micro` is unavailable, you can use a `t3.micro` instance under the free tier). If you launch an instance that is not within the free tier, you incur the standard Amazon EC2 usage fees for the instance. For more information, see the [Amazon EC2 Pricing](#).

You can launch an instance using the following methods.

Method	Documentation
[Amazon EC2 console] Use the launch instance wizard to specify the launch parameters.	Launching an instance using the Launch Instance Wizard (p. 499)
[Amazon EC2 console] Create a launch template and launch the instance from the launch template.	Launching an instance from a launch template (p. 504)
[Amazon EC2 console] Use an existing instance as the base.	Launching an instance using parameters from an existing instance (p. 521)
[Amazon EC2 console] Use an AMI that you purchased from the AWS Marketplace.	Launching an AWS Marketplace instance (p. 522)
[AWS CLI] Use an AMI that you select.	Using Amazon EC2 through the AWS CLI
[AWS Tools for Windows PowerShell] Use an AMI that you select.	Amazon EC2 from the AWS Tools for Windows PowerShell
[AWS CLI] Use EC2 Fleet to provision capacity across different EC2 instance types and Availability Zones, and across On-Demand Instance, Reserved Instance, and Spot Instance purchase models.	Launching instances using an EC2 Fleet (p. 524)

When you launch your instance, you can launch your instance in a subnet that is associated with one of the following resources:

- An Availability Zone - This option is the default.
- A Local Zone - To launch an instance in a Local Zone, you must opt in to the Local Zone, and then create a subnet in the zone. For more information, see [Local Zones](#)
- A Wavelength Zone - To launch an instance in a Wavelength Zone, you must opt in to the Wavelength Zone, and then create a subnet in the zone. For information about how to launch an instance in a Wavelength Zone, see [Get started with AWS Wavelength](#) in the *AWS Wavelength Developer Guide*.
- An Outpost - To launch an instance in an Outpost, you must create an Outpost. For information about how to create an Outpost, see [Get Started with AWS Outposts](#) in the *AWS Outposts User Guide*.

After you launch your instance, you can connect to it and use it. To begin, the instance state is `pending`. When the instance state is `running`, the instance has started booting. There might be a short time before you can connect to the instance. Note that bare metal instance types might take longer to launch. For more information about bare metal instances, see [Instances built on the Nitro System \(p. 205\)](#).

The instance receives a public DNS name that you can use to contact the instance from the internet. The instance also receives a private DNS name that other instances within the same VPC can use to contact the instance. For more information about connecting to your instance, see [Connect to your Linux instance \(p. 563\)](#).

When you are finished with an instance, be sure to terminate it. For more information, see [Terminate your instance \(p. 607\)](#).

Launching an instance using the Launch Instance Wizard

You can launch an instance using the launch instance wizard. The launch instance wizard specifies all the launch parameters required for launching an instance. Where the launch instance wizard provides a default value, you can accept the default or specify your own value. At the very least, you need to select an AMI and a key pair to launch an instance.

Before you launch your instance, be sure that you are set up. For more information, see [Setting up with Amazon EC2 \(p. 26\)](#).

Important

When you launch an instance that's not within the [AWS Free Tier](#), you are charged for the time that the instance is running, even if it remains idle.

Steps to launch an instance:

- [Initiate instance launch \(p. 499\)](#)
- [Step 1: Choose an Amazon Machine Image \(AMI\) \(p. 499\)](#)
- [Step 2: Choose an Instance Type \(p. 500\)](#)
- [Step 3: Configure Instance Details \(p. 500\)](#)
- [Step 4: Add Storage \(p. 503\)](#)
- [Step 5: Add Tags \(p. 503\)](#)
- [Step 6: Configure Security Group \(p. 503\)](#)
- [Step 7: Review Instance Launch and Select Key Pair \(p. 504\)](#)

Initiate instance launch

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation bar at the top of the screen, the current Region is displayed (for example, US East (Ohio)). Select a Region for the instance that meets your needs. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can't. For more information, see [Resource locations \(p. 1222\)](#).
3. From the Amazon EC2 console dashboard, choose **Launch instance**.

Step 1: Choose an Amazon Machine Image (AMI)

When you launch an instance, you must select a configuration, known as an Amazon Machine Image (AMI). An AMI contains the information required to create a new instance. For example, an AMI might contain the software required to act as a web server, such as Linux, Apache, and your website.

When you launch an instance, you can either select an AMI from the list, or you can select a Systems Manager parameter that points to an AMI ID. For more information, see [Using a Systems Manager parameter to find an AMI](#).

On the **Choose an Amazon Machine Image (AMI)** page, use one of two options to choose an AMI. Either [search the list of AMIs \(p. 499\)](#), or [search by Systems Manager parameter \(p. 500\)](#).

By searching the list of AMIs

1. Select the type of AMI to use in the left pane:

Quick Start

A selection of popular AMIs to help you get started quickly. To select an AMI that is eligible for the free tier, choose **Free tier only** in the left pane. These AMIs are marked **Free tier eligible**.

My AMIs

The private AMIs that you own, or private AMIs that have been shared with you. To view AMIs that are shared with you, choose **Shared with me** in the left pane.

AWS Marketplace

An online store where you can buy software that runs on AWS, including AMIs. For more information about launching an instance from the AWS Marketplace, see [Launching an AWS Marketplace instance \(p. 522\)](#).

Community AMIs

The AMIs that AWS community members have made available for others to use. To filter the list of AMIs by operating system, choose the appropriate check box under **Operating system**. You can also filter by architecture and root device type.

2. Check the **Root device type** listed for each AMI. Notice which AMIs are the type that you need, either `ebs` (backed by Amazon EBS) or `instance-store` (backed by instance store). For more information, see [Storage for the root device \(p. 100\)](#).
3. Check the **Virtualization type** listed for each AMI. Notice which AMIs are the type that you need, either `hvm` or `paravirtual`. For example, some instance types require HVM. For more information, see [Linux AMI virtualization types \(p. 102\)](#).
4. Choose an AMI that meets your needs, and then choose **Select**.

By Systems Manager parameter

1. Choose **Search by Systems Manager parameter** (at top right).
2. For **Systems Manager parameter**, select a parameter. The corresponding AMI ID appears next to **Currently resolves to**.
3. Choose **Search**. The AMIs that match the AMI ID appear in the list.
4. Select the AMI from the list, and choose **Select**.

Step 2: Choose an Instance Type

On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. Larger instance types have more CPU and memory. For more information, see [Instance types \(p. 200\)](#).

To remain eligible for the free tier, choose the `t2.micro` instance type (or the `t3.micro` instance type in Regions where `t2.micro` is unavailable). For more information, see [Burstable performance instances \(p. 219\)](#).

By default, the wizard displays current generation instance types, and selects the first available instance type based on the AMI that you selected. To view previous generation instance types, choose **All generations** from the filter list.

Note

To set up an instance quickly for testing purposes, choose **Review and Launch** to accept the default configuration settings, and launch your instance. Otherwise, to configure your instance further, choose **Next: Configure Instance Details**.

Step 3: Configure Instance Details

On the **Configure Instance Details** page, change the following settings as necessary (expand **Advanced Details** to see all the settings), and then choose **Next: Add Storage**:

- **Number of instances:** Enter the number of instances to launch.

Tip

To ensure faster instance launches, break up large requests into smaller batches. For example, create five separate launch requests for 100 instances each instead of one launch request for 500 instances.

- (Optional) To help ensure that you maintain the correct number of instances to handle demand on your application, you can choose **Launch into Auto Scaling Group** to create a launch configuration and an Auto Scaling group. Auto Scaling scales the number of instances in the group according to your specifications. For more information, see the [Amazon EC2 Auto Scaling User Guide](#).

Note

If Amazon EC2 Auto Scaling marks an instance that is in an Auto Scaling group as unhealthy, the instance is automatically scheduled for replacement where it is terminated and another is launched, and you lose your data on the original instance. An instance is marked as unhealthy if you stop or reboot the instance, or if another event marks the instance as unhealthy. For more information, see [Health Checks for Auto Scaling Instances](#) in the *Amazon EC2 Auto Scaling User Guide*.

- **Purchasing option:** Choose **Request Spot instances** to launch a Spot Instance. This adds and removes options from this page. Set your maximum price, and optionally update the request type, interruption behavior, and request validity. For more information, see [Creating a Spot Instance request \(p. 371\)](#).
- **Network:** Select the VPC, or to create a new VPC, choose **Create new VPC** to go to the Amazon VPC console. When you have finished, return to the wizard and choose **Refresh** to load your VPC in the list.
- **Subnet:** You can launch an instance in a subnet associated with an Availability Zone, Local Zone, Wavelength Zone or Outpost.

To launch the instance in an Availability Zone, select the subnet into which to launch your instance. You can select **No preference** to let AWS choose a default subnet in any Availability Zone. To create a new subnet, choose **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and choose **Refresh** to load your subnet in the list.

To launch the instance in a Local Zone, select a subnet that you created in the Local Zone.

To launch an instance in an Outpost, select a subnet in a VPC that you associated with an Outpost.

- **Auto-assign Public IP:** Specify whether your instance receives a public IPv4 address. By default, instances in a default subnet receive a public IPv4 address and instances in a nondefault subnet do not. You can select **Enable** or **Disable** to override the subnet's default setting. For more information, see [Public IPv4 addresses and external DNS hostnames \(p. 760\)](#).
- **Auto-assign IPv6 IP:** Specify whether your instance receives an IPv6 address from the range of the subnet. Select **Enable** or **Disable** to override the subnet's default setting. This option is only available if you've associated an IPv6 CIDR block with your VPC and subnet. For more information, see [Your VPC and Subnets](#) in the *Amazon VPC User Guide*.
- **Domain join directory:** Select the AWS Directory Service directory (domain) to which your Linux instance is joined after launch. If you select a domain, you must select an IAM role with the required permissions. For more information, see [Seamlessly Join a Linux EC2 Instance to Your AWS Managed Microsoft AD Directory](#).
- **Placement group:** A placement group determines the placement strategy of your instances. Select an existing placement group, or create a new one. This option is only available if you've selected an instance type that supports placement groups. For more information, see [Placement groups \(p. 869\)](#).
- **Capacity Reservation:** Specify whether to launch the instance into shared capacity, any open Capacity Reservation, a specific Capacity Reservation, or a Capacity Reservation group. For more information, see [Launching instances into an existing Capacity Reservation \(p. 483\)](#).
- **IAM role:** Select an AWS Identity and Access Management (IAM) role to associate with the instance. For more information, see [IAM roles for Amazon EC2 \(p. 974\)](#).
- **CPU options:** Choose **Specify CPU options** to specify a custom number of vCPUs during launch. Set the number of CPU cores and threads per core. For more information, see [Optimizing CPU options \(p. 632\)](#).

- **Shutdown behavior:** Select whether the instance should stop or terminate when shut down. For more information, see [Changing the instance initiated shutdown behavior \(p. 610\)](#).
- **Stop - Hibernate behavior:** To enable hibernation, select this check box. This option is only available if your instance meets the hibernation prerequisites. For more information, see [Hibernate your Linux instance \(p. 591\)](#).
- **Enable termination protection:** To prevent accidental termination, select this check box. For more information, see [Enabling termination protection \(p. 609\)](#).
- **Monitoring:** Select this check box to enable detailed monitoring of your instance using Amazon CloudWatch. Additional charges apply. For more information, see [Monitoring your instances using CloudWatch \(p. 715\)](#).
- **EBS-optimized instance:** An Amazon EBS-optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. If the instance type supports this feature, select this check box to enable it. Additional charges apply. For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#).
- **Tenancy:** If you are launching your instance into a VPC, you can choose to run your instance on isolated, dedicated hardware (**Dedicated**) or on a Dedicated Host (**Dedicated host**). Additional charges may apply. For more information, see [Dedicated Instances \(p. 469\)](#) and [Dedicated Hosts \(p. 438\)](#).
- **T2/T3 Unlimited:** Select this check box to enable applications to burst beyond the baseline for as long as needed. Additional charges may apply. For more information, see [Burstable performance instances \(p. 219\)](#).
- **File systems:** Choose **Add file system** to mount one or more Amazon EFS file systems to your instance. For more information, see [Using Amazon EFS with Amazon EC2 \(p. 1206\)](#).
- **Network interfaces:** If you selected a specific subnet, you can specify up to two network interfaces for your instance:
 - For **Network Interface**, select **New network interface** to let AWS create a new interface, or select an existing, available network interface.
 - For **Primary IP**, enter a private IPv4 address from the range of your subnet, or leave **Auto-assign** to let AWS choose a private IPv4 address for you.
 - For **Secondary IP addresses**, choose **Add IP** to assign more than one private IPv4 address to the selected network interface.
 - (IPv6-only) For **IPv6 IPs**, choose **Add IP**, and enter an IPv6 address from the range of the subnet, or leave **Auto-assign** to let AWS choose one for you.
 - Choose **Add Device** to add a secondary network interface. A secondary network interface can reside in a different subnet of the VPC, provided it's in the same Availability Zone as your instance.

For more information, see [Elastic network interfaces \(p. 789\)](#). If you specify more than one network interface, your instance cannot receive a public IPv4 address. Additionally, if you specify an existing network interface for eth0, you cannot override the subnet's public IPv4 setting using **Auto-assign Public IP**. For more information, see [Assigning a public IPv4 address during instance launch \(p. 764\)](#).

- **Kernel ID:** (Only valid for paravirtual (PV) AMIs) Select **Use default** unless you want to use a specific kernel.
- **RAM disk ID:** (Only valid for paravirtual (PV) AMIs) Select **Use default** unless you want to use a specific RAM disk. If you have selected a kernel, you may need to select a specific RAM disk with the drivers to support it.
- **Metadata accessible:** You can enable or disable access to the instance metadata. For more information, see [Configuring the instance metadata service \(p. 659\)](#).
- **Metadata version:** If you enable access to the instance metadata, you can choose to require the use of Instance Metadata Service Version 2 when requesting instance metadata. For more information, see [Configuring instance metadata options for new instances \(p. 663\)](#).
- **Metadata token response hop limit:** If you enable instance metadata, you can set the allowable number of network hops for the metadata token. For more information, see [Configuring the instance metadata service \(p. 659\)](#).

- **User data:** You can specify user data to configure an instance during launch, or to run a configuration script. To attach a file, select the **As file** option and browse for the file to attach.

Step 4: Add Storage

The AMI you selected includes one or more volumes of storage, including the root device volume. On the **Add Storage** page, you can specify additional volumes to attach to the instance by choosing **Add New Volume**. Configure each volume as follows, and then choose **Next: Add Tags**.

- **Type:** Select instance store or Amazon EBS volumes to associate with your instance. The types of volume available in the list depend on the instance type you've chosen. For more information, see [Amazon EC2 instance store \(p. 1188\)](#) and [Amazon EBS volumes \(p. 1021\)](#).
- **Device:** Select from the list of available device names for the volume.
- **Snapshot:** Enter the name or ID of the snapshot from which to restore a volume. You can also search for available shared and public snapshots by typing text into the **Snapshot** field. Snapshot descriptions are case-sensitive.
- **Size:** For EBS volumes, you can specify a storage size. Even if you have selected an AMI and instance that are eligible for the free tier, to stay within the free tier, you must stay under 30 GiB of total storage. For more information, see [Constraints on the size and configuration of an EBS volume \(p. 1037\)](#).
- **Volume Type:** For EBS volumes, select a volume type. For more information, see [Amazon EBS volume types \(p. 1023\)](#).
- **IOPS:** If you have selected a Provisioned IOPS SSD volume type, then you can enter the number of I/O operations per second (IOPS) that the volume can support.
- **Delete on Termination:** For Amazon EBS volumes, select this check box to delete the volume when the instance is terminated. For more information, see [Preserving Amazon EBS volumes on instance termination \(p. 610\)](#).
- **Encrypted:** If the instance type supports EBS encryption, you can specify the encryption state of the volume. If you have enabled encryption by default in this Region, the default CMK is selected for you. You can select a different key or disable encryption. For more information, see [Amazon EBS encryption \(p. 1121\)](#).

Step 5: Add Tags

On the **Add Tags** page, specify [tags \(p. 1229\)](#) by providing key and value combinations. You can tag the instance, the volumes, or both. For Spot Instances, you can tag the Spot Instance request only. Choose **Add another tag** to add more than one tag to your resources. Choose **Next: Configure Security Group** when you are done.

Step 6: Configure Security Group

On the **Configure Security Group** page, use a security group to define firewall rules for your instance. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored. (For more information about security groups, see [Amazon EC2 security groups for Linux instances \(p. 999\)](#).) Select or create a security group as follows, and then choose **Review and Launch**.

- To select an existing security group, choose **Select an existing security group**, and select your security group. You can't edit the rules of an existing security group, but you can copy them to a new group by choosing **Copy to new**. Then you can add rules as described in the next step.
- To create a new security group, choose **Create a new security group**. The wizard automatically defines the launch-wizard-x security group and creates an inbound rule to allow you to connect to your instance over SSH (port 22).
- You can add rules to suit your needs. For example, if your instance is a web server, open ports 80 (HTTP) and 443 (HTTPS) to allow internet traffic.

To add a rule, choose **Add Rule**, select the protocol to open to network traffic, and then specify the source. Choose **My IP** from the **Source** list to let the wizard add your computer's public IP address. However, if you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

Warning

Rules that enable all IP addresses (0.0.0.0/0) to access your instance over SSH or RDP are acceptable for this short exercise, but are unsafe for production environments. You should authorize only a specific IP address or range of addresses to access your instance.

Step 7: Review Instance Launch and Select Key Pair

On the **Review Instance Launch** page, check the details of your instance, and make any necessary changes by choosing the appropriate **Edit** link.

When you are ready, choose **Launch**.

In the **Select an existing key pair or create a new key pair** dialog box, you can choose an existing key pair, or create a new one. For example, choose **Choose an existing key pair**, then select the key pair you created when getting set up. For more information, see [Amazon EC2 key pairs and Linux instances \(p. 985\)](#).

Important

If you choose the **Proceed without key pair** option, you won't be able to connect to the instance unless you choose an AMI that is configured to allow users another way to log in.

To launch your instance, select the acknowledgment check box, then choose **Launch Instances**.

(Optional) You can create a status check alarm for the instance (additional fees may apply). (If you're not sure, you can always add one later.) On the confirmation screen, choose **Create status check alarms** and follow the directions. For more information, see [Creating and editing status check alarms \(p. 701\)](#).

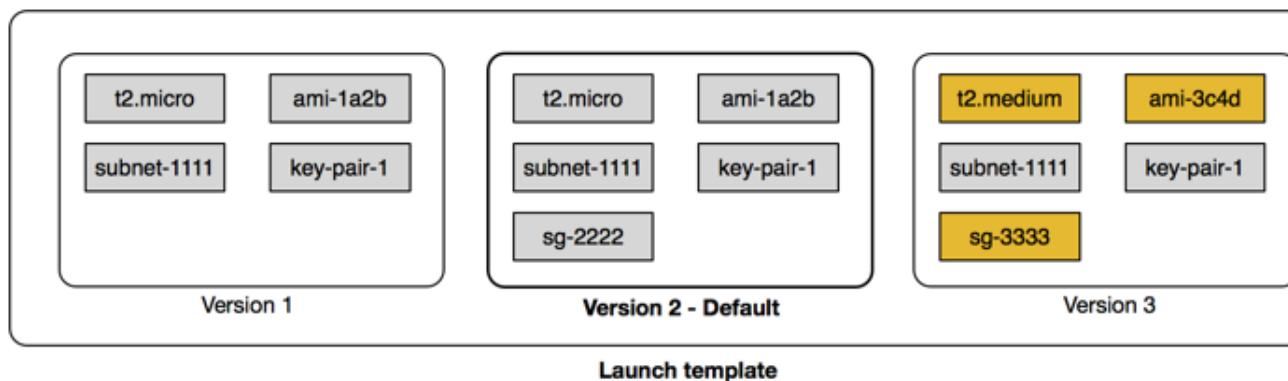
If the instance fails to launch or the state immediately goes to terminated instead of running, see [Troubleshooting instance launch issues \(p. 1244\)](#).

Launching an instance from a launch template

You can create a *launch template* that contains the configuration information to launch an instance. Launch templates enable you to store launch parameters so that you do not have to specify them every time you launch an instance. For example, a launch template can contain the AMI ID, instance type, and network settings that you typically use to launch instances. When you launch an instance using the Amazon EC2 console, an AWS SDK, or a command line tool, you can specify the launch template to use.

For each launch template, you can create one or more numbered *launch template versions*. Each version can have different launch parameters. When you launch an instance from a launch template, you can use any version of the launch template. If you do not specify a version, the default version is used. You can set any version of the launch template as the default version—by default, it's the first version of the launch template.

The following diagram shows a launch template with three versions. The first version specifies the instance type, AMI ID, subnet, and key pair to use to launch the instance. The second version is based on the first version and also specifies a security group for the instance. The third version uses different values for some of the parameters. Version 2 is set as the default version. If you launched an instance from this launch template, the launch parameters from version 2 would be used if no other version were specified.



Contents

- [Launch template restrictions \(p. 505\)](#)
- [Using launch templates to control launch parameters \(p. 505\)](#)
- [Controlling the use of launch templates \(p. 506\)](#)
- [Creating a launch template \(p. 506\)](#)
- [Managing launch template versions \(p. 515\)](#)
- [Launching an instance from a launch template \(p. 518\)](#)
- [Using launch templates with Amazon EC2 Auto Scaling \(p. 520\)](#)
- [Using launch templates with EC2 Fleet \(p. 520\)](#)
- [Using launch templates with Spot Fleet \(p. 520\)](#)
- [Deleting a launch template \(p. 521\)](#)

Launch template restrictions

The following rules apply to launch templates and launch template versions:

- You are limited to creating 5,000 launch templates per Region and 10,000 versions per launch template.
- Launch template parameters are optional. However, you must ensure that your request to launch an instance includes all required parameters. For example, if your launch template does not include an AMI ID, you must specify both the launch template and an AMI ID when you launch an instance.
- Launch template parameters are not fully validated when you create the launch template. If you specify incorrect values for parameters, or if you do not use supported parameter combinations, no instances can launch using this launch template. Ensure that you specify the correct values for the parameters and that you use supported parameter combinations. For example, to launch an instance in a placement group, you must specify a supported instance type.
- You can tag a launch template, but you cannot tag a launch template version.
- Launch template versions are numbered in the order in which they are created. When you create a launch template version, you cannot specify the version number yourself.

Using launch templates to control launch parameters

A launch template can contain all or some of the parameters to launch an instance. When you launch an instance using a launch template, you can override parameters that are specified in the launch template. Or, you can specify additional parameters that are not in the launch template.

Note

You cannot remove launch template parameters during launch (for example, you cannot specify a null value for the parameter). To remove a parameter, create a new version of the launch template without the parameter and use that version to launch the instance.

To launch instances, IAM users must have permissions to use the `ec2:RunInstances` action. You must also have permissions to create or use the resources that are created or associated with the instance. You can use resource-level permissions for the `ec2:RunInstances` action to control the launch parameters that users can specify. Alternatively, you can grant users permissions to launch an instance using a launch template. This enables you to manage launch parameters in a launch template rather than in an IAM policy, and to use a launch template as an authorization vehicle for launching instances. For example, you can specify that users can only launch instances using a launch template, and that they can only use a specific launch template. You can also control the launch parameters that users can override in the launch template. For example policies, see [Launch templates \(p. 951\)](#).

Controlling the use of launch templates

By default, IAM users do not have permissions to work with launch templates. You can create an IAM user policy that grants users permissions to create, modify, describe, and delete launch templates and launch template versions. You can also apply resource-level permissions to some launch template actions to control a user's ability to use specific resources for those actions. For more information, see the following example policies: [Example: Working with launch templates \(p. 962\)](#).

Take care when granting users permissions to use the `ec2:CreateLaunchTemplate` and `ec2:CreateLaunchTemplateVersion` actions. You cannot use resource-level permissions to control which resources users can specify in the launch template. To restrict the resources that are used to launch an instance, ensure that you grant permissions to create launch templates and launch template versions only to appropriate administrators.

Creating a launch template

Create a new launch template using parameters that you define, or use an existing launch template or an instance as the basis for a new launch template.

Tasks

- [Creating a new launch template using parameters you define \(p. 506\)](#)
- [Creating a launch template from an existing launch template \(p. 512\)](#)
- [Creating a launch template from an instance \(p. 513\)](#)

[Creating a new launch template using parameters you define](#)

New console

To create a new launch template using defined parameters using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**, and then choose **Create launch template**.
3. For **Launch template name**, enter a descriptive name for the launch template.
4. For **Template version description**, provide a brief description of the launch template version.
5. To tag the launch template on creation, expand **Template tags**, choose **Add tag**, and then enter a tag key and value pair.
6. For **Launch template contents**, provide the following information:
 - **AMI:** An AMI from which to launch the instance. To search through all available AMIs, choose **Search for AMI**. To select a commonly used AMI, choose **Quick Start**. Or, choose **AWS Marketplace** or **Community AMIs**. You can use an AMI that you own or [find a suitable AMI](#).

- **Instance type:** Ensure that the instance type is compatible with the AMI that you've specified. For more information, see [Instance types \(p. 200\)](#).
 - **Key pair name:** The key pair for the instance. For more information, see [Amazon EC2 key pairs and Linux instances \(p. 985\)](#).
 - **Network platform:** If applicable, whether to launch the instance into a VPC or EC2-Classic. If you choose **VPC**, specify the subnet in the **Network interfaces** section. If you choose **Classic**, ensure that the specified instance type is supported in EC2-Classic and specify the Availability Zone for the instance.
 - **Security groups:** One or more security groups to associate with the instance. If you add a network interface to the launch template, omit this setting and specify the security groups as part of the network interface specification. You cannot launch an instance from a launch template that specifies security groups and a network interface. For more information, see [Amazon EC2 security groups for Linux instances \(p. 999\)](#).
7. For **Storage (volumes)**, specify volumes to attach to the instance besides the volumes specified by the AMI (**Volume 1 (AMI Root)**). To add a new volume, choose **Add new volume**.
 - **Volume type:** The instance store or Amazon EBS volumes with which to associate your instance. The type of volume depends on the instance type that you've chosen. For more information, see [Amazon EC2 instance store \(p. 1188\)](#) and [Amazon EBS volumes \(p. 1021\)](#).
 - **Device name:** A device name for the volume.
 - **Snapshot:** The ID of the snapshot from which to create the volume.
 - **Size:** For Amazon EBS volumes, the storage size.
 - **Volume type:** For Amazon EBS volumes, the volume type. For more information, see [Amazon EBS volume types \(p. 1023\)](#).
 - **IOPS:** For the Provisioned IOPS SSD volume type, the number of I/O operations per second (IOPS) that the volume can support.
 - **Delete on termination:** For Amazon EBS volumes, whether to delete the volume when the instance is terminated. For more information, see [Preserving Amazon EBS volumes on instance termination \(p. 610\)](#).
 - **Encrypted:** If the instance type supports EBS encryption, you can enable encryption for the volume. If you have enabled encryption by default in this Region, encryption is enabled for you. For more information, see [Amazon EBS encryption \(p. 1121\)](#).
 - **Key:** The CMK to use for EBS encryption. You can specify the ARN of any customer master key (CMK) that you created using the AWS Key Management Service. If you specify a CMK, you must also use **Encrypted** to enable encryption.
 8. For **Resource tags**, specify [tags \(p. 1229\)](#) by providing key and value combinations. You can tag the instance, the volumes, Spot Instance requests, or all three.
 9. For **Network interfaces**, you can specify up to two [network interfaces \(p. 789\)](#) for the instance.
 - **Device index:** The device number for the network interface, for example, `eth0` for the primary network interface. If you leave the field blank, AWS creates the primary network interface.
 - **Network interface:** The ID of the network interface, or leave blank to let AWS create a new network interface.
 - **Description:** (Optional) A description for the new network interface.
 - **Subnet:** The subnet in which to create a new network interface. For the primary network interface (`eth0`), this is the subnet in which the instance is launched. If you've entered an existing network interface for `eth0`, the instance is launched in the subnet in which the network interface is located.

- **Auto-assign public IP:** Whether to automatically assign a public IP address to the network interface with the device index of eth0. This setting can only be enabled for a single, new network interface.
 - **Primary IP:** A private IPv4 address from the range of your subnet. Leave blank to let AWS choose a private IPv4 address for you.
 - **Secondary IP:** A secondary private IPv4 address from the range of your subnet. Leave blank to let AWS choose one for you.
 - **(IPv6-only) IPv6 IPs:** An IPv6 address from the range of the subnet.
 - **Security groups:** One or more security groups in your VPC with which to associate the network interface.
 - **Delete on termination:** Whether the network interface is deleted when the instance is deleted.
 - **Elastic Fabric Adapter:** Indicates whether the network interface is an Elastic Fabric Adapter. For more information, see [Elastic Fabric Adapter](#).
10. For **Advanced details**, expand the section to view the fields and specify any additional parameters for the instance.
- **Purchasing option:** The purchasing model. Choose **Request Spot Instances** to request Spot Instances at the Spot price, capped at the On-Demand price, and choose **Customize** to change the default Spot Instance settings. If you do not request a Spot Instance, EC2 launches an On-Demand Instance by default. For more information, see [Spot Instances \(p. 351\)](#).
 - **IAM instance profile:** An AWS Identity and Access Management (IAM) instance profile to associate with the instance. For more information, see [IAM roles for Amazon EC2 \(p. 974\)](#).
 - **Shutdown behavior:** Whether the instance should stop or terminate when shut down. For more information, see [Changing the instance initiated shutdown behavior \(p. 610\)](#).
 - **Stop - Hibernate behavior:** Whether the instance is enabled for hibernation. This field is only valid for instances that meet the hibernation prerequisites. For more information, see [Hibernate your Linux instance \(p. 591\)](#).
 - **Termination protection:** Whether to prevent accidental termination. For more information, see [Enabling termination protection \(p. 609\)](#).
 - **Detailed CloudWatch monitoring:** Whether to enable detailed monitoring of the instance using Amazon CloudWatch. Additional charges apply. For more information, see [Monitoring your instances using CloudWatch \(p. 715\)](#).
 - **Elastic inference:** An elastic inference accelerator to attach to your EC2 CPU instance. For more information, see [Working with Amazon Elastic Inference](#) in the [Amazon Elastic Inference Developer Guide](#).
 - **T2/T3 Unlimited:** Whether to enable applications to burst beyond the baseline for as long as needed. This field is only valid for T2, T3, and T3a instances. Additional charges may apply. For more information, see [Burstable performance instances \(p. 219\)](#).
 - **Placement group name:** Specify a placement group in which to launch the instance. Not all instance types can be launched in a placement group. For more information, see [Placement groups \(p. 869\)](#).
 - **EBS-optimized instance:** Provides additional, dedicated capacity for Amazon EBS I/O. Not all instance types support this feature, and additional charges apply. For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#).
 - **Capacity Reservation:** Specify whether to launch the instance into shared capacity, any open Capacity Reservation, a specific Capacity Reservation, or a Capacity Reservation group. For more information, see [Launching instances into an existing Capacity Reservation \(p. 483\)](#).
 - **Tenancy:** Choose whether to run your instance on shared hardware (**Shared**), isolated, dedicated hardware (**Dedicated**), or on a Dedicated Host (**Dedicated host**). If you choose to launch the instance onto a Dedicated Host, you can specify whether to launch the instance into a host resource group or you can target a specific Dedicated Host. Additional

charges may apply. For more information, see [Dedicated Instances \(p. 469\)](#) and [Dedicated Hosts \(p. 438\)](#).

- **RAM disk ID:** (Only valid for paravirtual (PV) AMIs) A RAM disk for the instance. If you have specified a kernel, you may need to specify a specific RAM disk with the drivers to support it.
- **Kernel ID:** (Only valid for paravirtual (PV) AMIs) A kernel for the instance.
- **License configurations:** You can launch instances against the specified license configuration to track your license usage. For more information, see [Create a License Configuration](#) in the *AWS License Manager User Guide*.
- **Metadata accessible:** Whether to enable or disable access to the instance metadata. For more information, see [Configuring the instance metadata service \(p. 659\)](#).
- **Metadata version:** If you enable access to the instance metadata, you can choose to require the use of Instance Metadata Service Version 2 when requesting instance metadata. For more information, see [Configuring instance metadata options for new instances \(p. 663\)](#).
- **Metadata response hop limit:** If you enable instance metadata, you can set the allowable number of network hops for the metadata token. For more information, see [Configuring the instance metadata service \(p. 659\)](#).
- **User data:** You can specify user data to configure an instance during launch, or to run a configuration script. For more information, see [Running commands on your Linux instance at launch \(p. 653\)](#).

11. Choose **Create launch template**.

Old console

To create a new launch template using defined parameters using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**, and then choose **Create launch template**.
3. For **Launch template name**, enter a descriptive name for the launch template. To tag the launch template on creation, choose **Show Tags**, **Add Tag**, and then enter a tag key and value pair.
4. For **Template version description**, provide a brief description of the launch template version.
5. For **Launch template contents**, provide the following information:
 - **AMI ID:** An AMI from which to launch the instance. To search through all available AMIs, choose **Search for AMI**. To select a commonly used AMI, choose **Quick Start**. Or, choose **AWS Marketplace** or **Community AMIs**. You can use an AMI that you own or [find a suitable AMI](#).
 - **Instance type:** Ensure that the instance type is compatible with the AMI that you've specified. For more information, see [Instance types \(p. 200\)](#).
 - **Key pair name:** The key pair for the instance. For more information, see [Amazon EC2 key pairs and Linux instances \(p. 985\)](#).
 - **Network type:** If applicable, whether to launch the instance into a VPC or EC2-Classic. If you choose **VPC**, specify the subnet in the **Network interfaces** section. If you choose **Classic**, ensure that the specified instance type is supported in EC2-Classic and specify the Availability Zone for the instance.
 - **Security Groups:** One or more security groups to associate with the instance. For more information, see [Amazon EC2 security groups for Linux instances \(p. 999\)](#).
6. For **Network interfaces**, you can specify up to two [network interfaces \(p. 789\)](#) for the instance.
 - **Device:** The device number for the network interface, for example, `eth0` for the primary network interface. If you leave the field blank, AWS creates the primary network interface.
 - **Network interface:** The ID of the network interface, or leave blank to let AWS create a new network interface.

- **Description:** (Optional) A description for the new network interface.
 - **Subnet:** The subnet in which to create a new network interface. For the primary network interface (`eth0`), this is the subnet in which the instance is launched. If you've entered an existing network interface for `eth0`, the instance is launched in the subnet in which the network interface is located.
 - **Auto-assign public IP:** Whether to automatically assign a public IP address to the network interface with the device index of `eth0`. This setting can only be enabled for a single, new network interface.
 - **Primary IP:** A private IPv4 address from the range of your subnet. Leave blank to let AWS choose a private IPv4 address for you.
 - **Secondary IP:** A secondary private IPv4 address from the range of your subnet. Leave blank to let AWS choose one for you.
 - **(IPv6-only) IPv6 IPs:** An IPv6 address from the range of the subnet.
 - **Security group ID:** The ID of a security group in your VPC with which to associate the network interface.
 - **Delete on termination:** Whether the network interface is deleted when the instance is deleted.
 - **Elastic Fabric Adapter:** Indicates whether the network interface is an Elastic Fabric Adapter. For more information, see [Elastic Fabric Adapter](#).
7. For **Storage (Volumes)**, specify volumes to attach to the instance besides the volumes specified by the AMI.
- **Volume type:** The instance store or Amazon EBS volumes with which to associate your instance. The type of volume depends on the instance type that you've chosen. For more information, see [Amazon EC2 instance store \(p. 1188\)](#) and [Amazon EBS volumes \(p. 1021\)](#).
 - **Device name:** A device name for the volume.
 - **Snapshot:** The ID of the snapshot from which to create the volume.
 - **Size:** For Amazon EBS volumes, the storage size.
 - **Volume type:** For Amazon EBS volumes, the volume type. For more information, see [Amazon EBS volume types \(p. 1023\)](#).
 - **IOPS:** For the Provisioned IOPS SSD volume type, the number of I/O operations per second (IOPS) that the volume can support.
 - **Delete on termination:** For Amazon EBS volumes, whether to delete the volume when the instance is terminated. For more information, see [Preserving Amazon EBS volumes on instance termination \(p. 610\)](#).
 - **Encrypted:** If the instance type supports EBS encryption, you can enable encryption for the volume. If you have enabled encryption by default in this Region, encryption is enabled for you. For more information, see [Amazon EBS encryption \(p. 1121\)](#).
 - **Key:** The CMK to use for EBS encryption. You can specify the ARN of any customer master key (CMK) that you created using the AWS Key Management Service. If you specify a CMK, you must also use **Encrypted** to enable encryption.
8. For **Instance tags**, specify [tags \(p. 1229\)](#) by providing key and value combinations. You can tag the instance, the volumes, or both.
9. For **Advanced Details**, expand the section to view the fields and specify any additional parameters for the instance.
- **Purchasing option:** The purchasing model. Choose **Request Spot instances** to request Spot Instances at the Spot price, capped at the On-Demand price, and choose **Customize Spot parameters** to change the default Spot Instance settings. If you do not request a Spot Instance, EC2 launches an On-Demand Instance by default. For more information, see [Spot Instances \(p. 351\)](#).

- **IAM instance profile:** An AWS Identity and Access Management (IAM) instance profile to associate with the instance. For more information, see [IAM roles for Amazon EC2 \(p. 974\)](#).
- **Shutdown behavior:** Whether the instance should stop or terminate when shut down. For more information, see [Changing the instance initiated shutdown behavior \(p. 610\)](#).
- **Stop - Hibernate behavior:** Whether the instance is enabled for hibernation. This field is only valid for instances that meet the hibernation prerequisites. For more information, see [Hibernate your Linux instance \(p. 591\)](#).
- **Termination protection:** Whether to prevent accidental termination. For more information, see [Enabling termination protection \(p. 609\)](#).
- **Monitoring:** Whether to enable detailed monitoring of the instance using Amazon CloudWatch. Additional charges apply. For more information, see [Monitoring your instances using CloudWatch \(p. 715\)](#).
- **T2/T3 Unlimited:** Whether to enable applications to burst beyond the baseline for as long as needed. This field is only valid for T2 and T3 instances. Additional charges may apply. For more information, see [Burstable performance instances \(p. 219\)](#).
- **Placement group name:** Specify a placement group in which to launch the instance. Not all instance types can be launched in a placement group. For more information, see [Placement groups \(p. 869\)](#).
- **EBS-optimized instance:** Provides additional, dedicated capacity for Amazon EBS I/O. Not all instance types support this feature, and additional charges apply. For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#).
- **Tenancy:** Choose whether to run your instance on shared hardware (**Shared**), isolated, dedicated hardware (**Dedicated**), or on a Dedicated Host (**Dedicated host**). If you choose to launch the instance onto a Dedicated Host, you can specify whether to launch the instance into a host resource group or you can target a specific Dedicated Host. Additional charges may apply. For more information, see [Dedicated Instances \(p. 469\)](#) and [Dedicated Hosts \(p. 438\)](#).
- **RAM disk ID:** A RAM disk for the instance. If you have specified a kernel, you may need to specify a specific RAM disk with the drivers to support it. Only valid for paravirtual (PV) AMIs.
- **Kernel ID:** A kernel for the instance. Only valid for paravirtual (PV) AMIs.
- **User data:** You can specify user data to configure an instance during launch, or to run a configuration script. For more information, see [Running commands on your Linux instance at launch \(p. 653\)](#).

10. Choose **Create launch template**.

AWS CLI

To create a launch template using the AWS CLI

- Use the [create-launch-template](#) command. The following example creates a launch template that specifies the following:
 - A tag for the launch template (`purpose=production`)
 - The instance type (`r4.4xlarge`) and AMI (`ami-8c1be5f6`) to launch
 - The number of cores (4) and threads per core (2) for a total of 8 vCPUs (4 cores x 2 threads)
 - The subnet in which to launch the instance (`subnet-7b16de0c`)

The template assigns a public IP address and an IPv6 address to the instance and creates a tag for the instance(`Name=webserver`).

```
aws ec2 create-launch-template \
--launch-template-name TemplateForWebServer \
```

```
--version-description WebVersion1 \
--tag-specifications 'ResourceType=launch-
template,Tags=[{Key=purpose,Value=production}]' \
--launch-template-data file://template-data.json
```

The following is an example template-data.json file.

```
{
    "NetworkInterfaces": [
        {
            "AssociatePublicIpAddress": true,
            "DeviceIndex": 0,
            "Ipv6AddressCount": 1,
            "SubnetId": "subnet-7b16de0c"
        }
    ],
    "ImageId": "ami-8c1be5f6",
    "InstanceType": "r4.4xlarge",
    "TagSpecifications": [
        {
            "ResourceType": "instance",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "webserver"
                }
            ]
        },
        "CpuOptions": {
            "CoreCount": 4,
            "ThreadsPerCore": 2
        }
    ]
}
```

The following is example output.

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-01238c059e3466abc",
        "LaunchTemplateName": "TemplateForWebServer",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:root",
        "CreateTime": "2017-11-27T09:13:24.000Z"
    }
}
```

Creating a launch template from an existing launch template

New console

To create a launch template from an existing launch template using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**, and then choose **Create launch template**.
3. For **Launch template name**, enter a descriptive name for the launch template.
4. For **Template version description**, provide a brief description of the launch template version.
5. To tag the launch template on creation, expand **Template tags**, choose **Add tag**, and then enter a tag key and value pair.
6. Expand **Source template**, and for **Launch template name** choose a launch template on which to base the new launch template.
7. For **Source template version**, choose the launch template version on which to base the new launch template.

8. Adjust any launch parameters as required, and then choose **Create launch template**.

Old console

To create a launch template from an existing launch template using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. Choose **Create launch template**. Provide a name, description, and tags for the launch template.
4. For **Source template**, choose a launch template on which to base the new launch template.
5. For **Source template version**, choose the launch template version on which to base the new launch template.
6. Adjust any launch parameters as required, and then choose **Create launch template**.

[Creating a launch template from an instance](#)

New console

To create a launch template from an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, and choose **Actions, Create template from instance**.
4. Provide a name, description, and tags, and adjust the launch parameters as required.

Note

When you create a launch template from an instance, the instance's network interface IDs and IP addresses are not included in the template.

5. Choose **Create launch template**.

Old console

To create a launch template from an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, and choose **Actions, Create Template From Instance**.
4. Provide a name, description, and tags, and adjust the launch parameters as required.

Note

When you create a launch template from an instance, the instance's network interface IDs and IP addresses are not included in the template.

5. Choose **Create Template From Instance**.

AWS CLI

You can use the AWS CLI to create a launch template from an existing instance by first getting the launch template data from an instance, and then creating a launch template using the launch template data.

To get launch template data from an instance using the AWS CLI

- Use the `get-launch-template-data` command and specify the instance ID. You can use the output as a base to create a new launch template or launch template version. By default, the output

includes a top-level `LaunchTemplateData` object, which cannot be specified in your launch template data. Use the `--query` option to exclude this object.

```
aws ec2 get-launch-template-data \
--instance-id i-0123d646e8048babc \
--query "LaunchTemplateData"
```

The following is example output.

```
{
    "Monitoring": {},
    "ImageId": "ami-8c1be5f6",
    "BlockDeviceMappings": [
        {
            "DeviceName": "/dev/xvda",
            "Ebs": {
                "DeleteOnTermination": true
            }
        }
    ],
    "EbsOptimized": false,
    "Placement": {
        "Tenancy": "default",
        "GroupName": "",
        "AvailabilityZone": "us-east-1a"
    },
    "InstanceType": "t2.micro",
    "NetworkInterfaces": [
        {
            "Description": "",
            "NetworkInterfaceId": "eni-35306abc",
            "PrivateIpAddresses": [
                {
                    "Primary": true,
                    "PrivateIpAddress": "10.0.0.72"
                }
            ],
            "SubnetId": "subnet-7b16de0c",
            "Groups": [
                "sg-7c227019"
            ],
            "Ipv6Addresses": [
                {
                    "Ipv6Address": "2001:db8:1234:1a00::123"
                }
            ],
            "PrivateIpAddress": "10.0.0.72"
        }
    ]
}
```

You can write the output directly to a file, for example:

```
aws ec2 get-launch-template-data \
--instance-id i-0123d646e8048babc \
--query "LaunchTemplateData" >> instance-data.json
```

To create a launch template using launch template data

Use the [create-launch-template](#) command to create a launch template using the output from the previous procedure. For more information about creating a launch template using the AWS CLI, see [Creating a new launch template using parameters you define \(p. 506\)](#).

Managing launch template versions

You can create launch template versions for a specific launch template, set the default version, describe a launch template version, and delete versions that you no longer require.

Tasks

- [Creating a launch template version \(p. 515\)](#)
- [Setting the default launch template version \(p. 516\)](#)
- [Describing a launch template version \(p. 517\)](#)
- [Deleting a launch template version \(p. 517\)](#)

Creating a launch template version

When you create a launch template version, you can specify new launch parameters or use an existing version as the base for the new version. For more information about the launch parameters, see [Creating a launch template \(p. 506\)](#).

New console

To create a launch template version using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. Select a launch template, and then choose **Actions, Modify template (Create new version)**.
4. For **Template version description**, enter a description for the launch template version.
5. (Optional) Expand **Source template** and select a version of the launch template to use as a base for the new launch template version. The new launch template version inherits the launch parameters from this launch template version.
6. Modify the launch parameters as required, and choose **Create launch template**.

Old console

To create a launch template version using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. Choose **Create launch template**.
4. For **What would you like to do**, choose **Create a new template version**
5. For **Launch template name**, select the name of the existing launch template from the list.
6. For **Template version description**, enter a description for the launch template version.
7. (Optional) Select a version of the launch template, or a version of a different launch template, to use as a base for the new launch template version. The new launch template version inherits the launch parameters from this launch template version.
8. Modify the launch parameters as required, and choose **Create launch template**.

AWS CLI

To create a launch template version using the AWS CLI

- Use the [create-launch-template-version](#) command. You can specify a source version on which to base the new version. The new version inherits the launch parameters from this version, and you can override parameters using `--launch-template-data`. The following example creates a new version based on version 1 of the launch template and specifies a different AMI ID.

```
aws ec2 create-launch-template-version \
--launch-template-id lt-0abcd290751193123 \
--version-description WebVersion2 \
--source-version 1 \
--launch-template-data "ImageId=ami-c998b6b2"
```

Setting the default launch template version

You can set the default version for the launch template. When you launch an instance from a launch template and do not specify a version, the instance is launched using the parameters of the default version.

New console

To set the default launch template version using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. Select the launch template and choose **Actions, Set default version**.
4. For **Template version**, select the version number to set as the default version and choose **Set as default version**.

Old console

To set the default launch template version using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. Select the launch template and choose **Actions, Set default version**.
4. For **Default version**, select the version number and choose **Set as default version**.

AWS CLI

To set the default launch template version using the AWS CLI

- Use the [modify-launch-template](#) command and specify the version that you want to set as the default.

```
aws ec2 modify-launch-template \
--launch-template-id lt-0abcd290751193123 \
--default-version 2
```

Describing a launch template version

Using the console, you can view all the versions of the selected launch template, or get a list of the launch templates whose latest or default version matches a specific version number. Using the AWS CLI, you can describe all versions, individual versions, or a range of versions of a specified launch template. You can also describe all the latest versions or all the default versions of all the launch templates in your account.

New console

To describe a launch template version using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. You can view a version of a specific launch template, or get a list of the launch templates whose latest or default version matches a specific version number.
 - To view a version of a launch template: Select the launch template. On the **Versions** tab, from **Version**, select a version to view its details.
 - To get a list of all the launch templates whose latest version matches a specific version number: From the search bar, choose **Latest version**, and then choose a version number.
 - To get a list of all the launch templates whose default version matches a specific version number: From the search bar, choose **Default version**, and then choose a version number.

AWS CLI

To describe a launch template version using the AWS CLI

- Use the [describe-launch-template-versions](#) command and specify the version numbers. In the following example, versions 1 and 3 are specified.

```
aws ec2 describe-launch-template-versions \
    --launch-template-id lt-abcd290751193123 \
    --versions 1 3
```

To describe all the latest and default launch template versions in your account using the AWS CLI

- Use the [describe-launch-template-versions](#) command and specify `$Latest`, `$Default`, or both. You must omit the launch template ID and name in the call. You cannot specify version numbers.

```
aws ec2 describe-launch-template-versions \
    --versions "$Latest,$Default"
```

Deleting a launch template version

If you no longer require a launch template version, you can delete it. You cannot replace the version number after you delete it. You cannot delete the default version of the launch template; you must first assign a different version as the default.

New console

To delete a launch template version using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. Select the launch template and choose **Actions, Delete template version**.
4. Select the version to delete and choose **Delete**.

Old console

To delete a launch template version using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. Select the launch template and choose **Actions, Delete template version**.
4. Select the version to delete and choose **Delete launch template version**.

AWS CLI

To delete a launch template version using the AWS CLI

- Use the [delete-launch-template-versions](#) command and specify the version numbers to delete.

```
aws ec2 delete-launch-template-versions \
  --launch-template-id lt-0abcd290751193123 \
  --versions 1
```

Launching an instance from a launch template

You can use the parameters contained in a launch template to launch an instance. You have the option to override or add launch parameters before you launch the instance.

Instances that are launched using a launch template are automatically assigned two tags with the keys `aws:ec2launchtemplate:id` and `aws:ec2launchtemplate:version`. You cannot remove or edit these tags.

New console

To launch an instance from a launch template using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. Select the launch template and choose **Actions, Launch instance from template**.
4. For **Source template version**, select the launch template version to use.
5. For **Number of instances**, specify the number of instances to launch.
6. (Optional) You can override or add launch template parameters by changing and adding parameters in the **Instance details** section.
7. Choose **Launch instance from template**.

Old console

To launch an instance from a launch template using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. Select the launch template and choose **Actions, Launch instance from template**.
4. Select the launch template version to use.
5. (Optional) You can override or add launch template parameters by changing and adding parameters in the **Instance details** section.
6. Choose **Launch instance from template**.

AWS CLI

To launch an instance from a launch template using the AWS CLI

- Use the `run-instances` command and specify the `--launch-template` parameter. Optionally specify the launch template version to use. If you don't specify the version, the default version is used.

```
aws ec2 run-instances \
--launch-template LaunchTemplateId=lt-0abcd290751193123,Version=1
```

- To override a launch template parameter, specify the parameter in the `run-instances` command. The following example overrides the instance type that's specified in the launch template (if any).

```
aws ec2 run-instances \
--launch-template LaunchTemplateId=lt-0abcd290751193123 \
--instance-type t2.small
```

- If you specify a nested parameter that's part of a complex structure, the instance is launched using the complex structure as specified in the launch template plus any additional nested parameters that you specify.

In the following example, the instance is launched with the tag `Owner=TeamA` as well as any other tags that are specified in the launch template. If the launch template has an existing tag with a key of `Owner`, the value is replaced with `TeamA`.

```
aws ec2 run-instances \
--launch-template LaunchTemplateId=lt-0abcd290751193123 \
--tag-specifications "ResourceType=instance,Tags=[{Key=Owner,Value=TeamA}]"
```

In the following example, the instance is launched with a volume with the device name `/dev/xvdb` as well as any other block device mappings that are specified in the launch template. If the launch template has an existing volume defined for `/dev/xvdb`, its values are replaced with the specified values.

```
aws ec2 run-instances \
--launch-template LaunchTemplateId=lt-0abcd290751193123 \
--block-device-mappings "DeviceName=/dev/xvdb,Ebs={VolumeSize=20,VolumeType=gp2}"
```

If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see [Troubleshooting instance launch issues \(p. 1244\)](#).

Using launch templates with Amazon EC2 Auto Scaling

You can create an Auto Scaling group and specify a launch template to use for the group. When Amazon EC2 Auto Scaling launches instances in the Auto Scaling group, it uses the launch parameters defined in the associated launch template. For more information, see [Creating an Auto Scaling Group Using a Launch Template](#) in the *Amazon EC2 Auto Scaling User Guide*.

Before you can create an Auto Scaling group using a launch template, you must create a launch template that includes the parameters required to launch an instance in an Auto Scaling group, such as the ID of the AMI. The new console provides guidance to help you create a template that you can use with Auto Scaling.

To create a launch template to use with Auto Scaling using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**, and then choose **Create launch template**.
3. For **Launch template name**, enter a descriptive name for the launch template.
4. For **Template version description**, provide a brief description of the launch template version.
5. Under **Auto Scaling guidance**, select the checkbox to have Amazon EC2 provide guidance to help create a template to use with Auto Scaling.
6. Modify the launch parameters as required. Because you selected Auto Scaling guidance, some fields are required and some fields are not available. For considerations to keep in mind when creating a launch template, and for information about how to configure the launch parameters for Auto Scaling, see [Creating a Launch Template for an Auto Scaling Group](#) in the *Amazon EC2 Auto Scaling User Guide*.
7. Choose **Create launch template**.
8. (Optional) To create an Auto Scaling group using this launch template, in the **Next steps** page, choose **Create Auto Scaling group**.

To create or update an Amazon EC2 Auto Scaling group with a launch template using the AWS CLI

- Use the `create-auto-scaling-group` or the `update-auto-scaling-group` command and specify the `--launch-template` parameter.

Using launch templates with EC2 Fleet

You can create an EC2 Fleet request and specify a launch template in the instance configuration. When Amazon EC2 fulfills the EC2 Fleet request, it uses the launch parameters defined in the associated launch template. You can override some of the parameters that are specified in the launch template.

For more information, see [Creating an EC2 Fleet \(p. 543\)](#).

To create an EC2 Fleet with a launch template using the AWS CLI

- Use the `create-fleet` command. Use the `--launch-template-configs` parameter to specify the launch template and any overrides for the launch template.

Using launch templates with Spot Fleet

You can create a Spot Fleet request and specify a launch template in the instance configuration. When Amazon EC2 fulfills the Spot Fleet request, it uses the launch parameters defined in the associated launch template. You can override some of the parameters that are specified in the launch template.

For more information, see [Spot Fleet requests \(p. 385\)](#).

To create a Spot Fleet request with a launch template using the AWS CLI

- Use the [request-spot-fleet](#) command. Use the `LaunchTemplateConfigs` parameter to specify the launch template and any overrides for the launch template.

Deleting a launch template

If you no longer require a launch template, you can delete it. Deleting a launch template deletes all of its versions.

New console

To delete a launch template (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. Select the launch template and choose **Actions, Delete template**.
4. Enter **Delete** to confirm deletion, and then choose **Delete**.

Old console

To delete a launch template (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. Select the launch template and choose **Actions, Delete template**.
4. Choose **Delete launch template**.

AWS CLI

To delete a launch template (AWS CLI)

- Use the [delete-launch-template](#) (AWS CLI) command and specify the launch template.

```
aws ec2 delete-launch-template --launch-template-id lt-01238c059e3466abc
```

Launching an instance using parameters from an existing instance

The Amazon EC2 console provides a **Launch more like this** wizard option that enables you to use a current instance as a base for launching other instances. This option automatically populates the Amazon EC2 launch wizard with certain configuration details from the selected instance.

Note

The **Launch more like this** wizard option does not clone your selected instance; it only replicates some configuration details. To create a copy of your instance, first create an AMI from it, then launch more instances from the AMI.

Alternatively, create a [launch template \(p. 504\)](#) to store the launch parameters for your instances.

The following configuration details are copied from the selected instance into the launch wizard:

- AMI ID
- Instance type
- Availability Zone, or the VPC and subnet in which the selected instance is located
- Public IPv4 address. If the selected instance currently has a public IPv4 address, the new instance receives a public IPv4 address - regardless of the selected instance's default public IPv4 address setting. For more information about public IPv4 addresses, see [Public IPv4 addresses and external DNS hostnames \(p. 760\)](#).
- Placement group, if applicable
- IAM role associated with the instance, if applicable
- Shutdown behavior setting (stop or terminate)
- Termination protection setting (true or false)
- CloudWatch monitoring (enabled or disabled)
- Amazon EBS-optimization setting (true or false)
- Tenancy setting, if launching into a VPC (shared or dedicated)
- Kernel ID and RAM disk ID, if applicable
- User data, if specified
- Tags associated with the instance, if applicable
- Security groups associated with the instance

The following configuration details are not copied from your selected instance. Instead, the wizard applies their default settings or behavior:

- Number of network interfaces: The default is one network interface, which is the primary network interface (eth0).
- Storage: The default storage configuration is determined by the AMI and the instance type.

To use your current instance as a template

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance you want to use, and then choose **Actions, Launch more like this**.
4. The launch wizard opens on the **Review Instance Launch** page. You can check the details of your instance, and make any necessary changes by clicking the appropriate **Edit** link.

When you are ready, choose **Launch** to select a key pair and launch your instance.

5. If the instance fails to launch or the state immediately goes to terminated instead of running, see [Troubleshooting instance launch issues \(p. 1244\)](#).

Launching an AWS Marketplace instance

You can subscribe to an AWS Marketplace product and launch an instance from the product's AMI using the Amazon EC2 launch wizard. For more information about paid AMIs, see [Paid AMIs \(p. 119\)](#). To cancel your subscription after launch, you first have to terminate all instances running from it. For more information, see [Managing your AWS Marketplace subscriptions \(p. 122\)](#).

To launch an instance from the AWS Marketplace using the launch wizard

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the Amazon EC2 dashboard, choose **Launch instance**.

3. On the **Choose an Amazon Machine Image (AMI)** page, choose the **AWS Marketplace** category on the left. Find a suitable AMI by browsing the categories, or using the search functionality. Choose **Select** to choose your product.
4. A dialog displays an overview of the product you've selected. You can view the pricing information, as well as any other information that the vendor has provided. When you're ready, choose **Continue**.

Note

You are not charged for using the product until you have launched an instance with the AMI. Take note of the pricing for each supported instance type, as you will be prompted to select an instance type on the next page of the wizard. Additional taxes may also apply to the product.

5. On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. When you're done, choose **Next: Configure Instance Details**.
6. On the next pages of the wizard, you can configure your instance, add storage, and add tags. For more information about the different options you can configure, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#). Choose **Next** until you reach the **Configure Security Group** page.

The wizard creates a new security group according to the vendor's specifications for the product. The security group may include rules that allow all IPv4 addresses (0.0.0.0/0) access on SSH (port 22) on Linux or RDP (port 3389) on Windows. We recommend that you adjust these rules to allow only a specific address or range of addresses to access your instance over those ports.

When you are ready, choose **Review and Launch**.

7. On the **Review Instance Launch** page, check the details of the AMI from which you're about to launch the instance, as well as the other configuration details you set up in the wizard. When you're ready, choose **Launch** to select or create a key pair, and launch your instance.
8. Depending on the product you've subscribed to, the instance may take a few minutes or more to launch. You are first subscribed to the product before your instance can launch. If there are any problems with your credit card details, you will be asked to update your account details. When the launch confirmation page displays, choose **View Instances** to go to the Instances page.

Note

You are charged the subscription price as long as your instance is running, even if it is idle. If your instance is stopped, you may still be charged for storage.

9. When your instance is in the `running` state, you can connect to it. To do this, select your instance in the list and choose **Connect**. Follow the instructions in the dialog. For more information about connecting to your instance, see [Connect to your Linux instance \(p. 563\)](#).

Important

Check the vendor's usage instructions carefully, as you may need to use a specific user name to log in to the instance. For more information about accessing your subscription details, see [Managing your AWS Marketplace subscriptions \(p. 122\)](#).

10. If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see [Troubleshooting instance launch issues \(p. 1244\)](#).

Launching an AWS Marketplace AMI instance using the API and CLI

To launch instances from AWS Marketplace products using the API or command line tools, first ensure that you are subscribed to the product. You can then launch an instance with the product's AMI ID using the following methods:

Method	Documentation
AWS CLI	Use the run-instances command, or see the following topic for more information: Launching an Instance .

Method	Documentation
AWS Tools for Windows PowerShell	Use the New-EC2Instance command, or see the following topic for more information: Launch an Amazon EC2 Instance Using Windows PowerShell
Query API	Use the RunInstances request.

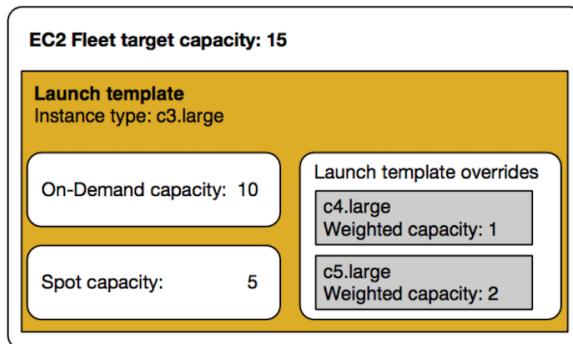
Launching instances using an EC2 Fleet

An *EC2 Fleet* contains the configuration information to launch a fleet—or group—of instances. In a single API call, a fleet can launch multiple instance types across multiple Availability Zones, using the On-Demand Instance, Reserved Instance, and Spot Instance purchasing options together. Using EC2 Fleet, you can:

- Define separate On-Demand and Spot capacity targets and the maximum amount you’re willing to pay per hour
- Specify the instance types that work best for your applications
- Specify how Amazon EC2 should distribute your fleet capacity within each purchasing option

You can also set a maximum amount per hour that you’re willing to pay for your fleet, and EC2 Fleet launches instances until it reaches the maximum amount. When the maximum amount you’re willing to pay is reached, the fleet stops launching instances even if it hasn’t met the target capacity.

The EC2 Fleet attempts to launch the number of instances that are required to meet the target capacity specified in your request. If you specified a total maximum price per hour, it fulfills the capacity until it reaches the maximum amount that you’re willing to pay. The fleet can also attempt to maintain its target Spot capacity if your Spot Instances are interrupted. For more information, see [How Spot Instances work \(p. 356\)](#).



You can specify an unlimited number of instance types per EC2 Fleet. Those instance types can be provisioned using both On-Demand and Spot purchasing options. You can also specify multiple Availability Zones, specify different maximum Spot prices for each instance, and choose additional Spot options for each fleet. Amazon EC2 uses the specified options to provision capacity when the fleet launches.

While the fleet is running, if Amazon EC2 reclaims a Spot Instance because of a price increase or instance failure, EC2 Fleet can try to replace the instances with any of the instance types that you specify. This makes it easier to regain capacity during a spike in Spot pricing. You can develop a flexible and elastic resourcing strategy for each fleet. For example, within specific fleets, your primary capacity can be On-Demand supplemented with less-expensive Spot capacity if available.

If you have Reserved Instances and you specify On-Demand Instances in your fleet, EC2 Fleet uses your Reserved Instances. For example, if your fleet specifies an On-Demand Instance as `c4.large`, and you have Reserved Instances for `c4.large`, you receive the Reserved Instance pricing.

There is no additional charge for using EC2 Fleet. You pay only for the EC2 instances that the fleet launches for you.

Contents

- [EC2 Fleet limitations \(p. 525\)](#)
- [EC2 Fleet limits \(p. 525\)](#)
- [Burstable performance instances \(p. 525\)](#)
- [EC2 Fleet configuration strategies \(p. 526\)](#)
- [Managing an EC2 Fleet \(p. 535\)](#)

EC2 Fleet limitations

The following limitations apply to EC2 Fleet:

- EC2 Fleet is available only through the API or AWS CLI.
- An EC2 Fleet request can't span AWS Regions. You need to create a separate EC2 Fleet for each Region.
- An EC2 Fleet request can't span different subnets from the same Availability Zone.

EC2 Fleet limits

The usual Amazon EC2 limits apply to instances launched by an EC2 Fleet, such as Spot request price limits, instance limits, and volume limits. In addition, the following limits apply:

- The number of active EC2 Fleets per AWS Region: 1,000 * †
- The number of Spot Instance pools (unique combination of instance type and subnet): 300* ‡
- The size of the user data in a launch specification: 16 KB †
- The target capacity per EC2 Fleet: 10,000
- The target capacity across all EC2 Fleets in a Region: 100,000 *
- An EC2 Fleet request can't span Regions.
- An EC2 Fleet request can't span different subnets from the same Availability Zone.

If you need more than the default limits for target capacity, complete the AWS Support Center [Create case](#) form to request a limit increase. For **Limit type**, choose **EC2 Fleet**, choose a Region, and then choose **Target Fleet Capacity per Fleet (in units)** or **Target Fleet Capacity per Region (in units)**, or both.

* These limits apply to both your EC2 Fleets and your Spot Fleets.

† These are hard limits. You cannot request a limit increase for these limits.

‡ This limit only applies to fleets of type `request` or `maintain`. This limit does not apply to instant fleets.

Burstable performance instances

If you launch your Spot Instances using a [burstable performance instance type \(p. 219\)](#), and if you plan to use your burstable performance Spot Instances immediately and for a short duration, with no idle time for accruing CPU credits, we recommend that you launch them in [Standard mode \(p. 231\)](#) to avoid paying higher costs. If you launch burstable performance Spot Instances in [Unlimited mode \(p. 223\)](#) and burst CPU immediately, you'll spend surplus credits for bursting. If you use the instance for a short

duration, the instance doesn't have time to accrue CPU credits to pay down the surplus credits, and you are charged for the surplus credits when you terminate the instance.

Unlimited mode is suitable for burstable performance Spot Instances only if the instance runs long enough to accrue CPU credits for bursting. Otherwise, paying for surplus credits makes burstable performance Spot Instances more expensive than using other instances. For more information, see [When to use unlimited mode versus fixed CPU \(p. 224\)](#).

Launch credits are meant to provide a productive initial launch experience for T2 instances by providing sufficient compute resources to configure the instance. Repeated launches of T2 instances to access new launch credits is not permitted. If you require sustained CPU, you can earn credits (by idling over some period), use [Unlimited mode \(p. 223\)](#) for T2 Spot Instances, or use an instance type with dedicated CPU.

EC2 Fleet configuration strategies

An *EC2 Fleet* is a group of On-Demand Instances and Spot Instances.

The EC2 Fleet attempts to launch the number of instances that are required to meet the target capacity that you specify in the fleet request. The fleet can comprise only On-Demand Instances, only Spot Instances, or a combination of both On-Demand Instances and Spot Instances. The request for Spot Instances is fulfilled if there is available capacity and the maximum price per hour for your request exceeds the Spot price. The fleet also attempts to maintain its target capacity if your Spot Instances are interrupted.

You can also set a maximum amount per hour that you're willing to pay for your fleet, and EC2 Fleet launches instances until it reaches the maximum amount. When the maximum amount you're willing to pay is reached, the fleet stops launching instances even if it hasn't met the target capacity.

A *Spot Instance pool* is a set of unused EC2 instances with the same instance type, operating system, Availability Zone, and network platform. When you create an EC2 Fleet, you can include multiple launch specifications, which vary by instance type, Availability Zone, subnet, and maximum price. The fleet selects the Spot Instance pools that are used to fulfill the request, based on the launch specifications included in your request, and the configuration of the request. The Spot Instances come from the selected pools.

An EC2 Fleet enables you to provision large amounts of EC2 capacity that makes sense for your application based on number of cores or instances, or amount of memory. For example, you can specify an EC2 Fleet to launch a target capacity of 200 instances, of which 130 are On-Demand Instances and the rest are Spot Instances.

Use the appropriate configuration strategies to create an EC2 Fleet that meets your needs.

Contents

- [Planning an EC2 Fleet \(p. 526\)](#)
- [EC2 Fleet request types \(p. 527\)](#)
- [Allocation strategies for Spot Instances \(p. 527\)](#)
- [Configuring EC2 Fleet for On-Demand backup \(p. 529\)](#)
- [Maximum price overrides \(p. 530\)](#)
- [Control spending \(p. 530\)](#)
- [EC2 Fleet instance weighting \(p. 530\)](#)
- [Tutorial: Using EC2 Fleet with instance weighting \(p. 532\)](#)
- [Tutorial: Using EC2 Fleet with On-Demand as the primary capacity \(p. 534\)](#)

Planning an EC2 Fleet

When planning your EC2 Fleet, we recommend that you do the following:

- Determine whether you want to create an EC2 Fleet that submits a synchronous or asynchronous one-time request for the desired target capacity, or one that maintains a target capacity over time. For more information, see [EC2 Fleet request types \(p. 527\)](#).
- Determine the instance types that meet your application requirements.
- If you plan to include Spot Instances in your EC2 Fleet, review [Spot Best Practices](#) before you create the fleet. Use these best practices when you plan your fleet so that you can provision the instances at the lowest possible price.
- Determine the target capacity for your EC2 Fleet. You can set target capacity in instances or in custom units. For more information, see [EC2 Fleet instance weighting \(p. 530\)](#).
- Determine what portion of the EC2 Fleet target capacity must be On-Demand capacity and Spot capacity. You can specify 0 for On-Demand capacity or Spot capacity, or both.
- Determine your price per unit, if you are using instance weighting. To calculate the price per unit, divide the price per instance hour by the number of units (or weight) that this instance represents. If you are not using instance weighting, the default price per unit is the price per instance hour.
- Determine the maximum amount per hour that you're willing to pay for your fleet. For more information, see [Control spending \(p. 530\)](#).
- Review the possible options for your EC2 Fleet. For more information, see the [EC2 Fleet JSON configuration file reference \(p. 541\)](#). For EC2 Fleet configuration examples, see [EC2 Fleet example configurations \(p. 551\)](#).

EC2 Fleet request types

There are three types of EC2 Fleet requests:

instant

If you configure the request type as `instant`, EC2 Fleet places a synchronous one-time request for your desired capacity. In the API response, it returns the instances that launched, along with errors for those instances that could not be launched.

request

If you configure the request type as `request`, EC2 Fleet places an asynchronous one-time request for your desired capacity. Thereafter, if capacity is diminished because of Spot interruptions, the fleet does not attempt to replenish Spot Instances, nor does it submit requests in alternative Spot Instance pools if capacity is unavailable.

maintain

(Default) If you configure the request type as `maintain`, EC2 Fleet places an asynchronous request for your desired capacity, and maintains capacity by automatically replenishing any interrupted Spot Instances.

All three types of requests benefit from an allocation strategy. For more information, see [Allocation strategies for Spot Instances \(p. 527\)](#).

Allocation strategies for Spot Instances

The allocation strategy for your EC2 Fleet determines how it fulfills your request for Spot Instances from the possible Spot Instance pools represented by its launch specifications. The following are the allocation strategies that you can specify in your fleet:

lowest-price

The Spot Instances come from the pool with the lowest price. This is the default strategy.

diversified

The Spot Instances are distributed across all pools.

capacity-optimized

The Spot Instances come from the pool with optimal capacity for the number of instances that are launching.

InstancePoolsToUseCount

The Spot Instances are distributed across the number of Spot pools that you specify. This parameter is valid only when used in combination with `lowest-price`.

Maintaining target capacity

After Spot Instances are terminated due to a change in the Spot price or available capacity of a Spot Instance pool, an EC2 Fleet of type `maintain` launches replacement Spot Instances. If the allocation strategy is `lowest-price`, the fleet launches replacement instances in the pool where the Spot price is currently the lowest. If the allocation strategy is `lowest-price` in combination with `InstancePoolsToUseCount`, the fleet selects the Spot pools with the lowest price and launches Spot Instances across the number of Spot pools that you specify. If the allocation strategy is `capacity-optimized`, the fleet launches replacement instances in the pool that has the most available Spot Instance capacity. If the allocation strategy is `diversified`, the fleet distributes the replacement Spot Instances across the remaining pools.

Configuring EC2 Fleet for cost optimization

To optimize the costs for your use of Spot Instances, specify the `lowest-price` allocation strategy so that EC2 Fleet automatically deploys the least expensive combination of instance types and Availability Zones based on the current Spot price.

For On-Demand Instance target capacity, EC2 Fleet always selects the cheapest instance type based on the public On-Demand price, while continuing to follow the allocation strategy (either `lowest-price`, `capacity-optimized`, or `diversified`) for Spot Instances.

Configuring EC2 Fleet for cost optimization and diversification

To create a fleet of Spot Instances that is both cheap and diversified, use the `lowest-price` allocation strategy in combination with `InstancePoolsToUseCount`. EC2 Fleet automatically deploys the least expensive combination of instance types and Availability Zones based on the current Spot price across the number of Spot pools that you specify. This combination can be used to avoid the most expensive Spot Instances.

Configuring EC2 Fleet for capacity optimization

With Spot Instances, pricing changes slowly over time based on long-term trends in supply and demand, but capacity fluctuates in real time. The `capacity-optimized` strategy automatically launches Spot Instances into the most available pools by looking at real-time capacity data and predicting which are the most available. This works well for workloads such as big data and analytics, image and media rendering, machine learning, and high performance computing that may have a higher cost of interruption associated with restarting work and checkpointing. By offering the possibility of fewer interruptions, the `capacity-optimized` strategy can lower the overall cost of your workload.

Choosing the appropriate allocation strategy

You can optimize your fleet based on your use case.

If your fleet is small or runs for a short time, the probability that your Spot Instances will be interrupted is low, even with all of the instances in a single Spot Instance pool. Therefore, the `lowest-price` strategy is likely to meet your needs while providing the lowest cost.

If your fleet is large or runs for a long time, you can improve the availability of your fleet by distributing the Spot Instances across multiple pools. For example, if your EC2 Fleet specifies 10 pools and a target

capacity of 100 instances, the fleet launches 10 Spot Instances in each pool. If the Spot price for one pool exceeds your maximum price for this pool, only 10% of your fleet is affected. Using this strategy also makes your fleet less sensitive to increases in the Spot price in any one pool over time.

With the diversified strategy, the EC2 Fleet does not launch Spot Instances into any pools with a Spot price that is equal to or higher than the [On-Demand price](#).

To create a cheap and diversified fleet, use the lowest-price strategy in combination with `InstancePoolsToUseCount`. You can use a low or high number of Spot pools across which to allocate your Spot Instances. For example, if you run batch processing, we recommend specifying a low number of Spot pools (for example, `InstancePoolsToUseCount=2`) to ensure that your queue always has compute capacity while maximizing savings. If you run a web service, we recommend specifying a high number of Spot pools (for example, `InstancePoolsToUseCount=10`) to minimize the impact if a Spot Instance pool becomes temporarily unavailable.

If your fleet runs workloads that may have a higher cost of interruption associated with restarting work and checkpointing, then use the capacity-optimized strategy. This strategy offers the possibility of fewer interruptions, which can lower the overall cost of your workload.

Configuring EC2 Fleet for On-Demand backup

If you have urgent, unpredictable scaling needs, such as a news website that must scale during a major news event or game launch, we recommend that you specify alternative instance types for your On-Demand Instances, in the event that your preferred option does not have sufficient available capacity. For example, you might prefer `c5.2xlarge` On-Demand Instances, but if there is insufficient available capacity, you'd be willing to use some `c4.2xlarge` instances during peak load. In this case, EC2 Fleet attempts to fulfill all of your target capacity using `c5.2xlarge` instances, but if there is insufficient capacity, it automatically launches `c4.2xlarge` instances to fulfill the target capacity.

Prioritizing instance types for On-Demand capacity

When EC2 Fleet attempts to fulfill your On-Demand capacity, it defaults to launching the lowest-priced instance type first. If `AllocationStrategy` is set to prioritized, EC2 Fleet uses priority to determine which instance type to use first in fulfilling On-Demand capacity. The priority is assigned to the launch template override, and the highest priority is launched first.

For example, you have configured three launch template overrides, each with a different instance type: `c3.large`, `c4.large`, and `c5.large`. The On-Demand price for `c5.large` is less than the price for `c4.large`. `c3.large` is the cheapest. If you do not use priority to determine the order, the fleet fulfills On-Demand capacity by starting with `c3.large`, and then `c5.large`. Because you often have unused Reserved Instances for `c4.large`, you can set the launch template override priority so that the order is `c4.large`, `c3.large`, and then `c5.large`.

Using Capacity Reservations for On-Demand Instances

You can configure a fleet to use On-Demand Capacity Reservations first when launching On-Demand Instances by setting the usage strategy for Capacity Reservations to `use-capacity-reservations-first`. You can use this setting in conjunction with the allocation strategy for On-Demand Instances (`lowest-price` or `prioritized`).

When unused Capacity Reservations are used to fulfil On-Demand capacity:

- The fleet uses unused Capacity Reservations to fulfill On-Demand capacity up to the target On-Demand capacity.
- If multiple instance pools have unused Capacity Reservations, the On-Demand allocation strategy (`lowest-price` or `prioritized`) is applied.
- If the number of unused Capacity Reservations is less than the On-Demand target capacity, the remaining On-Demand target capacity is launched according to the On-Demand allocation strategy (`lowest-price` or `prioritized`).

You can only use unused On-Demand Capacity Reservations for fleets of type `instant`.

For examples of how to configure a fleet to use Capacity Reservations to fulfil On-Demand capacity, see [EC2 Fleet example configurations \(p. 551\)](#). For more information, see [On-Demand Capacity Reservations \(p. 474\)](#) and the [On-Demand Capacity Reservation FAQs](#).

Maximum price overrides

Each EC2 Fleet can either include a global maximum price, or use the default (the On-Demand price). The fleet uses this as the default maximum price for each of its launch specifications.

You can optionally specify a maximum price in one or more launch specifications. This price is specific to the launch specification. If a launch specification includes a specific price, the EC2 Fleet uses this maximum price, overriding the global maximum price. Any other launch specifications that do not include a specific maximum price still use the global maximum price.

Control spending

EC2 Fleet stops launching instances when it has met one of the following parameters: the `TotalTargetCapacity` or the `MaxTotalPrice` (the maximum amount you're willing to pay). To control the amount you pay per hour for your fleet, you can specify the `MaxTotalPrice`. When the maximum total price is reached, EC2 Fleet stops launching instances even if it hasn't met the target capacity.

The following examples show two different scenarios. In the first, EC2 Fleet stops launching instances when it has met the target capacity. In the second, EC2 Fleet stops launching instances when it has reached the maximum amount you're willing to pay (`MaxTotalPrice`).

Example: Stop launching instances when target capacity is reached

Given a request for `m4.large` On-Demand Instances, where:

- On-Demand Price: \$0.10 per hour
- `OnDemandTargetCapacity`: 10
- `MaxTotalPrice`: \$1.50

EC2 Fleet launches 10 On-Demand Instances because the total of \$1.00 (10 instances x \$0.10) does not exceed the `MaxTotalPrice` of \$1.50 for On-Demand Instances.

Example: Stop launching instances when maximum total price is reached

Given a request for `m4.large` On-Demand Instances, where:

- On-Demand Price: \$0.10 per hour
- `OnDemandTargetCapacity`: 10
- `MaxTotalPrice`: \$0.80

If EC2 Fleet launches the On-Demand target capacity (10 On-Demand Instances), the total cost per hour would be \$1.00. This is more than the amount (\$0.80) specified for `MaxTotalPrice` for On-Demand Instances. To prevent spending more than you're willing to pay, EC2 Fleet launches only 8 On-Demand Instances (below the On-Demand target capacity) because launching more would exceed the `MaxTotalPrice` for On-Demand Instances.

EC2 Fleet instance weighting

When you create an EC2 Fleet, you can define the capacity units that each instance type would contribute to your application's performance. You can then adjust your maximum price for each launch specification by using *instance weighting*.

By default, the price that you specify is *per instance hour*. When you use the instance weighting feature, the price that you specify is *per unit hour*. You can calculate your price per unit hour by dividing your price for an instance type by the number of units that it represents. EC2 Fleet calculates the number of instances to launch by dividing the target capacity by the instance weight. If the result isn't an integer, the fleet rounds it up to the next integer, so that the size of your fleet is not below its target capacity. The fleet can select any pool that you specify in your launch specification, even if the capacity of the instances launched exceeds the requested target capacity.

The following table includes examples of calculations to determine the price per unit for an EC2 Fleet with a target capacity of 10.

Instance type	Instance weight	Target capacity	Number of instances launched	Price per instance hour	Price per unit hour
r3.xlarge	2	10	5 (10 divided by 2)	\$0.05	\$0.025 (.05 divided by 2)
r3.8xlarge	8	10	2 (10 divided by 8, result rounded up)	\$0.10	\$0.0125 (.10 divided by 8)

Use EC2 Fleet instance weighting as follows to provision the target capacity that you want in the pools with the lowest price per unit at the time of fulfillment:

1. Set the target capacity for your EC2 Fleet either in instances (the default) or in the units of your choice, such as virtual CPUs, memory, storage, or throughput.
2. Set the price per unit.
3. For each launch specification, specify the weight, which is the number of units that the instance type represents toward the target capacity.

Instance weighting example

Consider an EC2 Fleet request with the following configuration:

- A target capacity of 24
- A launch specification with an instance type r3.2xlarge and a weight of 6
- A launch specification with an instance type c3.xlarge and a weight of 5

The weights represent the number of units that instance type represents toward the target capacity. If the first launch specification provides the lowest price per unit (price for r3.2xlarge per instance hour divided by 6), the EC2 Fleet would launch four of these instances (24 divided by 6).

If the second launch specification provides the lowest price per unit (price for c3.xlarge per instance hour divided by 5), the EC2 Fleet would launch five of these instances (24 divided by 5, result rounded up).

Instance weighting and allocation strategy

Consider an EC2 Fleet request with the following configuration:

- A target capacity of 30 Spot Instances
- A launch specification with an instance type `c3.2xlarge` and a weight of 8
- A launch specification with an instance type `m3.xlarge` and a weight of 8
- A launch specification with an instance type `r3.xlarge` and a weight of 8

The EC2 Fleet would launch four instances (30 divided by 8, result rounded up). With the lowest-price strategy, all four instances come from the pool that provides the lowest price per unit. With the diversified strategy, the fleet launches one instance in each of the three pools, and the fourth instance in whichever of the three pools provides the lowest price per unit.

Tutorial: Using EC2 Fleet with instance weighting

This tutorial uses a fictitious company called Example Corp to illustrate the process of requesting an EC2 Fleet using instance weighting.

Objective

Example Corp, a pharmaceutical company, wants to use the computational power of Amazon EC2 for screening chemical compounds that might be used to fight cancer.

Planning

Example Corp first reviews [Spot Best Practices](#). Next, Example Corp determines the requirements for their EC2 Fleet.

Instance types

Example Corp has a compute- and memory-intensive application that performs best with at least 60 GB of memory and eight virtual CPUs (vCPUs). They want to maximize these resources for the application at the lowest possible price. Example Corp decides that any of the following EC2 instance types would meet their needs:

Instance type	Memory (GiB)	vCPUs
<code>r3.2xlarge</code>	61	8
<code>r3.4xlarge</code>	122	16
<code>r3.8xlarge</code>	244	32

Target capacity in units

With instance weighting, target capacity can equal a number of instances (the default) or a combination of factors such as cores (vCPUs), memory (GiBs), and storage (GBs). By considering the base for their application (60 GB of RAM and eight vCPUs) as one unit, Example Corp decides that 20 times this amount would meet their needs. So the company sets the target capacity of their EC2 Fleet request to 20.

Instance weights

After determining the target capacity, Example Corp calculates instance weights. To calculate the instance weight for each instance type, they determine the units of each instance type that are required to reach the target capacity as follows:

- `r3.2xlarge` (61.0 GB, 8 vCPUs) = 1 unit of 20
- `r3.4xlarge` (122.0 GB, 16 vCPUs) = 2 units of 20

- r3.8xlarge (244.0 GB, 32 vCPUs) = 4 units of 20

Therefore, Example Corp assigns instance weights of 1, 2, and 4 to the respective launch configurations in their EC2 Fleet request.

Price per unit hour

Example Corp uses the [On-Demand price](#) per instance hour as a starting point for their price. They could also use recent Spot prices, or a combination of the two. To calculate the price per unit hour, they divide their starting price per instance hour by the weight. For example:

Instance type	On-Demand price	Instance weight	Price per unit hour
r3.2xLarge	\$0.7	1	\$0.7
r3.4xLarge	\$1.4	2	\$0.7
r3.8xLarge	\$2.8	4	\$0.7

Example Corp could use a global price per unit hour of \$0.7 and be competitive for all three instance types. They could also use a global price per unit hour of \$0.7 and a specific price per unit hour of \$0.9 in the r3.8xlarge launch specification.

Verifying permissions

Before creating an EC2 Fleet, Example Corp verifies that it has an IAM role with the required permissions. For more information, see [EC2 Fleet prerequisites \(p. 536\)](#).

Creating a launch template

Next, Example Corp creates a launch template. The launch template ID is used in the following step. For more information, see [Creating a launch template \(p. 506\)](#).

Creating the EC2 Fleet

Example Corp creates a file, config.json, with the following configuration for its EC2 Fleet. In the following example, replace the resource identifiers with your own resource identifiers.

```
{  
    "LaunchTemplateConfigs": [  
        {  
            "LaunchTemplateSpecification": {  
                "LaunchTemplateId": "lt-07b3bc7625cdab851",  
                "Version": "1"  
            },  
            "Overrides": [  
                {  
                    "InstanceType": "r3.2xlarge",  
                    "SubnetId": "subnet-482e4972",  
                    "WeightedCapacity": 1  
                },  
                {  
                    "InstanceType": "r3.4xlarge",  
                    "SubnetId": "subnet-482e4972",  
                    "WeightedCapacity": 2  
                },  
                {  
                    "InstanceType": "r3.8xlarge",  
                    "SubnetId": "subnet-482e4972",  
                    "WeightedCapacity": 4  
                }  
            ]  
        }  
    ]  
}
```

```
        "MaxPrice": "0.90",
        "SubnetId": "subnet-482e4972",
        "WeightedCapacity": 4
    }
]
},
"TargetCapacitySpecification": {
    "TotalTargetCapacity": 20,
    "DefaultTargetCapacityType": "spot"
}
}
```

Example Corp creates the EC2 Fleet using the following [create-fleet](#) command.

```
aws ec2 create-fleet \
--cli-input-json file://config.json
```

For more information, see [Creating an EC2 Fleet \(p. 543\)](#).

Fulfillment

The allocation strategy determines which Spot Instance pools your Spot Instances come from.

With the lowest-price strategy (which is the default strategy), the Spot Instances come from the pool with the lowest price per unit at the time of fulfillment. To provide 20 units of capacity, the EC2 Fleet launches either 20 `r3.2xlarge` instances (20 divided by 1), 10 `r3.4xlarge` instances (20 divided by 2), or 5 `r3.8xlarge` instances (20 divided by 4).

If Example Corp used the diversified strategy, the Spot Instances would come from all three pools. The EC2 Fleet would launch 6 `r3.2xlarge` instances (which provide 6 units), 3 `r3.4xlarge` instances (which provide 6 units), and 2 `r3.8xlarge` instances (which provide 8 units), for a total of 20 units.

Tutorial: Using EC2 Fleet with On-Demand as the primary capacity

This tutorial uses a fictitious company called ABC Online to illustrate the process of requesting an EC2 Fleet with On-Demand as the primary capacity, and Spot capacity if available.

Objective

ABC Online, a restaurant delivery company, wants to be able to provision Amazon EC2 capacity across EC2 instance types and purchasing options to achieve their desired scale, performance, and cost.

Planning

ABC Online requires a fixed capacity to operate during peak periods, but would like to benefit from increased capacity at a lower price. ABC Online determines the following requirements for their EC2 Fleet:

- On-Demand Instance capacity – ABC Online requires 15 On-Demand Instances to ensure that they can accommodate traffic at peak periods.
- Spot Instance capacity – ABC Online would like to improve performance, but at a lower price, by provisioning 5 Spot Instances.

Verifying permissions

Before creating an EC2 Fleet, ABC Online verifies that it has an IAM role with the required permissions. For more information, see [EC2 Fleet prerequisites \(p. 536\)](#).

Creating a launch template

Next, ABC Online creates a launch template. The launch template ID is used in the following step. For more information, see [Creating a launch template \(p. 506\)](#).

Creating the EC2 Fleet

ABC Online creates a file, config.json, with the following configuration for its EC2 Fleet. In the following example, replace the resource identifiers with your own resource identifiers.

```
{  
    "LaunchTemplateConfigs": [  
        {  
            "LaunchTemplateSpecification": {  
                "LaunchTemplateId": "lt-07b3bc7625cdab851",  
                "Version": "2"  
            }  
        }  
    ],  
    "TargetCapacitySpecification": {  
        "TotalTargetCapacity": 20,  
        "OnDemandTargetCapacity": 15,  
        "DefaultTargetCapacityType": "spot"  
    }  
}
```

ABC Online creates the EC2 Fleet using the following [create-fleet](#) command.

```
aws ec2 create-fleet \  
    --cli-input-json file://config.json
```

For more information, see [Creating an EC2 Fleet \(p. 543\)](#).

Fulfillment

The allocation strategy determines that the On-Demand capacity is always fulfilled, while the balance of the target capacity is fulfilled as Spot if there is capacity and availability.

Managing an EC2 Fleet

To use an EC2 Fleet, you create a request that includes the total target capacity, On-Demand capacity, Spot capacity, one or more launch specifications for the instances, and the maximum price that you are willing to pay. The fleet request must include a launch template that defines the information that the fleet needs to launch an instance, such as an AMI, instance type, subnet or Availability Zone, and one or more security groups. You can specify launch specification overrides for the instance type, subnet, Availability Zone, and maximum price you're willing to pay, and you can assign weighted capacity to each launch specification override.

If your fleet includes Spot Instances, Amazon EC2 can attempt to maintain your fleet target capacity as Spot prices change.

An EC2 Fleet request remains active until it expires or you delete it. When you delete a fleet, you can specify whether deletion terminates the instances in that fleet.

Contents

- [EC2 Fleet request states \(p. 536\)](#)
- [EC2 Fleet prerequisites \(p. 536\)](#)
- [EC2 Fleet health checks \(p. 539\)](#)

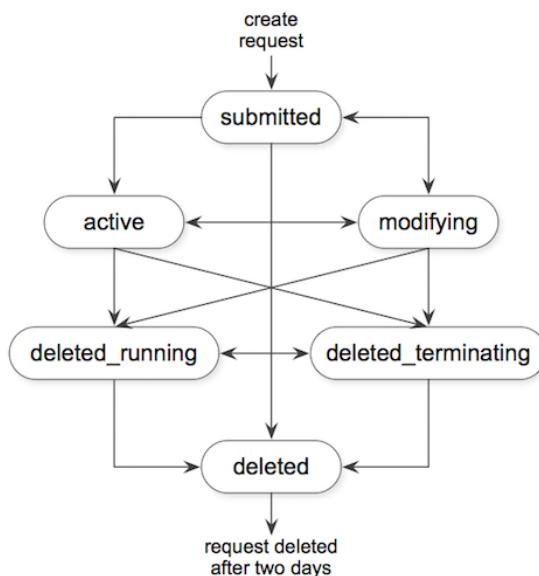
- [Generating an EC2 Fleet JSON configuration file \(p. 539\)](#)
- [Creating an EC2 Fleet \(p. 543\)](#)
- [Tagging an EC2 Fleet \(p. 546\)](#)
- [Monitoring your EC2 Fleet \(p. 547\)](#)
- [Modifying an EC2 Fleet \(p. 549\)](#)
- [Deleting an EC2 Fleet \(p. 550\)](#)
- [EC2 Fleet example configurations \(p. 551\)](#)

EC2 Fleet request states

An EC2 Fleet request can be in one of the following states:

- submitted – The EC2 Fleet request is being evaluated and Amazon EC2 is preparing to launch the target number of instances, which can include On-Demand Instances, Spot Instances, or both.
- active – The EC2 Fleet request has been validated and Amazon EC2 is attempting to maintain the target number of running instances. The request remains in this state until it is modified or deleted.
- modifying – The EC2 Fleet request is being modified. The request remains in this state until the modification is fully processed or the request is deleted. Only a maintain request type can be modified. This state does not apply to other request types.
- deleted_running – The EC2 Fleet request is deleted and does not launch additional instances. Its existing instances continue to run until they are interrupted or terminated. The request remains in this state until all instances are interrupted or terminated.
- deleted_terminating – The EC2 Fleet request is deleted and its instances are terminating. The request remains in this state until all instances are terminated.
- deleted – The EC2 Fleet is deleted and has no running instances. The request is deleted two days after its instances are terminated.

The following illustration represents the transitions between the EC2 Fleet request states. If you exceed your fleet limits, the request is deleted immediately.



EC2 Fleet prerequisites

To create an EC2 Fleet, the following prerequisites must be in place.

Launch template

A launch template includes information about the instances to launch, such as the instance type, Availability Zone, and the maximum price that you are willing to pay. For more information, see [Launching an instance from a launch template \(p. 504\)](#).

Service-linked role for EC2 Fleet

The **AWSServiceRoleForEC2Fleet** role grants the EC2 Fleet permission to request, launch, terminate, and tag instances on your behalf. Amazon EC2 uses this service-linked role to complete the following actions:

- **ec2:RunInstances** – Launch instances.
- **ec2:RequestSpotInstances** – Request Spot Instances.
- **ec2:TerminateInstances** – Terminate instances.
- **ec2:DescribeImages** – Describe Amazon Machine Images (AMIs) for the Spot Instances.
- **ec2:DescribeInstanceStatus** – Describe the status of the Spot Instances.
- **ec2:DescribeSubnets** – Describe the subnets for Spot Instances.
- **ec2:CreateTags** – Add tags to the EC2 Fleet, instances, and volumes.

Ensure that this role exists before you use the AWS CLI or an API to create an EC2 Fleet.

Note

An instant EC2 Fleet does not require this role.

To create the role, use the IAM console as follows.

To create the **AWSServiceRoleForEC2Fleet** role for EC2 Fleet

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**, and then choose **Create role**.
3. For **Select type of trusted entity**, choose **AWS service**.
4. For **Choose the service that will use this role**, choose **EC2 - Fleet**, and then choose **Next: Permissions**, **Next: Tags**, and **Next: Review**.
5. On the **Review** page, choose **Create role**.

If you no longer need to use EC2 Fleet, we recommend that you delete the **AWSServiceRoleForEC2Fleet** role. After this role is deleted from your account, you can create the role again if you create another fleet.

For more information, see [Using Service-Linked Roles](#) in the *IAM User Guide*.

Granting access to CMKs for use with encrypted AMIs and EBS snapshots

If you specify an [encrypted AMI \(p. 157\)](#) or an [encrypted Amazon EBS snapshot \(p. 1121\)](#) in your EC2 Fleet and you use a customer-managed customer master key (CMK) for encryption, you must grant the **AWSServiceRoleForEC2Fleet** role permission to use the CMK so that Amazon EC2 can launch instances on your behalf. To do this, you must add a grant to the CMK, as shown in the following procedure.

When providing permissions, grants are an alternative to key policies. For more information, see [Using Grants](#) and [Using Key Policies in AWS KMS](#) in the *AWS Key Management Service Developer Guide*.

To grant the **AWSServiceRoleForEC2Fleet** role permissions to use the CMK

- Use the `create-grant` command to add a grant to the CMK and to specify the principal (the **AWSServiceRoleForEC2Fleet** service-linked role) that is given permission to perform the operations

that the grant permits. The CMK is specified by the `key-id` parameter and the ARN of the CMK. The principal is specified by the `grantee-principal` parameter and the ARN of the `AWSServiceRoleForEC2Fleet` service-linked role.

```
aws kms create-grant \
    --region us-east-1 \
    --key-id arn:aws:kms:us-
east-1:444455566666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2Fleet \
    --operations "Decrypt" "Encrypt" "GenerateDataKey"
"GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
"ReEncryptTo"
```

EC2 Fleet and IAM users

If your IAM users will create or manage an EC2 Fleet, be sure to grant them the required permissions as follows.

To grant an IAM user permissions for EC2 Fleet

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**.
3. Choose **Create policy**.
4. On the **Create policy** page, choose the **JSON** tab, replace the text with the following, and choose **Review policy**.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam>ListRoles",
                "iam>PassRole",
                "iam>ListInstanceProfiles"
            ],
            "Resource": "*"
        }
    ]
}
```

The `ec2:*` grants an IAM user permission to call all Amazon EC2 API actions. To limit the user to specific Amazon EC2 API actions, specify those actions instead.

An IAM user must have permission to call the `iam>ListRoles` action to enumerate existing IAM roles, the `iam>PassRole` action to specify the EC2 Fleet role, and the `iam>ListInstanceProfiles` action to enumerate existing instance profiles.

(Optional) To enable an IAM user to create roles or instance profiles using the IAM console, you must also add the following actions to the policy:

- `iam>AddRoleToInstanceProfile`

- iam:AttachRolePolicy
 - iam>CreateInstanceProfile
 - iam>CreateRole
 - iam:GetRole
 - iam>ListPolicies
5. On the **Review policy** page, enter a policy name and description, and choose **Create policy**.
 6. In the navigation pane, choose **Users** and select the user.
 7. On the **Permissions** tab, choose **Add permissions**.
 8. Choose **Attach existing policies directly**. Select the policy that you created earlier and choose **Next: Review**.
 9. Choose **Add permissions**.

EC2 Fleet health checks

EC2 Fleet checks the health status of the instances in the fleet every two minutes. The health status of an instance is either **healthy** or **unhealthy**. The fleet determines the health status of an instance using the status checks provided by Amazon EC2. If the status of either the instance status check or the system status check is **impaired** for three consecutive health checks, the health status of the instance is **unhealthy**. Otherwise, the health status is **healthy**. For more information, see [Status checks for your instances \(p. 697\)](#).

You can configure your EC2 Fleet to replace unhealthy instances. After enabling health check replacement, an instance is replaced after its health status is reported as **unhealthy**. The fleet could go below its target capacity for up to a few minutes while an unhealthy instance is being replaced.

Requirements

- Health check replacement is supported only with EC2 Fleets that maintain a target capacity, not with one-time fleets.
- You can configure your EC2 Fleet to replace unhealthy instances only when you create it.
- IAM users can use health check replacement only if they have permission to call the `ec2:DescribeInstanceStatus` action.

Generating an EC2 Fleet JSON configuration file

To create an EC2 Fleet, you need only specify the launch template, total target capacity, and whether the default purchasing option is On-Demand or Spot. If you do not specify a parameter, the fleet uses the default value. To view the full list of fleet configuration parameters, you can generate a JSON file as follows.

To generate a JSON file with all possible EC2 Fleet parameters using the command line

- Use the `create-fleet` (AWS CLI) command and the `--generate-cli-skeleton` parameter to generate an EC2 Fleet JSON file:

```
aws ec2 create-fleet \
--generate-cli-skeleton
```

The following EC2 Fleet parameters are available:

```
{
  "DryRun": true,
  "ClientToken": "",
```

```
"SpotOptions": {  
    "AllocationStrategy": "lowest-price",  
    "InstanceInterruptionBehavior": "hibernate",  
    "InstancePoolsToUseCount": 0,  
    "SingleInstanceType": true,  
    "SingleAvailabilityZone": true,  
    "MaxTotalPrice": 0,  
    "MinTargetCapacity": 0  
},  
"OnDemandOptions": {  
    "AllocationStrategy": "prioritized",  
    "SingleInstanceType": true,  
    "SingleAvailabilityZone": true,  
    "MaxTotalPrice": 0,  
    "MinTargetCapacity": 0  
},  
"ExcessCapacityTerminationPolicy": "termination",  
"LaunchTemplateConfigs": [  
    {  
        "LaunchTemplateSpecification": {  
            "LaunchTemplateId": "",  
            "LaunchTemplateName": "",  
            "Version": ""  
        },  
        "Overrides": [  
            {  
                "InstanceType": "t2.micro",  
                "MaxPrice": "",  
                "SubnetId": "",  
                "AvailabilityZone": "",  
                "WeightedCapacity": null,  
                "Priority": null,  
                "Placement": {  
                    "AvailabilityZone": "",  
                    "Affinity": "",  
                    "GroupName": "",  
                    "PartitionNumber": 0,  
                    "HostId": "",  
                    "Tenancy": "dedicated",  
                    "SpreadDomain": ""  
                }  
            }  
        ]  
    }  
],  
"TargetCapacitySpecification": {  
    "TotalTargetCapacity": 0,  
    "OnDemandTargetCapacity": 0,  
    "SpotTargetCapacity": 0,  
    "DefaultTargetCapacityType": "spot"  
},  
"TerminateInstancesWithExpiration": true,  
"Type": "maintain",  
"ValidFrom": "1970-01-01T00:00:00",  
"ValidUntil": "1970-01-01T00:00:00",  
"ReplaceUnhealthyInstances": true,  
"TagSpecifications": [  
    {  
        "ResourceType": "fleet",  
        "Tags": [  
            {  
                "Key": "",  
                "Value": ""  
            }  
        ]  
    }  
]
```

}

EC2 Fleet JSON configuration file reference

Note

Use lowercase for all parameter values; otherwise, you get an error when Amazon EC2 uses the JSON file to launch the EC2 Fleet.

AllocationStrategy (for SpotOptions)

(Optional) Indicates how to allocate the Spot Instance target capacity across the Spot Instance pools specified by the EC2 Fleet. Valid values are lowest-price, diversified, and capacity-optimized. The default is lowest-price. Specify the allocation strategy that meets your needs. For more information, see [Allocation strategies for Spot Instances \(p. 527\)](#).

InstanceInterruptionBehavior

(Optional) The behavior when a Spot Instance is interrupted. Valid values are hibernate, stop, and terminate. By default, the Spot service terminates Spot Instances when they are interrupted. If the fleet type is maintain, you can specify that the Spot service hibernates or stops Spot Instances when they are interrupted.

InstancePoolsToUseCount

The number of Spot pools across which to allocate your target Spot capacity. Valid only when Spot **AllocationStrategy** is set to lowest-price. EC2 Fleet selects the cheapest Spot pools and evenly allocates your target Spot capacity across the number of Spot pools that you specify.

SingleInstanceType

Indicates that the fleet uses a single instance type to launch all Spot Instances in the fleet.

SingleAvailabilityZone

Indicates that the fleet launches all Spot Instances into a single Availability Zone.

MaxTotalPrice

The maximum amount per hour for Spot Instances that you're willing to pay.

MinTargetCapacity

The minimum target capacity for Spot Instances in the fleet. If the minimum target capacity is not reached, the fleet launches no instances.

AllocationStrategy (for OnDemandOptions)

The order of the launch template overrides to use in fulfilling On-Demand capacity. If you specify lowest-price, EC2 Fleet uses price to determine the order, launching the lowest price first. If you specify prioritized, EC2 Fleet uses the priority that you assigned to each launch template override, launching the highest priority first. If you do not specify a value, EC2 Fleet defaults to lowest-price.

SingleInstanceType

Indicates that the fleet uses a single instance type to launch all On-Demand Instances in the fleet.

SingleAvailabilityZone

Indicates that the fleet launches all On-Demand Instances into a single Availability Zone.

MaxTotalPrice

The maximum amount per hour for On-Demand Instances that you're willing to pay.

MinTargetCapacity

The minimum target capacity for On-Demand Instances in the fleet. If the minimum target capacity is not reached, the fleet launches no instances.

ExcessCapacityTerminationPolicy

(Optional) Indicates whether running instances should be terminated if the total target capacity of the EC2 Fleet is decreased below the current size of the EC2 Fleet. Valid values are no-termination and termination.

LaunchTemplateId

The ID of the launch template to use. You must specify either the launch template ID or launch template name. The launch template must specify an Amazon Machine Image (AMI). For information about creating launch templates, see [Launching an instance from a launch template \(p. 504\)](#).

LaunchTemplateName

The name of the launch template to use. You must specify either the launch template ID or launch template name. The launch template must specify an Amazon Machine Image (AMI). For more information, see [Launching an instance from a launch template \(p. 504\)](#).

Version

The launch template version number, \$Latest, or \$Default. You must specify a value, otherwise the request fails. If the value is \$Latest, Amazon EC2 uses the latest version of the launch template. If the value is \$Default, Amazon EC2 uses the default version of the launch template. For more information, see [Managing launch template versions \(p. 515\)](#).

InstanceType

(Optional) The instance type. If entered, this value overrides the launch template. The instance types must have the minimum hardware specifications that you need (vCPUs, memory, or storage).

MaxPrice

(Optional) The maximum price per unit hour that you are willing to pay for a Spot Instance. If entered, this value overrides the launch template. You can use the default maximum price (the On-Demand price) or specify the maximum price that you are willing to pay. Your Spot Instances are not launched if your maximum price is lower than the Spot price for the instance types that you specified.

SubnetId

(Optional) The ID of the subnet in which to launch the instances. If entered, this value overrides the launch template.

To create a new VPC, go the Amazon VPC console. When you are done, return to the JSON file and enter the new subnet ID.

AvailabilityZone

(Optional) The Availability Zone in which to launch the instances. The default is to let AWS choose the zones for your instances. If you prefer, you can specify specific zones. If entered, this value overrides the launch template.

Specify one or more Availability Zones. If you have more than one subnet in a zone, specify the appropriate subnet. To add subnets, go to the Amazon VPC console. When you are done, return to the JSON file and enter the new subnet ID.

WeightedCapacity

(Optional) The number of units provided by the specified instance type. If entered, this value overrides the launch template.

Priority

The priority for the launch template override. If **AllocationStrategy** is set to prioritized, EC2 Fleet uses priority to determine which launch template override to use first in fulfilling On-Demand

capacity. The highest priority is launched first. Valid values are whole numbers starting at 0. The lower the number, the higher the priority. If no number is set, the override has the lowest priority.

TotalTargetCapacity

The number of instances to launch. You can choose instances or performance characteristics that are important to your application workload, such as vCPUs, memory, or storage. If the request type is `maintain`, you can specify a target capacity of 0 and add capacity later.

OnDemandTargetCapacity

(Optional) The number of On-Demand Instances to launch. This number must be less than the `TotalTargetCapacity`.

SpotTargetCapacity

(Optional) The number of Spot Instances to launch. This number must be less than the `TotalTargetCapacity`.

DefaultTargetCapacityType

If the value for `TotalTargetCapacity` is higher than the combined values for `OnDemandTargetCapacity` and `SpotTargetCapacity`, the difference is launched as the instance purchasing option specified here. Valid values are `on-demand` or `spot`.

TerminateInstancesWithExpiration

(Optional) By default, Amazon EC2 terminates your instances when the EC2 Fleet request expires. The default value is `true`. To keep them running after your request expires, do not enter a value for this parameter.

Type

(Optional) Indicates whether the EC2 Fleet submits a synchronous one-time request for your desired capacity (`instant`), or an asynchronous one-time request for your desired capacity, but with no attempt maintain the capacity or to submit requests in alternative capacity pools if capacity is unavailable (`request`), or submits an asynchronous request for your desired capacity and continues to maintain your desired capacity by replenishing interrupted Spot Instances (`maintain`). Valid values are `instant`, `request`, and `maintain`. The default value is `maintain`. For more information, see [EC2 Fleet request types \(p. 527\)](#).

ValidFrom

(Optional) To create a request that is valid only during a specific time period, enter a start date.

ValidUntil

(Optional) To create a request that is valid only during a specific time period, enter an end date.

ReplaceUnhealthyInstances

(Optional) To replace unhealthy instances in an EC2 Fleet that is configured to `maintain` the fleet, enter `true`. Otherwise, leave this parameter empty.

TagSpecifications

(Optional) The key-value pair for tagging the EC2 Fleet request on creation. The value for `ResourceType` must be `fleet`, otherwise the fleet request fails. To tag instances at launch, specify the tags in the [launch template \(p. 506\)](#). For information about tagging after launch, see [Tagging your resources \(p. 1231\)](#).

[Creating an EC2 Fleet](#)

When you create an EC2 Fleet, you must specify a launch template that includes information about the instances to launch, such as the instance type, Availability Zone, and the maximum price you are willing to pay.

You can create an EC2 Fleet that includes multiple launch specifications that override the launch template. The launch specifications can vary by instance type, Availability Zone, subnet, and maximum price, and can include a different weighted capacity.

When you create an EC2 Fleet, use a JSON file to specify information about the instances to launch. For more information, see [EC2 Fleet JSON configuration file reference \(p. 541\)](#).

EC2 Fleets can only be created using the AWS CLI.

To create an EC2 Fleet (AWS CLI)

- Use the [create-fleet](#) (AWS CLI) command to create an EC2 Fleet.

```
aws ec2 create-fleet \  
  --cli-input-json file://file_name.json
```

For example configuration files, see [EC2 Fleet example configurations \(p. 551\)](#).

The following is example output for a fleet of type `request` or `maintain`.

```
{  
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"  
}
```

The following is example output for a fleet of type `instant` that launched the target capacity.

```
{  
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",  
    "Errors": [],  
    "Instances": [  
        {  
            "LaunchTemplateAndOverrides": {  
                "LaunchTemplateSpecification": {  
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",  
                    "Version": "1"  
                },  
                "Overrides": {  
                    "InstanceType": "c5.large",  
                    "AvailabilityZone": "us-east-1a"  
                }  
            },  
            "Lifecycle": "on-demand",  
            "InstanceIds": [  
                "i-1234567890abcdef0",  
                "i-9876543210abcdef9"  
            ],  
            "InstanceType": "c5.large",  
            "Platform": null  
        },  
        {  
            "LaunchTemplateAndOverrides": {  
                "LaunchTemplateSpecification": {  
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",  
                    "Version": "1"  
                },  
                "Overrides": {  
                    "InstanceType": "c4.large",  
                    "AvailabilityZone": "us-east-1a"  
                }  
            },  
            "Lifecycle": "instant",  
            "InstanceIds": [  
                "i-1234567890abcdef0",  
                "i-9876543210abcdef9"  
            ]  
        }  
    ]  
}
```

```
"Lifecycle": "on-demand",
"InstanceIds": [
    "i-5678901234abcdef0",
    "i-5432109876abcdef9"
],
"InstanceType": "c4.large",
"Platform": null
},
]
}
```

The following is example output for a fleet of type `instant` that launched part of the target capacity with errors for instances that were not launched.

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
    "Errors": [
        {
            "LaunchTemplateAndOverrides": {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
                    "Version": "1"
                },
                "Overrides": {
                    "InstanceType": "c4.xlarge",
                    "AvailabilityZone": "us-east-1a"
                }
            },
            "Lifecycle": "on-demand",
            "ErrorCode": "InsufficientInstanceCapacity",
            "ErrorMessage": "",
            "InstanceType": "c4.xlarge",
            "Platform": null
        },
    ],
    "Instances": [
        {
            "LaunchTemplateAndOverrides": {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
                    "Version": "1"
                },
                "Overrides": {
                    "InstanceType": "c5.large",
                    "AvailabilityZone": "us-east-1a"
                }
            },
            "Lifecycle": "on-demand",
            "InstanceIds": [
                "i-1234567890abcdef0",
                "i-9876543210abcdef9"
            ],
            "InstanceType": "c5.large",
            "Platform": null
        },
    ]
}
```

The following is example output for a fleet of type `instant` that launched no instances.

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
    "Errors": [
```

```
{  
    "LaunchTemplateAndOverrides": {  
        "LaunchTemplateSpecification": {  
            "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",  
            "Version": "1"  
        },  
        "Overrides": {  
            "InstanceType": "c4.xlarge",  
            "AvailabilityZone": "us-east-1a",  
        }  
    },  
    "Lifecycle": "on-demand",  
    "ErrorCode": "InsufficientCapacity",  
    "ErrorMessage": "",  
    "InstanceType": "c4.xlarge",  
    "Platform": null  
},  
{  
    "LaunchTemplateAndOverrides": {  
        "LaunchTemplateSpecification": {  
            "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",  
            "Version": "1"  
        },  
        "Overrides": {  
            "InstanceType": "c5.large",  
            "AvailabilityZone": "us-east-1a",  
        }  
    },  
    "Lifecycle": "on-demand",  
    "ErrorCode": "InsufficientCapacity",  
    "ErrorMessage": "",  
    "InstanceType": "c5.large",  
    "Platform": null  
},  
],  
"Instances": []  
}
```

Tagging an EC2 Fleet

To help categorize and manage your EC2 Fleet requests, you can tag them with custom metadata. You can assign a tag to an EC2 Fleet request when you create it, or afterward.

When you tag a fleet request, the instances and volumes that are launched by the fleet are not automatically tagged. You need to explicitly tag the instances and volumes launched by the fleet. You can choose to assign tags to only the fleet request, or to only the instances launched by the fleet, or to only the volumes attached to the instances launched by the fleet, or to all three.

Note

For instant fleet types, you can tag volumes that are attached to On-Demand Instances and Spot Instances. For request or maintain fleet types, you can only tag volumes that are attached to On-Demand Instances.

For more information about how tags work, see [Tagging your Amazon EC2 resources \(p. 1229\)](#).

Prerequisite

Grant the IAM user the permission to tag resources. For more information, see [Example: Tagging resources \(p. 958\)](#).

To grant an IAM user the permission to tag resources

Create a IAM policy that includes the following:

- The `ec2:CreateTags` action. This grants the IAM user permission to create tags.
- The `ec2:CreateFleet` action. This grants the IAM user permission to create an EC2 Fleet request.
- For `Resource`, we recommend that you specify `"*"`. This allows users to tag all resource types.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "TagEC2FleetRequest",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags",  
                "ec2:CreateFleet"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Important

We currently do not support resource-level permissions for the `create-fleet` resource. If you specify `create-fleet` as a resource, you will get an unauthorized exception when you try to tag the fleet. The following example illustrates how *not* to set the policy.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "ec2:CreateTags",  
        "ec2:CreateFleet"  
    ],  
    "Resource": "arn:aws:ec2:us-east-1:111122223333:create-fleet/*"  
}
```

To tag a new EC2 Fleet request

To tag an EC2 Fleet request when you create it, specify the key-value pair in the [JSON file \(p. 539\)](#) used to create the fleet. The value for `ResourceType` must be `fleet`. If you specify another value, the fleet request fails.

To tag instances and volumes launched by an EC2 Fleet

To tag instances and volumes when they are launched by the fleet, specify the tags in the [launch template \(p. 506\)](#) that is referenced in the EC2 Fleet request.

Note

You can't tag volumes attached to Spot Instances that are launched by a `request` or `maintain` fleet type.

To tag an existing EC2 Fleet request, instance, and volume (AWS CLI)

Use the `create-tags` command to tag existing resources.

```
aws ec2 create-tags \  
  --resources fleet-12a34b55-67cd-8ef9-  
  ba9b-9208dEXAMPLE i-1234567890abcdef0 vol-1234567890EXAMPLE \  
  --tags Key=purpose,Value=test
```

Monitoring your EC2 Fleet

The EC2 Fleet launches On-Demand Instances when there is available capacity, and launches Spot Instances when your maximum price exceeds the Spot price and capacity is available. The On-Demand

Instances run until you terminate them, and the Spot Instances run until they are interrupted or you terminate them.

The returned list of running instances is refreshed periodically and might be out of date.

To monitor your EC2 Fleet (AWS CLI)

Use the [describe-fleets](#) command to describe your EC2 Fleets.

```
aws ec2 describe-fleets
```

The following is example output.

```
{
    "Fleets": [
        {
            "Type": "maintain",
            "FulfilledCapacity": 2.0,
            "LaunchTemplateConfigs": [
                {
                    "LaunchTemplateSpecification": {
                        "Version": "2",
                        "LaunchTemplateId": "lt-07b3bc7625cdab851"
                    }
                }
            ],
            "TerminateInstancesWithExpiration": false,
            "TargetCapacitySpecification": {
                "OnDemandTargetCapacity": 0,
                "SpotTargetCapacity": 2,
                "TotalTargetCapacity": 2,
                "DefaultTargetCapacityType": "spot"
            },
            "FulfilledOnDemandCapacity": 0.0,
            "ActivityStatus": "fulfilled",
            "FleetId": "fleet-76e13e99-01ef-4bd6-ba9b-9208de883e7f",
            "ReplaceUnhealthyInstances": false,
            "SpotOptions": {
                "InstanceInterruptionBehavior": "terminate",
                "InstancePoolsToUseCount": 1,
                "AllocationStrategy": "lowest-price"
            },
            "FleetState": "active",
            "ExcessCapacityTerminationPolicy": "termination",
            "CreateTime": "2018-04-10T16:46:03.000Z"
        }
    ]
}
```

Use the [describe-fleet-instances](#) command to describe the instances for the specified EC2 Fleet.

```
aws ec2 describe-fleet-instances \
--fleet-id fleet-73fb2ce-aa30-494c-8788-1cee4EXAMPLE
```

```
{
    "ActiveInstances": [
        {
            "InstanceId": "i-09cd595998cb3765e",
            "InstanceHealth": "healthy",
            "InstanceType": "m4.large",
            "SpotInstanceRequestId": "sir-86k84j6p"
        },
    ]
```

```
{  
    "InstanceId": "i-09cf95167ca219f17",  
    "InstanceHealth": "healthy",  
    "InstanceType": "m4.large",  
    "SpotInstanceRequestId": "sir-dvxi7fsm"  
}  
,  
]  
,  
"FleetId": "fleet-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE"  
}
```

Use the [describe-fleet-history](#) command to describe the history for the specified EC2 Fleet for the specified time.

```
aws ec2 describe-fleet-history --fleet-request-id fleet-73fbcd2ce-  
aa30-494c-8788-1cee4EXAMPLE --start-time 2018-04-10T00:00:00Z
```

```
{  
    "HistoryRecords": [],  
    "FleetId": "fleet-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE",  
    "LastEvaluatedTime": "1970-01-01T00:00:00.000Z",  
    "StartTime": "2018-04-09T23:53:20.000Z"  
}
```

Modifying an EC2 Fleet

You can modify an EC2 Fleet that is in the submitted or active state. When you modify a fleet, it enters the modifying state.

You can only modify an EC2 Fleet that is of type `maintain`. You cannot modify an EC2 Fleet of type `request` or `instant`.

You can modify the following parameters of an EC2 Fleet:

- `target-capacity-specification` – Increase or decrease the target capacity for `TotalTargetCapacity`, `OnDemandTargetCapacity`, and `SpotTargetCapacity`.
- `excess-capacity-termination-policy` – Whether running instances should be terminated if the total target capacity of the EC2 Fleet is decreased below the current size of the fleet. Valid values are `no-termination` and `termination`.

When you increase the target capacity, the EC2 Fleet launches the additional instances according to the instance purchasing option specified for `DefaultTargetCapacityType`, which are either On-Demand Instances or Spot Instances.

If the `DefaultTargetCapacityType` is `spot`, the EC2 Fleet launches the additional Spot Instances according to its allocation strategy. If the allocation strategy is `lowest-price`, the fleet launches the instances from the lowest-priced Spot Instance pool in the request. If the allocation strategy is `diversified`, the fleet distributes the instances across the pools in the request.

When you decrease the target capacity, the EC2 Fleet deletes any open requests that exceed the new target capacity. You can request that the fleet terminate instances until the size of the fleet reaches the new target capacity. If the allocation strategy is `lowest-price`, the fleet terminates the instances with the highest price per unit. If the allocation strategy is `diversified`, the fleet terminates instances across the pools. Alternatively, you can request that EC2 Fleet keep the fleet at its current size, but not replace any Spot Instances that are interrupted or any instances that you terminate manually.

When an EC2 Fleet terminates a Spot Instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

To modify an EC2 Fleet (AWS CLI)

Use the [modify-fleet](#) command to update the target capacity of the specified EC2 Fleet.

```
aws ec2 modify-fleet \
--fleet-id fleet-73fb2ce-aa30-494c-8788-1cee4EXAMPLE \
--target-capacity-specification TotalTargetCapacity=20
```

If you are decreasing the target capacity but want to keep the fleet at its current size, you can modify the previous command as follows.

```
aws ec2 modify-fleet \
--fleet-id fleet-73fb2ce-aa30-494c-8788-1cee4EXAMPLE \
--target-capacity-specification TotalTargetCapacity=10 \
--excess-capacity-termination-policy no-termination
```

Deleting an EC2 Fleet

If you no longer require an EC2 Fleet, you can delete it. After you delete a fleet, it launches no new instances.

You must specify whether the EC2 Fleet must terminate its instances. If you specify that the instances must be terminated when the fleet is deleted, it enters the `deleted_terminating` state. Otherwise, it enters the `deleted_running` state, and the instances continue to run until they are interrupted or you terminate them manually.

You can only delete fleets of type `request` and `maintain`. You cannot delete an `instant` EC2 Fleet.

To delete an EC2 Fleet (AWS CLI)

Use the [delete-fleets](#) command and the `--terminate-instances` parameter to delete the specified EC2 Fleet and terminate the instances.

```
aws ec2 delete-fleets \
--fleet-ids fleet-73fb2ce-aa30-494c-8788-1cee4EXAMPLE \
--terminate-instances
```

The following is example output.

```
{
    "UnsuccessfulFleetDeletions": [],
    "SuccessfulFleetDeletions": [
        {
            "CurrentFleetState": "deleted_terminating",
            "PreviousFleetState": "active",
            "FleetId": "fleet-73fb2ce-aa30-494c-8788-1cee4EXAMPLE"
        }
    ]
}
```

You can modify the previous command using the `--no-terminate-instances` parameter to delete the specified EC2 Fleet without terminating the instances.

```
aws ec2 delete-fleets \
--fleet-ids fleet-73fb2ce-aa30-494c-8788-1cee4EXAMPLE \
--no-terminate-instances
```

The following is example output.

```
{
```

```
"UnsuccessfulFleetDeletions": [],
"SuccessfulFleetDeletions": [
    {
        "CurrentFleetState": "deleted_running",
        "PreviousFleetState": "active",
        "FleetId": "fleet-4b8aaae8-dfb5-436d-a4c6-3da4c6b7dcEXAMPLE"
    }
]
```

Reasons for a failed delete

If an EC2 Fleet fails to delete, `UnsuccessfulFleetDeletions` returns the ID of the EC2 Fleet, an error code, and an error message. The error codes are `fleetIdDoesNotExist`, `fleetIdMalformed`, `fleetNotInDeletableState`, and `unexpectedError`.

EC2 Fleet example configurations

The following examples show launch configurations that you can use with the `create-fleet` command to create an EC2 Fleet. For more information about the `create-fleet` parameters, see the [EC2 Fleet JSON configuration file reference \(p. 541\)](#).

Examples

- [Example 1: Launch Spot Instances as the default purchasing option \(p. 551\)](#)
- [Example 2: Launch On-Demand Instances as the default purchasing option \(p. 552\)](#)
- [Example 3: Launch On-Demand Instances as the primary capacity \(p. 552\)](#)
- [Example 4: Launch Spot Instances using the lowest-price allocation strategy \(p. 552\)](#)
- [Example 5: Launch On-Demand Instances using Capacity Reservations and the prioritized allocation strategy \(p. 553\)](#)
- [Example 6: Launch On-Demand Instances using Capacity Reservations and the prioritized allocation strategy when the total target capacity is more than the number of unused Capacity Reservations \(p. 555\)](#)
- [Example 7: Launch On-Demand Instances using Capacity Reservations and the lowest-price allocation strategy \(p. 557\)](#)
- [Example 8: Launch On-Demand Instances using Capacity Reservations and the lowest-price allocation strategy when the total target capacity is more than the number of unused Capacity Reservations \(p. 559\)](#)

Example 1: Launch Spot Instances as the default purchasing option

The following example specifies the minimum parameters required in an EC2 Fleet: a launch template, target capacity, and default purchasing option. The launch template is identified by its launch template ID and version number. The target capacity for the fleet is 2 instances, and the default purchasing option is `spot`, which results in the fleet launching 2 Spot Instances.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "spot"
    }
}
```

```
}
```

Example 2: Launch On-Demand Instances as the default purchasing option

The following example specifies the minimum parameters required in an EC2 Fleet: a launch template, target capacity, and default purchasing option. The launch template is identified by its launch template ID and version number. The target capacity for the fleet is 2 instances, and the default purchasing option is on-demand, which results in the fleet launching 2 On-Demand Instances.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "on-demand"
    }
}
```

Example 3: Launch On-Demand Instances as the primary capacity

The following example specifies the total target capacity of 2 instances for the fleet, and a target capacity of 1 On-Demand Instance. The default purchasing option is spot. The fleet launches 1 On-Demand Instance as specified, but needs to launch one more instance to fulfill the total target capacity. The purchasing option for the difference is calculated as `TotalTargetCapacity - OnDemandTargetCapacity = DefaultTargetCapacityType`, which results in the fleet launching 1 Spot Instance.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "OnDemandTargetCapacity": 1,
        "DefaultTargetCapacityType": "spot"
    }
}
```

Example 4: Launch Spot Instances using the lowest-price allocation strategy

If the allocation strategy for Spot Instances is not specified, the default allocation strategy, which is `lowest-price`, is used. The following example uses the `lowest-price` allocation strategy. The three launch specifications, which override the launch template, have different instance types but the same weighted capacity and subnet. The total target capacity is 2 instances and the default purchasing option is `spot`. The EC2 Fleet launches 2 Spot Instances using the instance type of the launch specification with the lowest price.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
        }
    ],
    "Overrides": [
        {
            "InstanceType": "c4.large",
            "WeightedCapacity": 1,
            "SubnetId": "subnet-a4f6c5d3"
        },
        {
            "InstanceType": "c3.large",
            "WeightedCapacity": 1,
            "SubnetId": "subnet-a4f6c5d3"
        },
        {
            "InstanceType": "c5.large",
            "WeightedCapacity": 1,
            "SubnetId": "subnet-a4f6c5d3"
        }
    ]
},
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "spot"
    }
}
```

[Example 5: Launch On-Demand Instances using Capacity Reservations and the prioritized allocation strategy](#)

You can configure a fleet to use On-Demand Capacity Reservations first when launching On-Demand Instances by setting the usage strategy for Capacity Reservations to `use-capacity-reservations-first`. And if multiple instance pools have unused Capacity Reservations, the chosen On-Demand allocation strategy is applied. In this example, the On-Demand allocation strategy is prioritized.

In this example, there are 15 available unused Capacity Reservations. This is more than the fleet's target On-Demand capacity of 12 On-Demand Instances.

The account has the following 15 unused Capacity Reservations in 3 different pools. The number of Capacity Reservations in each pool is indicated by `AvailableInstanceCount`.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c4.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c3.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5
}
```

```
        "AvailableInstanceCount": 5,
        "InstanceMatchCriteria": "open",
        "State": "active"
    }
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "c5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

The following fleet configuration shows only the pertinent configurations for this example. The On-Demand allocation strategy is prioritized, and the usage strategy for Capacity Reservations is `use-capacity-reservations-first`. The total target capacity is 12, and the default target capacity type is on-demand.

Note

The fleet type must be `instant`. Capacity Reservations are not supported for other fleet types.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-1234567890abcdefg",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "c4.large",
                    "AvailabilityZone": "us-east-1a",
                    "WeightedCapacity": 1,
                    "Priority": 1.0
                },
                {
                    "InstanceType": "c3.large",
                    "AvailabilityZone": "us-east-1a",
                    "WeightedCapacity": 1,
                    "Priority": 2.0
                },
                {
                    "InstanceType": "c5.large",
                    "AvailabilityZone": "us-east-1a",
                    "WeightedCapacity": 1,
                    "Priority": 3.0
                }
            ]
        },
        {
            "TargetCapacitySpecification": {
                "TotalTargetCapacity": 12,
                "DefaultTargetCapacityType": "on-demand"
            },
            "OnDemandOptions": {
                "AllocationStrategy": "prioritized",
                "CapacityReservationOptions": {
                    "UsageStrategy": "use-capacity-reservations-first"
                }
            },
            "Type": "instant",
        }
    ]
}
```

After you create the instant fleet using the preceding configuration, the following 12 instances are launched to meet the target capacity:

- 5 c4.large On-Demand Instances in us-east-1a – c4.large in us-east-1a is prioritized first, and there are 5 available unused c4.large Capacity Reservations
- 5 c3.large On-Demand Instances in us-east-1a – c3.large in us-east-1a is prioritized second, and there are 5 available unused c3.large Capacity Reservations
- 2 c5.large On-Demand Instances in us-east-1a – c5.large in us-east-1a is prioritized third, and there are 5 available unused c5.large Capacity Reservations of which only 2 are needed to meet the target capacity

After the fleet is launched, you can run [describe-capacity-reservations](#) to see how many unused Capacity Reservations are remaining. In this example, you should see the following response, which shows that all of the c4.large and c3.large Capacity Reservations were used, with 3 c5.large Capacity Reservations remaining unused.

```
{  
    "CapacityReservationId": "cr-111",  
    "InstanceType": "c4.large",  
    "AvailableInstanceCount": 0  
}  
  
{  
    "CapacityReservationId": "cr-222",  
    "InstanceType": "c3.large",  
    "AvailableInstanceCount": 0  
}  
  
{  
    "CapacityReservationId": "cr-333",  
    "InstanceType": "c5.large",  
    "AvailableInstanceCount": 3  
}
```

Example 6: Launch On-Demand Instances using Capacity Reservations and the prioritized allocation strategy when the total target capacity is more than the number of unused Capacity Reservations

You can configure a fleet to use On-Demand Capacity Reservations first when launching On-Demand Instances by setting the usage strategy for Capacity Reservations to `use-capacity-reservations-first`. And if the number of unused Capacity Reservations is less than the On-Demand target capacity, the remaining On-Demand target capacity is launched according to the chosen On-Demand allocation strategy. In this example, the On-Demand allocation strategy is prioritized.

In this example, there are 15 available unused Capacity Reservations. This is less than the fleet's On-Demand target capacity of 16 On-Demand Instances.

The account has the following 15 unused Capacity Reservations in 3 different pools. The number of Capacity Reservations in each pool is indicated by AvailableInstanceCount.

```
{  
    "CapacityReservationId": "cr-111",  
    "InstanceType": "c4.large",  
    "InstancePlatform": "Linux/UNIX",  
    "AvailabilityZone": "us-east-1a",  
    "AvailableInstanceCount": 5,  
    "InstanceMatchCriteria": "open",  
    "State": "active"  
}
```

```
{  
    "CapacityReservationId": "cr-111",  
    "InstanceType": "c3.large",  
    "InstancePlatform": "Linux/UNIX",  
    "AvailabilityZone": "us-east-1a",  
    "AvailableInstanceCount": 5,  
    "InstanceMatchCriteria": "open",  
    "State": "active"  
}  
  
{  
    "CapacityReservationId": "cr-111",  
    "InstanceType": "c5.large",  
    "InstancePlatform": "Linux/UNIX",  
    "AvailabilityZone": "us-east-1a",  
    "AvailableInstanceCount": 5,  
    "InstanceMatchCriteria": "open",  
    "State": "active"  
}
```

The following fleet configuration shows only the pertinent configurations for this example. The On-Demand allocation strategy is prioritized, and the usage strategy for Capacity Reservations is use-capacity-reservations-first. The total target capacity is 16, and the default target capacity type is on-demand.

Note

The fleet type must be instant. Capacity Reservations are not supported for other fleet types.

```
{  
    "LaunchTemplateConfigs": [  
        {  
            "LaunchTemplateSpecification": {  
                "LaunchTemplateId": "lt-0e8c754449b27161c",  
                "Version": "1"  
            },  
            "Overrides": [  
                {  
                    "InstanceType": "c4.large",  
                    "AvailabilityZone": "us-east-1a",  
                    "WeightedCapacity": 1,  
                    "Priority": 1.0  
                },  
                {  
                    "InstanceType": "c3.large",  
                    "AvailabilityZone": "us-east-1a",  
                    "WeightedCapacity": 1,  
                    "Priority": 2.0  
                },  
                {  
                    "InstanceType": "c5.large",  
                    "AvailabilityZone": "us-east-1a",  
                    "WeightedCapacity": 1,  
                    "Priority": 3.0  
                }  
            ]  
        }  
    ],  
    "TargetCapacitySpecification": {  
        "TotalTargetCapacity": 16,  
        "DefaultTargetCapacityType": "on-demand"  
    },  
    "OnDemandOptions": {
```

```
"AllocationStrategy": "prioritized"
"CapacityReservationOptions": [
    "UsageStrategy": "use-capacity-reservations-first"
]
},
"Type": "instant",
}
```

After you create the `instant` fleet using the preceding configuration, the following 16 instances are launched to meet the target capacity:

- 6 c4.large On-Demand Instances in us-east-1a – c4.large in us-east-1a is prioritized first, and there are 5 available unused c4.large Capacity Reservations. The Capacity Reservations are used first to launch 5 On-Demand Instances plus an additional On-Demand Instance is launched according to the On-Demand allocation strategy, which is `prioritized` in this example.
- 5 c3.large On-Demand Instances in us-east-1a – c3.large in us-east-1a is prioritized second, and there are 5 available unused c3.large Capacity Reservations
- 5 c5.large On-Demand Instances in us-east-1a – c5.large in us-east-1a is prioritized third, and there are 5 available unused c5.large Capacity Reservations

After the fleet is launched, you can run [describe-capacity-reservations](#) to see how many unused Capacity Reservations are remaining. In this example, you should see the following response, which shows that all of the Capacity Reservations in all of the pools were used.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c4.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c3.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "c5.large",
    "AvailableInstanceCount": 0
}
```

Example 7: Launch On-Demand Instances using Capacity Reservations and the `lowest-price` allocation strategy

You can configure a fleet to use On-Demand Capacity Reservations first when launching On-Demand Instances by setting the usage strategy for Capacity Reservations to `use-capacity-reservations-first`. And if multiple instance pools have unused Capacity Reservations, the chosen On-Demand allocation strategy is applied. In this example, the On-Demand allocation strategy is `lowest-price`.

In this example, there are 15 available unused Capacity Reservations. This is more than the fleet's target On-Demand capacity of 12 On-Demand Instances.

The account has the following 15 unused Capacity Reservations in 3 different pools. The number of Capacity Reservations in each pool is indicated by `AvailableInstanceCount`.

```
{
    "CapacityReservationId": "cr-111",
```

```
"InstanceType": "m5.large",
"InstancePlatform": "Linux/UNIX",
"AvailabilityZone": "us-east-1a",
"AvailableInstanceCount": 5,
"InstanceMatchCriteria": "open",
"State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

The following fleet configuration shows only the pertinent configurations for this example. The On-Demand allocation strategy is `lowest-price`, and the usage strategy for Capacity Reservations is `use-capacity-reservations-first`. The total target capacity is 12, and the default target capacity type is `on-demand`.

In this example, the On-Demand Instance price is:

- m5.large – \$0.096 per hour
- m4.xlarge – \$0.20 per hour
- m4.2xlarge – \$0.40 per hour

Note

The fleet type must be `instant`. Capacity Reservations are not supported for other fleet types.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
        },
        {
            "Overrides": [
                {
                    "InstanceType": "m5.large",
                    "AvailabilityZone": "us-east-1a",
                    "WeightedCapacity": 1
                },
                {
                    "InstanceType": "m4.xlarge",
                    "AvailabilityZone": "us-east-1a",
                    "WeightedCapacity": 1
                },
                {
                    "InstanceType": "m4.2xlarge",
                    "AvailabilityZone": "us-east-1a",
                    "WeightedCapacity": 1
                }
            ]
        }
    ]
}
```

```
        "AvailabilityZone": "us-east-1a",
        "WeightedCapacity": 1
    }
]
}
],
"TargetCapacitySpecification": {
    "TotalTargetCapacity": 12,
    "DefaultTargetCapacityType": "on-demand"
},
"OnDemandOptions": {
    "AllocationStrategy": "lowest-price"
    "CapacityReservationOptions": {
        "UsageStrategy": "use-capacity-reservations-first"
    }
},
"Type": "instant",
}
```

After you create the `instant` fleet using the preceding configuration, the following 12 instances are launched to meet the target capacity:

- 5 m5.large On-Demand Instances in us-east-1a – m5.large in us-east-1a is the lowest price, and there are 5 available unused m5.large Capacity Reservations
- 5 m4.xlarge On-Demand Instances in us-east-1a – m4.xlarge in us-east-1a is the next lowest price, and there are 5 available unused m4.xlarge Capacity Reservations
- 2 m4.2xlarge On-Demand Instances in us-east-1a – m4.2xlarge in us-east-1a is the third lowest price, and there are 5 available unused m4.2xlarge Capacity Reservations of which only 2 are needed to meet the target capacity

After the fleet is launched, you can run [describe-capacity-reservations](#) to see how many unused Capacity Reservations are remaining. In this example, you should see the following response, which shows that all of the m5.large and m4.xlarge Capacity Reservations were used, with 3 m4.2xlarge Capacity Reservations remaining unused.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "AvailableInstanceCount": 3
}
```

Example 8: Launch On-Demand Instances using Capacity Reservations and the `lowest-price` allocation strategy when the total target capacity is more than the number of unused Capacity Reservations

You can configure a fleet to use On-Demand Capacity Reservations first when launching On-Demand Instances by setting the usage strategy for Capacity Reservations to `use-capacity-reservations-`

first. And if the number of unused Capacity Reservations is less than the On-Demand target capacity, the remaining On-Demand target capacity is launched according to the chosen On-Demand allocation strategy. In this example, the On-Demand allocation strategy is `lowest-price`.

In this example, there are 15 available unused Capacity Reservations. This is less than the fleet's On-Demand target capacity of 16 On-Demand Instances.

The account has the following 15 unused Capacity Reservations in 3 different pools. The number of Capacity Reservations in each pool is indicated by `AvailableInstanceCount`.

```
{  
    "CapacityReservationId": "cr-111",  
    "InstanceType": "m5.large",  
    "InstancePlatform": "Linux/UNIX",  
    "AvailabilityZone": "us-east-1a",  
    "AvailableInstanceCount": 5,  
    "InstanceMatchCriteria": "open",  
    "State": "active"  
}  
  
{  
    "CapacityReservationId": "cr-222",  
    "InstanceType": "m4.xlarge",  
    "InstancePlatform": "Linux/UNIX",  
    "AvailabilityZone": "us-east-1a",  
    "AvailableInstanceCount": 5,  
    "InstanceMatchCriteria": "open",  
    "State": "active"  
}  
  
{  
    "CapacityReservationId": "cr-333",  
    "InstanceType": "m4.2xlarge",  
    "InstancePlatform": "Linux/UNIX",  
    "AvailabilityZone": "us-east-1a",  
    "AvailableInstanceCount": 5,  
    "InstanceMatchCriteria": "open",  
    "State": "active"  
}
```

The following fleet configuration shows only the pertinent configurations for this example. The On-Demand allocation strategy is `lowest-price`, and the usage strategy for Capacity Reservations is `use-capacity-reservations-first`. The total target capacity is 16, and the default target capacity type is `on-demand`.

In this example, the On-Demand Instance price is:

- m5.large – \$0.096 per hour
- m4.xlarge – \$0.20 per hour
- m4.2xlarge – \$0.40 per hour

Note

The fleet type must be `instant`. Capacity Reservations are not supported for other fleet types.

```
{  
    "LaunchTemplateConfigs": [  
        {  
            "LaunchTemplateSpecification": {  
                "LaunchTemplateId": "lt-0e8c754449b27161c",  
                "Version": "1"  
            }  
        }  
    ]  
}
```

```

        }
        "Overrides": [
            {
                "InstanceType": "m5.large",
                "AvailabilityZone": "us-east-1a",
                "WeightedCapacity": 1
            },
            {
                "InstanceType": "m4.xlarge",
                "AvailabilityZone": "us-east-1a",
                "WeightedCapacity": 1
            },
            {
                "InstanceType": "m4.2xlarge",
                "AvailabilityZone": "us-east-1a",
                "WeightedCapacity": 1
            }
        ]
    },
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 16,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price"
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        }
    },
    "Type": "instant",
}

```

After you create the `instant` fleet using the preceding configuration, the following 16 instances are launched to meet the target capacity:

- 6 `m5.large` On-Demand Instances in `us-east-1a` – `m5.large` in `us-east-1a` is the lowest price, and there are 5 available unused `m5.large` Capacity Reservations. The Capacity Reservations are used first to launch 5 On-Demand Instances plus an additional On-Demand Instance is launched according to the On-Demand allocation strategy, which is `lowest-price` in this example.
- 5 `m4.xlarge` On-Demand Instances in `us-east-1a` – `m4.xlarge` in `us-east-1a` is the next lowest price, and there are 5 available unused `m4.xlarge` Capacity Reservations
- 5 `m4.2xlarge` On-Demand Instances in `us-east-1a` – `m4.2xlarge` in `us-east-1a` is the third lowest price, and there are 5 available unused `m4.2xlarge` Capacity Reservations

After the fleet is launched, you can run [describe-capacity-reservations](#) to see how many unused Capacity Reservations are remaining. In this example, you should see the following response, which shows that all of the Capacity Reservations in all of the pools were used.

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "AvailableInstanceCount": 0
}
```

```
{  
    "CapacityReservationId": "cr-333",  
    "InstanceType": "m4.2xlarge",  
    "AvailableInstanceCount": 0  
}
```

Connect to your Linux instance

Connect to the Linux instances that you launched and transfer files between your local computer and your instance.

To connect to a Windows instance, see [Connecting to Your Windows Instance](#) in the *Amazon EC2 User Guide for Windows Instances*.

Connection options

The operating system of your local computer determines the options that you have to connect from your local computer to your Linux instance.

If your local computer operating system is Linux or macOS X

- [SSH client \(p. 566\)](#)
- [EC2 Instance Connect \(p. 569\)](#)
- [AWS Systems Manager Session Manager](#)

If your local computer operating system is Windows

- [PuTTY \(p. 578\)](#)
- [SSH client \(p. 566\)](#)
- [AWS Systems Manager Session Manager](#)
- [Windows Subsystem for Linux \(p. 583\)](#)

General prerequisites for connecting to your instance

Before you connect to your Linux instance, verify the following general prerequisites:

- [Get information about your instance \(p. 563\)](#)
- [Enable inbound traffic to your instance \(p. 565\)](#)
- [Locate the private key \(p. 565\)](#)
- [\(Optional\) Get the instance fingerprint \(p. 565\)](#)

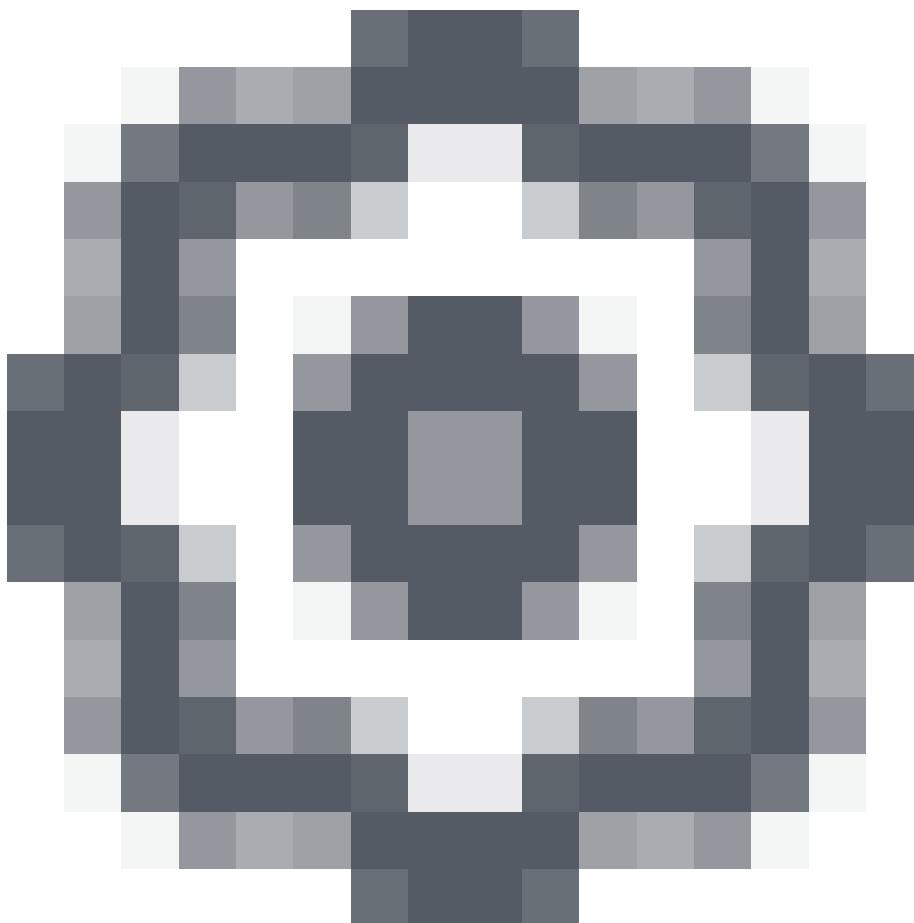
Get information about your instance

- **Get the ID of the instance.**

You can get the ID of your instance using the Amazon EC2 console (from the **Instance ID** column). If you prefer, you can use the [describe-instances](#) (AWS CLI) or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command.

- **Get the public DNS name of the instance.**

You can get the public DNS for your instance using the Amazon EC2 console. Check the **Public DNS (IPv4)** column. If this column is hidden, choose the settings icon (



) in the top-right

corner of the screen and select **Public DNS (IPv4)**. If you prefer, you can use the [describe-instances](#) (AWS CLI) or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command.

- **(IPv6 only) Get the IPv6 address of the instance.**

If you've assigned an IPv6 address to your instance, you can optionally connect to the instance using its IPv6 address instead of a public IPv4 address or public IPv4 DNS hostname. Your local computer must have an IPv6 address and must be configured to use IPv6. You can get the IPv6 address of your instance using the Amazon EC2 console. Check the **IPv6 IPs** field. If you prefer, you can use the [describe-instances](#) (AWS CLI) or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command. For more information about IPv6, see [IPv6 addresses \(p. 761\)](#).

- **Get the user name for your instance.**

You can connect to your instance using the user name for your user account or the default user name for the AMI that you used to launch your instance.

- **Get the user name for your user account.**

For more information about how to create a user account, see [Managing user accounts on your Amazon Linux instance \(p. 620\)](#).

- **Get the default user name for the AMI that you used to launch your instance:**

- For Amazon Linux 2 or the Amazon Linux AMI, the user name is `ec2-user`.
- For a CentOS AMI, the user name is `centos`.
- For a Debian AMI, the user name is `admin`.
- For a Fedora AMI, the user name is `ec2-user` or `fedora`.
- For a RHEL AMI, the user name is `ec2-user` or `root`.

- For a SUSE AMI, the user name is `ec2-user` or `root`.
- For an Ubuntu AMI, the user name is `ubuntu`.
- Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.

Enable inbound traffic to your instance

- **Enable inbound SSH traffic from your IP address to your instance.**

Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. The default security group for the VPC does not allow incoming SSH traffic by default. The security group created by the launch instance wizard enables SSH traffic by default. For more information, see [Authorizing inbound traffic for your Linux instances \(p. 983\)](#).

Locate the private key

- **Locate the private key**

Get the fully-qualified path to the location on your computer of the `.pem` file for the key pair that you specified when you launched the instance. For more information about how you created your key pair, see [Creating a Key Pair Using Amazon EC2](#).

- **Set the permissions of your private key**

If you will use an SSH client on a macOS or Linux computer to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.

```
chmod 400 my-key-pair.pem
```

If you do not set these permissions, then you cannot connect to your instance using this key pair. For more information, see [Error: Unprotected private key file \(p. 1252\)](#).

(Optional) Get the instance fingerprint

To protect yourself from man-in-the-middle attacks, you can verify the RSA key fingerprint when you connect to your instance. Verifying the fingerprint is useful if you've launched your instance from a public AMI from a third party.

First you get the instance fingerprint. Then, when you connect to the instance, you are prompted to verify the fingerprint. You can compare the fingerprint you obtained with the fingerprint displayed for verification. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, you can confidently connect to your instance.

Prerequisites for getting the instance fingerprint:

- To get the instance fingerprint, you must use the AWS CLI. For information about installing the AWS CLI, see [Installing the AWS Command Line Interface](#) in the [AWS Command Line Interface User Guide](#).
- The instance must not be in the pending state. The fingerprint is available only after the first boot of the instance is complete.

To get the instance fingerprint

1. On your local computer (not on the instance), use the `get-console-output` (AWS CLI) command as follows to obtain the fingerprint:

```
aws ec2 get-console-output --instance-id instance_id --output text
```

2. Here is an example of what you should look for in the output. The exact output can vary by the operating system, AMI version, and whether you had AWS create the key.

```
ec2: #####  
ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----  
ec2: 1024 SHA256:7HItIgTONZ/b0CH9c5Dq1ijggQ6kFn86uQhQ5E/F9pU root@ip-10-0-2-182 (DSA)  
ec2: 256 SHA256:14UB/neBad9tvkgJf1QZWxheQmR59WgrgzEimCG6kZY root@ip-10-0-2-182 (ECDSA)  
ec2: 256 SHA256:kpEa+rw/Uq3zxaYZN8KT501iBtJOIdHG52dFi66EEfQ no comment (ED25519)  
ec2: 2048 SHA256:L816pepcA7iqW/jBecQjVZC1UrKY+o2cHLIOiHerbVc root@ip-10-0-2-182 (RSA)  
ec2: -----END SSH HOST KEY FINGERPRINTS-----  
ec2: #####
```

Connecting to your Linux instance using SSH

After you launch your instance, you can connect to it and use it the way that you'd use a computer sitting in front of you.

The following instructions explain how to connect to your instance using an SSH client. If you receive an error while attempting to connect to your instance, see [Troubleshooting connecting to your instance \(p. 1246\)](#). For more connection options, see [Connect to your Linux instance \(p. 563\)](#).

Prerequisites

Before you connect to your Linux instance, complete the following prerequisites.

Verify that the instance is ready

After you launch an instance, it can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks. You can view this information in the **Status Checks** column on the [Instances](#) page.

Verify the general prerequisites for connecting to your instance

To find the public DNS name or IP address of your instance and the user name that you should use to connect to your instance, see [General prerequisites for connecting to your instance \(p. 563\)](#).

Install an SSH client on your local computer as needed

Your local computer might have an SSH client installed by default. You can verify this by typing **ssh** at the command line. If your compute doesn't recognize the command, you can install an SSH client.

- Recent versions of Windows Server 2019 and Windows 10 - OpenSSH is included as an installable component. For information, see [OpenSSH in Windows](#).
- Earlier versions of Windows - Download and install OpenSSH. For more information, see [Win32-OpenSSH](#).
- Linux and macOS X - Download and install OpenSSH. For more information, see <http://www.openssh.com>.

Connect to your Linux instance using an SSH client

Use the following procedure to connect to your Linux instance using an SSH client. If you receive an error while attempting to connect to your instance, see [Troubleshooting connecting to your instance \(p. 1246\)](#).

To connect to your instance using SSH

1. In a terminal window, use the `ssh` command to connect to the instance. You specify the path and file name of the private key (`.pem`), the user name for your instance, and the public DNS name or IPv6 address for your instance. For more information about how to find the private key, the user name for your instance, and the DNS name or IPv6 address for an instance, see [Locate the private key \(p. 565\)](#) and [Get information about your instance \(p. 563\)](#). To connect to your instance, use one of the following commands.

- (Public DNS) To connect using your instance's public DNS name, enter the following command.

```
ssh -i /path/my-key-pair.pem my-instance-user-name@my-instance-public-dns-name
```

- (IPv6) Alternatively, if your instance has an IPv6 address, to connect using your instance's IPv6 address, enter the following command.

```
ssh -i /path/my-key-pair.pem my-instance-user-name@my-instance-IPv6-address
```

You see a response like the following:

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (198-51-100-1)'  
can't be established.  
ECDSA key fingerprint is 14UB/neBad9tvkgJf1QZWxheOmR59WgrgzEimCG6kZY.  
Are you sure you want to continue connecting (yes/no)?
```

2. (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you previously obtained in [\(Optional\) Get the instance fingerprint \(p. 565\)](#). If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.
3. Enter **yes**.

You see a response like the following:

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (ECDSA) to the  
list of known hosts.
```

Transferring files to Linux instances from Linux using SCP

One way to transfer files between your local computer and a Linux instance is to use the secure copy protocol (SCP). This section describes how to transfer files with SCP. The procedure is similar to the procedure for connecting to an instance with SSH.

Prerequisites

- **Verify the general prerequisites for transferring files to your instance.**

The general prerequisites for transferring files to an instance are the same as the general prerequisites for connecting to an instance. For more information, see [General prerequisites for connecting to your instance \(p. 563\)](#).

- **Install an SCP client**

Most Linux, Unix, and Apple computers include an SCP client by default. If yours doesn't, the OpenSSH project provides a free implementation of the full suite of SSH tools, including an SCP client. For more information, see <http://www.openssh.org>.

The following procedure steps you through using SCP to transfer a file. If you've already connected to the instance with SSH and have verified its fingerprints, you can start with the step that contains the SCP command (step 4).

To use SCP to transfer a file

- Transfer a file to your instance using the instance's public DNS name, or the IPv6 address if your instance has one. For example, if the name of your private key file is `my-key-pair`, the file to transfer is `SampleFile.txt`, the user name for your instance is `my-instance-user-name`, and the public DNS name of the instance is `my-instance-public-dns-name`, or `my-instance-IPv6-address` if your instance has an IPv6 address, use one of the following commands to copy the file to the `my-instance-user-name` home directory.
 - (Public DNS) To transfer a file to your instance using your instance's public DNS name, enter the following command.

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt my-instance-user-name@my-instance-public-dns-name:-
```

- (IPv6) Alternatively, if your instance has an IPv6 address, to transfer a file using the instance's IPv6 address, enter the following command. The IPv6 address must be enclosed in square brackets ([]), which must be escaped (\).

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt my-instance-user-name@\[my-instance-IPv6-address\]:-
```

You see a response like the following:

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)' can't be established.  
RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.  
Are you sure you want to continue connecting (yes/no)?
```

- (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you previously obtained in [\(Optional\) Get the instance fingerprint \(p. 565\)](#). If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.
- Enter **yes**.

You see a response like the following:

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA) to the list of known hosts.  
Sending file modes: C0644 20 SampleFile.txt  
Sink: C0644 20 SampleFile.txt  
SampleFile.txt 100% 20 0.0KB/s 00:00
```

If you receive a "bash: scp: command not found" error, you must first install `scp` on your Linux instance. For some operating systems, this is located in the `openssh-clients` package. For Amazon Linux variants, such as the Amazon ECS-optimized AMI, use the following command to install `scp`:

```
[ec2-user ~]$ sudo yum install -y openssh-clients
```

- To transfer files in the other direction (from your Amazon EC2 instance to your local computer), reverse the order of the host parameters. For example, to transfer the `SampleFile.txt` file from your EC2 instance back to the home directory on your local computer as `SampleFile2.txt`, use of the following commands on your local computer.

- (Public DNS) To transfer a file from your instance using your instance's public DNS name, enter the following command.

```
scp -i /path/my-key-pair.pem my-instance-user-name@my-instance-public-dns-name:~/  
SampleFile.txt ~/SampleFile2.txt
```

- (IPv6) Alternatively, if your instance has an IPv6 address, to transfer a file using the instance's IPv6 address, enter the following command. The IPv6 address must be enclosed in square brackets ([]), which must be escaped (\).

```
scp -i /path/my-key-pair.pem my-instance-user-name@[my-instance-IPv6-address]:~/  
SampleFile.txt ~/SampleFile2.txt
```

Connecting to your Linux instance using EC2 Instance Connect

Amazon EC2 Instance Connect provides a simple and secure way to connect to your instances using Secure Shell (SSH). With EC2 Instance Connect, you use AWS Identity and Access Management (IAM) policies and principals to control SSH access to your instances, removing the need to share and manage SSH keys. All connection requests using EC2 Instance Connect are [logged to AWS CloudTrail so that you can audit connection requests \(p. 757\)](#).

You can use Instance Connect to connect to your Linux instances using a browser-based client, the Amazon EC2 Instance Connect CLI, or the SSH client of your choice.

When you connect to an instance using EC2 Instance Connect, the Instance Connect API pushes a one-time-use SSH public key to the [instance metadata \(p. 658\)](#) where it remains for 60 seconds. An IAM policy attached to your IAM user authorizes your IAM user to push the public key to the instance metadata. The SSH daemon uses `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser`, which are configured when Instance Connect is installed, to look up the public key from the instance metadata for authentication, and connects you to the instance.

Tip

If you are connecting to a Linux instance from a local computer running Windows, see the following documentation instead:

- [Connecting to your Linux instance from Windows using PuTTY \(p. 578\)](#)
- [Connecting to your Linux instance using SSH \(p. 566\)](#)
- [Connecting to your Linux instance from Windows using Windows Subsystem for Linux \(p. 583\)](#)

Contents

- [Set up EC2 Instance Connect \(p. 569\)](#)
- [Connect using EC2 Instance Connect \(p. 575\)](#)
- [Uninstall EC2 Instance Connect \(p. 577\)](#)

Set up EC2 Instance Connect

To use EC2 Instance Connect to connect to an instance, you need to configure every instance that will support using Instance Connect (this is a one-time requirement for each instance), and you need to grant permission to every IAM user that will use Instance Connect.

Tasks to set up Instance Connect

- [Task 1: Configure network access to an instance \(p. 570\)](#)
- [Task 2: \(Conditional\) Install EC2 Instance Connect on an instance \(p. 571\)](#)
- [Task 3: \(Optional\) Install the EC2 Instance Connect CLI \(p. 573\)](#)
- [Task 4: Configure IAM permissions for EC2 Instance Connect \(p. 573\)](#)

For more information about setting up EC2 Instance Connect, see [Securing your bastion hosts with Amazon EC2 Instance Connect](#).

Limitations

- The following Linux distributions are supported:
 - Amazon Linux 2 (any version)
 - Ubuntu 16.04 or later
- If you configured the `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser` settings for SSH authentication, the EC2 Instance Connect installation will not update them. As a result, you cannot use Instance Connect.

Prerequisites

- **Verify the general prerequisites for connecting to your instance using SSH.**

For more information, see [General prerequisites for connecting to your instance \(p. 563\)](#).

- **Install an SSH client on your local computer.**

Your local computer most likely has an SSH client installed by default. You can check for an SSH client by typing `ssh` at the command line. If your local computer doesn't recognize the command, you can install an SSH client. For information about installing an SSH client on Linux or macOS X, see <http://www.openssh.com>. For information about installing an SSH client on Windows 10, see [OpenSSH in Windows](#).

- **Install the AWS CLI on your local computer.**

To configure the IAM permissions, you must use the AWS CLI. For more information about installing the AWS CLI, see [Installing the AWS CLI in the AWS Command Line Interface User Guide](#).

- **[Ubuntu] Install the AWS CLI on your instance.**

To install EC2 Instance Connect on an Ubuntu instance, you must use the AWS CLI on the instance. For more information about installing the AWS CLI, see [Installing the AWS CLI in the AWS Command Line Interface User Guide](#).

Task 1: Configure network access to an instance

You must configure the following network access to your instance so that you can install EC2 Instance Connect and enable your users to connect to your instance:

- Ensure that the security group associated with your instance [allows inbound SSH traffic \(p. 983\)](#) on port 22 from your IP address. The default security group for the VPC does not allow incoming SSH traffic by default. The security group created by the launch wizard allows incoming SSH traffic by default. For more information, see [Authorizing inbound traffic for your Linux instances \(p. 983\)](#).
- (Browser-based client) We recommend that your instance allows inbound SSH traffic from the [recommended IP block published for the service](#). Use the `EC2_INSTANCE_CONNECT` filter for the service parameter to get the IP address ranges in the EC2 Instance Connect subset. For more information, see [AWS IP Address Ranges in the Amazon Web Services General Reference](#).

Task 2: (Conditional) Install EC2 Instance Connect on an instance

Amazon Linux 2 2.0.20190618 or later and Ubuntu 20.04 or later are preconfigured with EC2 Instance Connect. If you launched your instance using one of these AMIs, you can skip this task. For other supported Linux distributions, you must install Instance Connect on every instance that will support connecting using Instance Connect.

Installing Instance Connect configures the SSH daemon on the instance. The procedure for installing Instance Connect is different for instances launched using Amazon Linux 2 and Ubuntu.

Amazon Linux 2

To install EC2 Instance Connect on an instance launched with Amazon Linux 2

1. Connect to your instance using SSH.

Use the SSH key pair that was assigned to your instance when you launched it and the default user name of the AMI that you used to launch your instance. For Amazon Linux 2, the default user name is `ec2-user`.

For example, if your instance was launched using Amazon Linux 2, your instance's public DNS name is `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`, and the key pair is `my_ec2_private_key.pem`, use the following command to SSH into your instance:

```
$ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
```

For more information about connecting to your instance, see [Connecting to your Linux instance using SSH \(p. 566\)](#).

2. Install the EC2 Instance Connect package on your instance.

For Amazon Linux 2, use the `yum install` command.

```
[ec2-user ~]$ sudo yum install ec2-instance-connect
```

You should see four new files in the `/opt/aws/bin/` folder:

```
eic_curlAuthorizedKeys  
eic_harvestHostkeys  
eic_parseAuthorizedKeys  
eic_runAuthorizedKeys
```

3. (Optional) Verify that Instance Connect was successfully installed on your instance.

Use the `sudo less` command to check that the `/etc/ssh/sshd_config` file was correctly updated as follows:

```
[ec2-user ~]$ sudo less /etc/ssh/sshd_config
```

Instance Connect was successfully installed if the `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser` lines in the `/etc/ssh/sshd_config` file contain the following values:

```
AuthorizedKeysCommand /opt/aws/bin/eic_runAuthorizedKeys %u %f  
AuthorizedKeysCommandUser ec2-instance-connect
```

- `AuthorizedKeysCommand` sets the `eic_run_authorized_keys` file to look up the keys from the instance metadata
- `AuthorizedKeysCommandUser` sets the system user as `ec2-instance-connect`

Note

If you previously configured `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser`, the Instance Connect installation will not change the values and you will not be able to use Instance Connect.

Ubuntu

To install EC2 Instance Connect on an instance launched with Ubuntu 16.04 or later

1. Connect to your instance using SSH.

Use the SSH key pair that was assigned to your instance when you launched it and use the default user name of the AMI that you used to launch your instance. For an Ubuntu AMI, the user name is `ubuntu`.

If your instance was launched using Ubuntu, your instance's public DNS name is `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`, and the key pair is `my_ec2_private_key.pem`, use the following command to SSH into your instance:

```
$ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
```

For more information about connecting to your instance, see [Connecting to your Linux instance using SSH \(p. 566\)](#).

2. (Optional) Ensure your instance has the latest Ubuntu AMI.

For Ubuntu, use the following commands to update all the packages on your instance.

```
ubuntu:~$ sudo apt-get update
```

```
ubuntu:~$ sudo apt-get upgrade
```

3. Install the Instance Connect package on your instance.

For Ubuntu, use the `sudo apt-get` command to install the `.deb` package.

```
ubuntu:~$ sudo apt-get install ec2-instance-connect
```

You should see four new files in the `/usr/share/ec2-instance-connect/` folder:

```
eic_curl_authorized_keys
eic_harvest_hostkeys
eic_parse_authorized_keys
eic_run_authorized_keys
```

4. (Optional) Verify that Instance Connect was successfully installed on your instance.

Use the `sudo less` command to check that the `/lib/systemd/system/ssh.service.d/ec2-instance-connect.conf` was correctly updated as follows:

```
ubuntu:~$ sudo less /lib/systemd/system/ssh.service.d/ec2-instance-connect.conf
```

Instance Connect was successfully installed if the `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser` lines in the `/lib/systemd/system/ssh.service.d/ec2-instance-connect.conf` file contain the following values:

```
AuthorizedKeysCommand /usr/share/ec2-instance-connect/eic_run_authorized_keys %u %f
AuthorizedKeysCommandUser ec2-instance-connect
```

- `AuthorizedKeysCommand` sets the `eic_run_authorized_keys` file to look up the keys from the instance metadata
- `AuthorizedKeysCommandUser` sets the system user as `ec2-instance-connect`

Note

If you previously configured `AuthorizedKeysCommand` and `AuthorizedKeysCommandUser`, the Instance Connect installation will not change the values and you will not be able to use Instance Connect.

For more information about the EC2 Instance Connect package, see [aws/aws-ec2-instance-connect-config](#) on the GitHub website.

Task 3: (Optional) Install the EC2 Instance Connect CLI

The EC2 Instance Connect CLI provides a similar interface to standard SSH calls, which includes querying EC2 instance information, generating and publishing ephemeral public keys, and establishing an SSH connection through a single command, `mssh instance_id`.

Note

There is no need to install the EC2 Instance Connect CLI if users will only use the browser-based client or an SSH client to connect to an instance.

To install the EC2 Instance Connect CLI package

Use `pip` to install the `ec2instanceconnectcli` package. For more information, see [aws/aws-ec2-instance-connect-cli](#) on the GitHub website, and <https://pypi.org/project/ec2instanceconnectcli/> on the Python Package Index (PyPI) website.

```
$ pip install ec2instanceconnectcli
```

Task 4: Configure IAM permissions for EC2 Instance Connect

For your IAM users to connect to an instance using EC2 Instance Connect, you must grant them permission to push the public key to the instance. You grant them the permission by creating an IAM policy and attaching the policy to the IAM users that require the permission. For more information, see [Actions, Resources, and Condition Keys for Amazon EC2 Instance Connect](#) in the *IAM User Guide*.

The following instructions explain how to create the policy and attach it using the AWS CLI. For instructions that use the AWS Management Console, see [Creating IAM Policies \(Console\)](#) and [Adding Permissions by Attaching Policies Directly to the User](#) in the *IAM User Guide*.

To grant an IAM user permission for EC2 Instance Connect (AWS CLI)

1. Create a JSON policy document that includes the following:

- The `ec2-instance-connect:SendSSHPublicKey` action. This grants an IAM user permission to push the public key to an instance. With `ec2-instance-connect:SendSSHPublicKey`,

consider restricting access to specific EC2 instances. Otherwise, all IAM users with this permission can connect to all EC2 instances.

- The `ec2:osuser` condition. This specifies the name of the OS user that can push the public key to an instance. Use the default user name for the AMI that you used to launch the instance. The default user name for Amazon Linux 2 is `ec2-user`, and for Ubuntu it's `ubuntu`.
- The `ec2:DescribeInstances` action. This is required when using the EC2 Instance Connect CLI because the wrapper calls this action. IAM users might already have permission to call this action from another policy.

The following is an example policy document. You can omit the statement for the `ec2:DescribeInstances` action if your users will only use an SSH client to connect to your instances. You can replace the specified instances in `Resource` with the wildcard `*` to grant users access to all EC2 instances using EC2 Instance Connect.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2-instance-connect:SendSSHPublicKey",  
            "Resource": [  
                "arn:aws:ec2:region:account-id:instance/i-1234567890abcdef0",  
                "arn:aws:ec2:region:account-id:instance/i-0598c7d356eba48d7"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "ec2:osuser": "ami-username"  
                }  
            },  
            {  
                "Effect": "Allow",  
                "Action": "ec2:DescribeInstances",  
                "Resource": "*"  
            }  
        ]  
    }  
}
```

The preceding policy allows access to specific instances, identified by their instance ID. Alternatively, you can use resource tags to control access to an instance. Attribute-based access control is an authorization strategy that defines permissions based on tags that can be attached to users and AWS resources. For example, the following policy allows an IAM user to access an instance only if that instance has a resource tag with `key=tag-key` and `value=tag-value`. For more information about using tags to control access to your AWS resources, see [Controlling Access to AWS Resources](#) in the *IAM User Guide*.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2-instance-connect:SendSSHPublicKey",  
            "Resource": "arn:aws:ec2:region:account-id:instance/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:ResourceTag/tag-key": "tag-value"  
                }  
            },  
            {  
                "Effect": "Allow",  
                "Action": "ec2:DescribeInstances",  
                "Resource": "*"  
            }  
        }  
    ]  
}
```

```
        "Effect": "Allow",
        "Action": "ec2:DescribeInstances",
        "Resource": "*"
    }
}
```

2. Use the [create-policy](#) command to create a new managed policy, and specify the JSON document that you created to use as the content for the new policy.

```
$ aws iam create-policy --policy-name my-policy --policy-document file://JSON-file-name
```

3. Use the [attach-user-policy](#) command to attach the managed policy to the specified IAM user. For the `--user-name` parameter, specify the friendly name (not the ARN) of the IAM user.

```
$ aws iam attach-user-policy --policy-arn arn:aws:iam::account-id:policy/my-policy --
user-name IAM-friendly-name
```

Connect using EC2 Instance Connect

The following instructions explain how to connect to your Linux instance using EC2 Instance Connect.

Options to connect using EC2 Instance Connect

- [Connect using the browser-based client \(p. 576\)](#)
- [Connect using the EC2 Instance Connect CLI \(p. 576\)](#)
- [Connect using your own key and SSH client \(p. 576\)](#)

Limitations

- The following Linux distributions are supported:
 - Amazon Linux 2 (any version)
 - Ubuntu 16.04 or later
- To connect using the browser-based client, the instance must have a public IPv4 address.
- If the instance does not have a public IP address, then you can only connect to the instance using the EC2 Instance Connect CLI, and only from a machine within the same VPC.
- EC2 Instance Connect does not support connecting using an IPv6 address.

Prerequisites

- **Install Instance Connect on your instance.**

For more information, see [Set up EC2 Instance Connect \(p. 569\)](#).

- **(Optional) Install an SSH client on your local computer.**

There is no need to install an SSH client if users only use the console or the EC2 Instance Connect CLI to connect to an instance. Your local computer most likely has an SSH client installed by default. You can check for an SSH client by typing `ssh` at the command line. If your local computer doesn't recognize the command, you can install an SSH client. For information about installing an SSH client on Linux or macOS X, see <http://www.openssh.com>. For information about installing an SSH client on Windows 10, see [OpenSSH in Windows](#).

- **(Optional) Install the EC2 Instance Connect CLI on your local computer.**

There is no need to install the EC2 Instance Connect CLI if users only use the console or an SSH client to connect to an instance. For more information, see [Task 3: \(Optional\) Install the EC2 Instance Connect CLI \(p. 573\)](#).

Connect using the browser-based client

You can connect to an instance using the browser-based client by selecting the instance from the Amazon EC2 console and choosing to connect using EC2 Instance Connect. Instance Connect handles the permissions and provides a successful connection.

To connect to your instance using the browser-based client from the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Connect**.
4. Choose **EC2 Instance Connect (browser-based SSH connection)**, **Connect**.

A window opens, and you are connected to your instance.

Connect using the EC2 Instance Connect CLI

You can connect to an instance using the EC2 Instance Connect CLI by providing only the instance ID, while the Instance Connect CLI performs the following three actions in one call: it generates a one-time-use SSH public key, pushes the key to the instance where it remains for 60 seconds, and connects the user to the instance. You can use basic SSH/SFTP commands with the Instance Connect CLI.

Note

`-i` is not supported when using **mssh**. When using the **mssh** command to connect to your instance, you do not need to specify any kind of identity file because Instance Connect manages the key pair.

Amazon Linux 2

To connect to an instance using the EC2 Instance Connect CLI

Use the **mssh** command with the instance ID as follows. You do not need to specify the user name for the AMI.

```
$ mssh i-001234a4bf70dec41EXAMPLE
```

Ubuntu

To connect to an instance using the EC2 Instance Connect CLI

Use the **mssh** command with the instance ID and the default user name for the Ubuntu AMI as follows. You must specify the user name for the AMI or you get the following error: Authentication failed.

```
$ mssh ubuntu@i-001234a4bf70dec41EXAMPLE
```

Connect using your own key and SSH client

You can use your own SSH key and connect to your instance from the SSH client of your choice while using the EC2 Instance Connect API. This enables you to benefit from the Instance Connect capability to push a public key to the instance.

Requirement

The supported RSA key types are OpenSSH and SSH2. The supported lengths are 2048 and 4096. For more information, see [Option 2: Import your own public key to Amazon EC2 \(p. 987\)](#).

To connect to your instance using your own key and any SSH client

1. (Optional) Generate new SSH private and public keys

You can generate new SSH private and public keys, `my_rsa_key` and `my_rsa_key.pub`, using the following command:

```
$ ssh-keygen -t rsa -f my_rsa_key
```

2. Push your SSH public key to the instance

Use the [send-ssh-public-key](#) command to push your SSH public key to the instance. If you launched your instance using Amazon Linux 2, the default user name for the AMI is `ec2-user`. If you launched your instance using Ubuntu, the default user name for the AMI is `ubuntu`.

The following example pushes the public key to the specified instance in the specified Availability Zone, to authenticate `ec2-user`:

```
$ aws ec2-instance-connect send-ssh-public-key \
  --instance-id i-001234a4bf70dec41EXAMPLE \
  --availability-zone us-west-2b \
  --instance-os-user ec2-user \
  --ssh-public-key file:///my_rsa_key.pub
```

3. Connect to the instance using your private key

Use the `ssh` command to connect to the instance using the private key before the public key is removed from the instance metadata (you have 60 seconds before it is removed). Specify the private key that corresponds to the public key, the default user name for the AMI that you used to launch your instance, and the instance's public DNS name. Add the `IdentitiesOnly=yes` option to ensure that only the files in the `ssh config` and the specified key are used for the connection.

```
$ ssh -o "IdentitiesOnly=yes" -i my_rsa_key ec2-
user@ec2-198-51-100-1.compute-1.amazonaws.com
```

Uninstall EC2 Instance Connect

To disable EC2 Instance Connect, connect to your instance and uninstall the `ec2-instance-connect` package that you installed on the OS. If the `sshd` configuration matches what it was set to when you installed EC2 Instance Connect, uninstalling `ec2-instance-connect` also removes the `sshd` configuration. If you modified the `sshd` configuration after installing EC2 Instance Connect, you must update it manually.

Amazon Linux 2

You can uninstall EC2 Instance Connect on Amazon Linux 2 2.0.20190618 or later, where EC2 Instance Connect is preconfigured.

To uninstall EC2 Instance Connect on an instance launched with Amazon Linux 2

1. Connect to your instance using SSH. Specify the SSH key pair you used for your instance when you launched it and the default user name for the Amazon Linux 2 AMI, which is `ec2-user`.

For example, the following **ssh** command connects to the instance with the public DNS name `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`, using the key pair `my_ec2_private_key.pem`.

```
$ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
```

2. Uninstall the `ec2-instance-connect` package using the **yum** command.

```
[ec2-user ~]$ sudo yum remove ec2-instance-connect
```

Ubuntu

To uninstall EC2 Instance Connect on an instance launched with an Ubuntu AMI

1. Connect to your instance using SSH. Specify the SSH key pair you used for your instance when you launched it and the default user name for the Ubuntu AMI, which is `ubuntu`.

For example, the following **ssh** command connects to the instance with the public DNS name `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`, using the key pair `my_ec2_private_key.pem`.

```
$ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
```

2. Uninstall the `ec2-instance-connect` package using the **apt-get** command.

```
ubuntu:~$ sudo apt-get remove ec2-instance-connect
```

Connecting to your Linux instance from Windows using PuTTY

After you launch your instance, you can connect to it and use it the way that you'd use a computer sitting in front of you.

The following instructions explain how to connect to your instance using PuTTY, a free SSH client for Windows. If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](#).

Prerequisites

Before you connect to your Linux instance using PuTTY, complete the following prerequisites.

Verify that the instance is ready

After you launch an instance, it can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks. You can view this information in the **Status Checks** column on the [Instances](#) page.

Verify the general prerequisites for connecting to your instance

To find the public DNS name or IP address of your instance and the user name that you should use to connect to your instance, see [General prerequisites for connecting to your instance \(p. 563\)](#).

Install PuTTY on your local computer

Download and install PuTTY from the [PuTTY download page](#). If you already have an older version of PuTTY installed, we recommend that you download the latest version. Be sure to install the entire suite.

Convert your private key using PuTTYgen

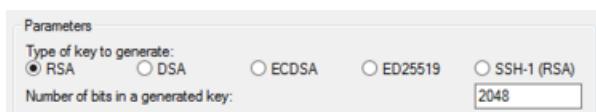
Locate the private key (.pem file) for the key pair that you specified when you launched the instance. Convert the .pem file to a .ppk file for use with PuTTY. For more information, follow the steps in the next section.

Convert your private key using PuTTYgen

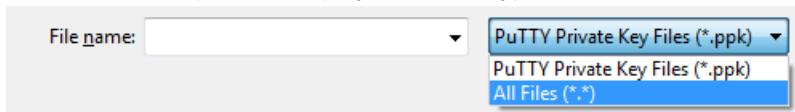
PuTTY does not natively support the private key format for SSH keys. PuTTY provides a tool named PuTTYgen, which converts keys to the required format for PuTTY. You must convert your private key (.pem file) into this format (.ppk file) as follows in order to connect to your instance using PuTTY.

To convert your private key

1. From the **Start** menu, choose **All Programs, PuTTY, PuTTYgen**.
2. Under **Type of key to generate**, choose **RSA**. If you're using an older version of PuTTYgen, choose **SSH-2 RSA**.



3. Choose **Load**. By default, PuTTYgen displays only files with the extension .ppk. To locate your .pem file, choose the option to display files of all types.



4. Select your .pem file for the key pair that you specified when you launched your instance and choose **Open**. PuTTYgen displays a notice that the .pem file was successfully imported. Choose **OK**.
5. To save the key in the format that PuTTY can use, choose **Save private key**. PuTTYgen displays a warning about saving the key without a passphrase. Choose **Yes**.

Note

A passphrase on a private key is an extra layer of protection. Even if your private key is discovered, it can't be used without the passphrase. The downside to using a passphrase is that it makes automation harder because human intervention is needed to log on to an instance, or to copy files to an instance.

6. Specify the same name for the key that you used for the key pair (for example, my-key-pair) and choose **Save**. PuTTY automatically adds the .ppk file extension.

Your private key is now in the correct format for use with PuTTY. You can now connect to your instance using PuTTY's SSH client.

Connecting to your Linux instance

Use the following procedure to connect to your Linux instance using PuTTY. You need the .ppk file that you created for your private key. For more information, see [Convert your private key using PuTTYgen \(p. 579\)](#) in the preceding section. If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](#).

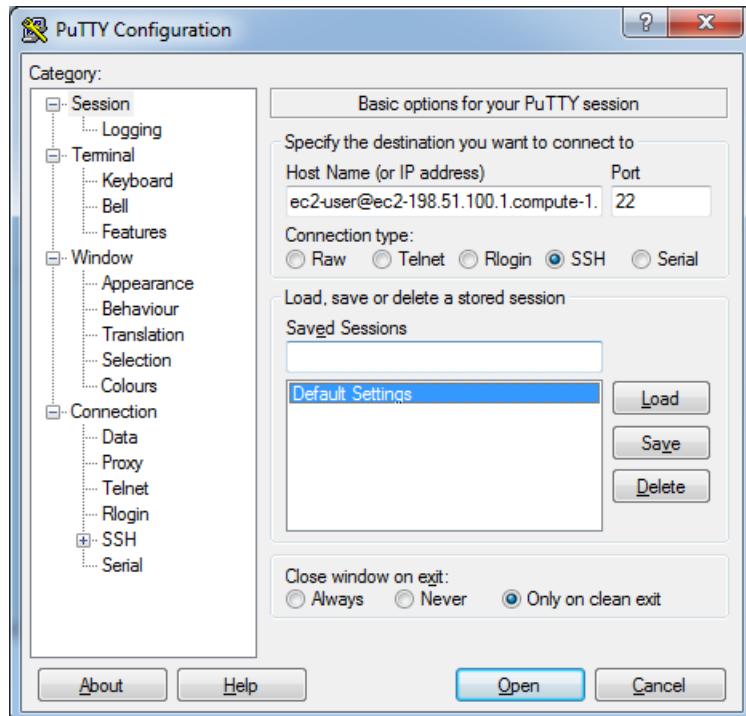
To connect to your instance using PuTTY

1. Start PuTTY (from the **Start** menu, choose **All Programs, PuTTY, PuTTY**).
2. In the **Category** pane, choose **Session** and complete the following fields:
 - a. In the **Host Name** box, do one of the following:

- (Public DNS) To connect using your instance's public DNS name, enter *my-instance-user-name@my-instance-public-dns-name*.
- (IPv6) Alternatively, if your instance has an IPv6 address, to connect using your instance's IPv6 address, enter *my-instance-user-name@my-instance-IPv6-address*.

For information about how to get the user name for your instance, and the public DNS name or IPv6 address of your instance, see [Get information about your instance \(p. 563\)](#).

- Ensure that the **Port** value is 22.
- Under **Connection type**, select **SSH**.



- (Optional) You can configure PuTTY to automatically send 'keepalive' data at regular intervals to keep the session active. This is useful to avoid disconnecting from your instance due to session inactivity. In the **Category** pane, choose **Connection**, and then enter the required interval in the **Seconds between keepalives** field. For example, if your session disconnects after 10 minutes of inactivity, enter 180 to configure PuTTY to send keepalive data every 3 minutes.
- In the **Category** pane, expand **Connection**, expand **SSH**, and then choose **Auth**. Complete the following:
 - Choose **Browse**.
 - Select the .ppk file that you generated for your key pair and choose **Open**.
 - (Optional) If you plan to start this session again later, you can save the session information for future use. Under **Category**, choose **Session**, enter a name for the session in **Saved Sessions**, and then choose **Save**.
 - Choose **Open**.
- If this is the first time you have connected to this instance, PuTTY displays a security alert dialog box that asks whether you trust the host to which you are connecting.
 - (Optional) Verify that the fingerprint in the security alert dialog box matches the fingerprint that you previously obtained in [\(Optional\) Get the instance fingerprint \(p. 565\)](#). If these

fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.

- b. Choose **Yes**. A window opens and you are connected to your instance.

Note

If you specified a passphrase when you converted your private key to PuTTY's format, you must provide that passphrase when you log in to the instance.

If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](#).

Transferring files to your Linux instance using the PuTTY Secure Copy client

The PuTTY Secure Copy client (PSCP) is a command line tool that you can use to transfer files between your Windows computer and your Linux instance. If you prefer a graphical user interface (GUI), you can use an open source GUI tool named WinSCP. For more information, see [Transferring files to your Linux instance using WinSCP \(p. 581\)](#).

To use PSCP, you need the private key you generated in [Convert your private key using PuTTYgen \(p. 579\)](#). You also need the public DNS name of your Linux instance, or the IPv6 address if your instance has one.

The following example transfers the file `Sample_file.txt` from the C:\ drive on a Windows computer to the `my-instance-user-name` home directory on an Amazon Linux instance. To transfer a file, use one of the following commands.

- (Public DNS) To transfer a file using your instance's public DNS name, enter the following command.

```
pscp -i C:\\path\\my-key-pair.ppk C:\\path\\Sample_file.txt my-instance-user-name@my-
instance-public-dns-name:/home/my-instance-user-name/Sample_file.txt
```

- (IPv6) Alternatively, if your instance has an IPv6 address, to transfer a file using your instance's IPv6 address, enter the following command. The IPv6 address must be enclosed in square brackets ([]).

```
pscp -i C:\\path\\my-key-pair.ppk C:\\path\\Sample_file.txt my-instance-user-name@[my-
instance-IPv6-address]:/home/my-instance-user-name/Sample_file.txt
```

Transferring files to your Linux instance using WinSCP

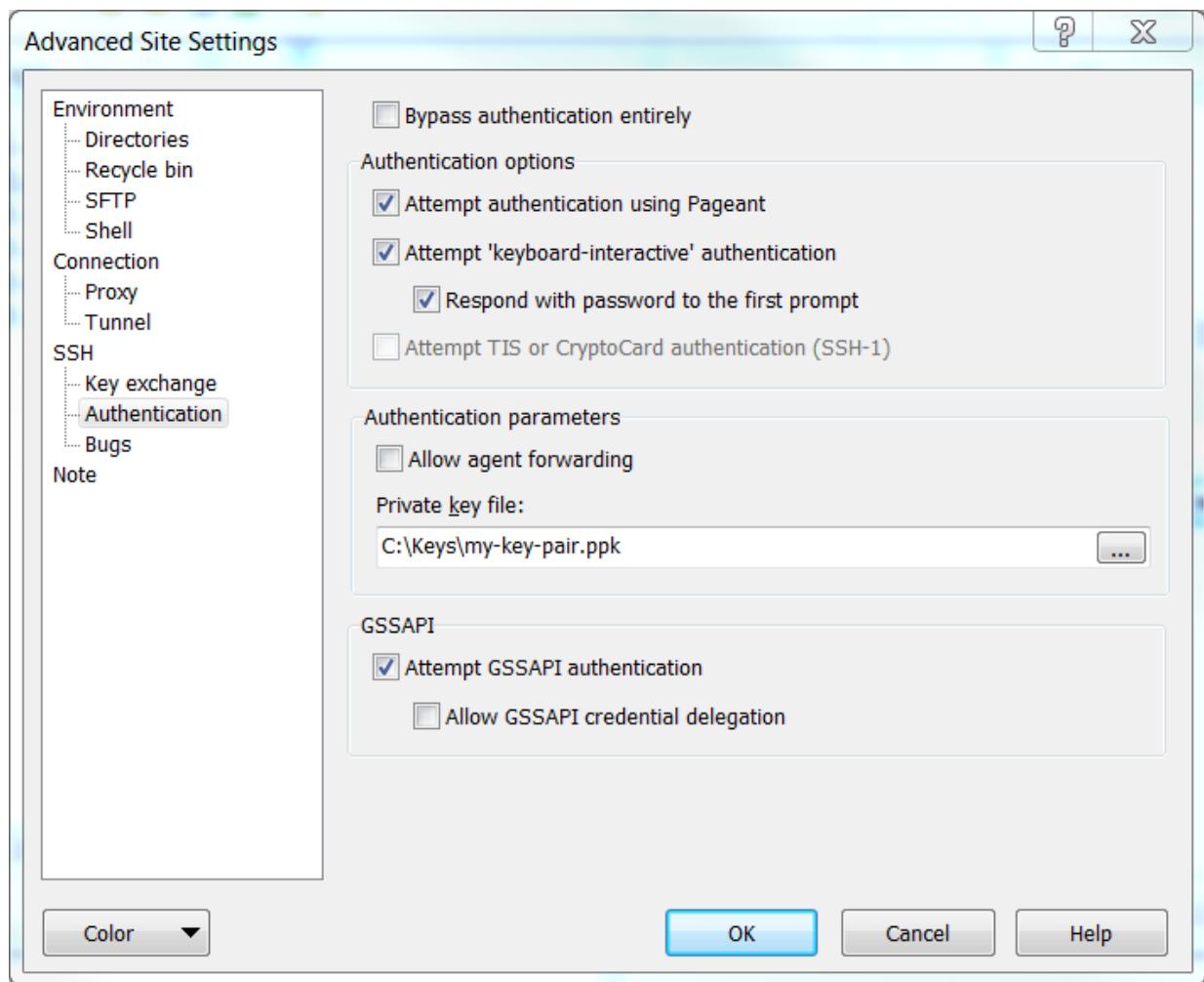
WinSCP is a GUI-based file manager for Windows that allows you to upload and transfer files to a remote computer using the SFTP, SCP, FTP, and FTPS protocols. WinSCP allows you to drag and drop files from your Windows computer to your Linux instance or synchronize entire directory structures between the two systems.

To use WinSCP, you need the private key that you generated in [Convert your private key using PuTTYgen \(p. 579\)](#). You also need the public DNS name of your Linux instance.

1. Download and install WinSCP from <http://winscp.net/eng/download.php>. For most users, the default installation options are OK.
2. Start WinSCP.
3. At the **WinSCP login** screen, for **Host name**, enter one of the following:
 - (Public DNS or IPv4 address) To log in using your instance's public DNS name or public IPv4 address, enter the public DNS name or public IPv4 address for your instance.
 - (IPv6) Alternatively, if your instance has an IPv6 address, to log in using your instance's IPv6 address, enter the IPv6 address for your instance.

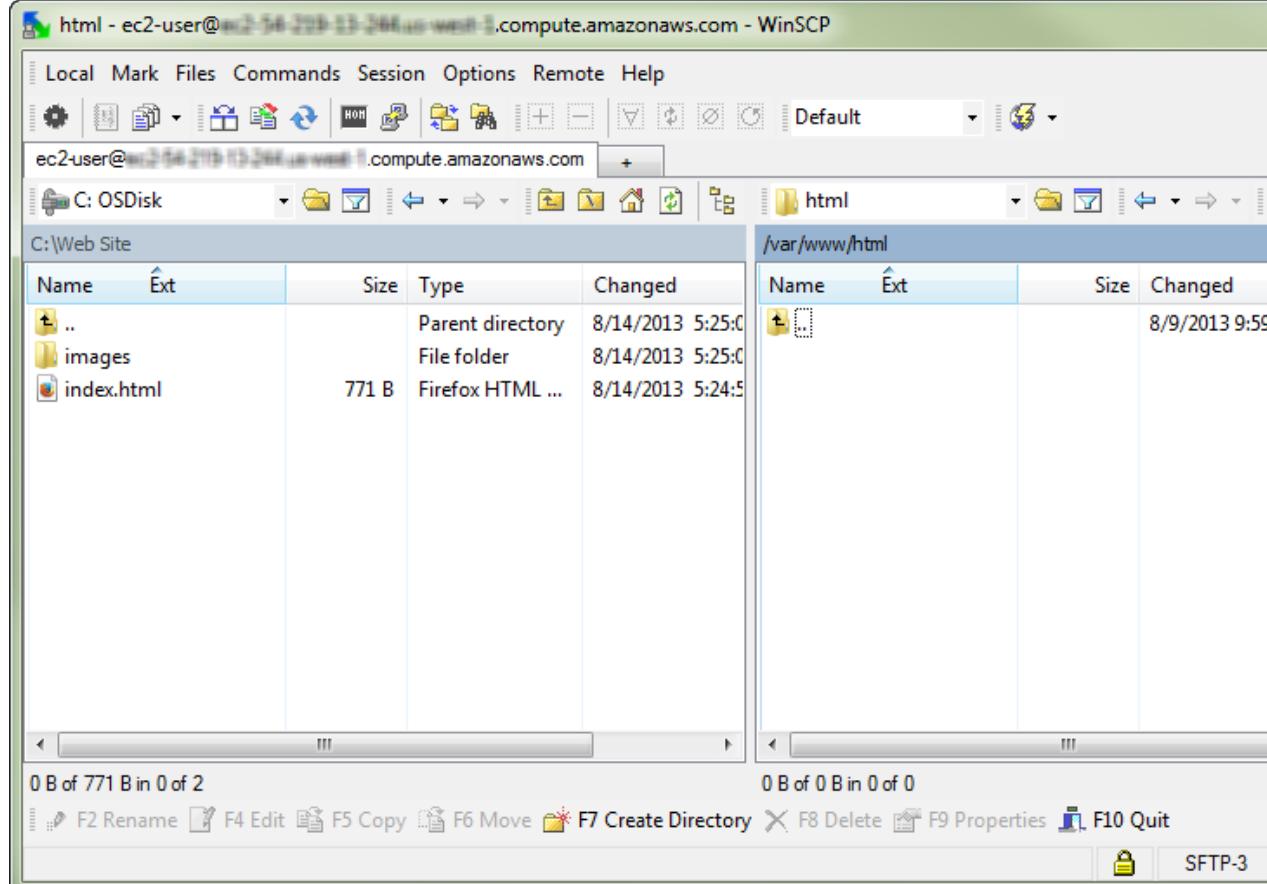
4. For **User name**, enter the default user name for your AMI.
 - For Amazon Linux 2 or the Amazon Linux AMI, the user name is `ec2-user`.
 - For a CentOS AMI, the user name is `centos`.
 - For a Debian AMI, the user name is `admin`.
 - For a Fedora AMI, the user name is `ec2-user` or `fedora`.
 - For a RHEL AMI, the user name is `ec2-user` or `root`.
 - For a SUSE AMI, the user name is `ec2-user` or `root`.
 - For an Ubuntu AMI, the user name is `ubuntu`.
 - Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.
5. Specify the private key for your instance. For **Private key**, enter the path to your private key, or choose the "... " button to browse for the file. To open the advanced site settings, for newer versions of WinSCP, choose **Advanced**. To find the **Private key file** setting, under **SSH**, choose **Authentication**.

Here is a screenshot from WinSCP version 5.9.4:



WinSCP requires a PuTTY private key file (`.ppk`). You can convert a `.pem` security key file to the `.ppk` format using PuTTYgen. For more information, see [Convert your private key using PuTTYgen \(p. 579\)](#).

6. (Optional) In the left panel, choose **Directories**. For **Remote directory**, enter the path for the directory to which to add files. To open the advanced site settings for newer versions of WinSCP, choose **Advanced**. To find the **Remote directory** setting, under **Environment**, choose **Directories**.
7. Choose **Login**. To add the host fingerprint to the host cache, choose **Yes**.



8. After the connection is established, in the connection window your Linux instance is on the right and your local machine is on the left. You can drag and drop files directly into the remote file system from your local machine. For more information on WinSCP, see the project documentation at <http://winscp.net/eng/docs/start>.

If you receive a "Cannot execute SCP to start transfer" error, you must first install **scp** on your Linux instance. For some operating systems, this is located in the **openssh-clients** package. For Amazon Linux variants, such as the Amazon ECS-optimized AMI, use the following command to install **scp**.

```
[ec2-user ~]$ sudo yum install -y openssh-clients
```

Connecting to your Linux instance from Windows using Windows Subsystem for Linux

After you launch your instance, you can connect to it and use it the way that you'd use a computer sitting in front of you.

The following instructions explain how to connect to your instance using a Linux distribution on the Windows Subsystem for Linux (WSL). WSL is a free download and enables you to run native Linux

command line tools directly on Windows, alongside your traditional Windows desktop, without the overhead of a virtual machine.

By installing WSL, you can use a native Linux environment to connect to your Linux EC2 instances instead of using PuTTY or PuTTYgen. The Linux environment makes it easier to connect to your Linux instances because it comes with a native SSH client that you can use to connect to your Linux instances and change the permissions of the .pem key file. The Amazon EC2 console provides the SSH command for connecting to the Linux instance, and you can get verbose output from the SSH command for troubleshooting. For more information, see the [Windows Subsystem for Linux Documentation](#).

Note

After you've installed the WSL, all the prerequisites and steps are the same as those described in [Connecting to your Linux instance using SSH \(p. 566\)](#), and the experience is just like using native Linux.

If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](#).

Contents

- [Prerequisites \(p. 566\)](#)
- [Connect to your Linux instance using WSL \(p. 585\)](#)
- [Transferring files to Linux instances from Linux using SCP \(p. 585\)](#)
- [Uninstalling WSL \(p. 587\)](#)

Prerequisites

Before you connect to your Linux instance, complete the following prerequisites.

Verify that the instance is ready

After you launch an instance, it can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks. You can view this information in the **Status Checks** column on the [Instances](#) page.

Verify the general prerequisites for connecting to your instance

To find the public DNS name or IP address of your instance and the user name that you should use to connect to your instance, see [General prerequisites for connecting to your instance \(p. 563\)](#).

Install the Windows Subsystem for Linux (WSL) and a Linux distribution on your local computer

Install the WSL and a Linux distribution using the instructions in the [Windows 10 Installation Guide](#). The example in the instructions installs the Ubuntu distribution of Linux, but you can install any distribution. You are prompted to restart your computer for the changes to take effect.

Copy the private key from Windows to WSL

In a WSL terminal window, copy the .pem file (for the key pair that you specified when you launched the instance) from Windows to WSL. Note the fully-qualified path to the .pem file on WSL to use when connecting to your instance. For information about how to specify the path to your Windows hard drive, see [How do I access my C drive?](#). For more information about key pairs and Windows instances, see [Amazon EC2 key pairs and Windows instances](#).

```
cp /mnt/<Windows drive letter>/path/my-key-pair.pem ~/WSL-path/my-key-pair.pem
```

Connect to your Linux instance using WSL

Use the following procedure to connect to your Linux instance using the Windows Subsystem for Linux (WSL). If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](#).

To connect to your instance using SSH

1. In a terminal window, use the `ssh` command to connect to the instance. You specify the path and file name of the private key (`.pem`), the user name for your instance, and the public DNS name or IPv6 address for your instance. For more information about how to find the private key, the user name for your instance, and the DNS name or IPv6 address for an instance, see [Locate the private key \(p. 565\)](#) and [Get information about your instance \(p. 563\)](#). To connect to your instance, use one of the following commands.

- (Public DNS) To connect using your instance's public DNS name, enter the following command.

```
sudo ssh -i /path/my-key-pair.pem my-instance-user-name@my-instance-public-dns-name
```

- (IPv6) Alternatively, if your instance has an IPv6 address, you can connect to the instance using its IPv6 address. Specify the `ssh` command with the path to the private key (`.pem`) file, the appropriate user name, and the IPv6 address.

```
sudo ssh -i /path/my-key-pair.pem my-instance-user-name@my-instance-IPv6-address
```

You see a response like the following:

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)'  
can't be established.  
RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.  
Are you sure you want to continue connecting (yes/no)?
```

2. (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you previously obtained in [\(Optional\) Get the instance fingerprint \(p. 565\)](#). If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.
3. Enter yes.

You see a response like the following:

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)  
to the list of known hosts.
```

Transferring files to Linux instances from Linux using SCP

One way to transfer files between your local computer and a Linux instance is to use the secure copy protocol (SCP). This section describes how to transfer files with SCP. The procedure is similar to the procedure for connecting to an instance with SSH.

Prerequisites

- **Verify the general prerequisites for transferring files to your instance.**

The general prerequisites for transferring files to an instance are the same as the general prerequisites for connecting to an instance. For more information, see [General prerequisites for connecting to your instance \(p. 563\)](#).

- **Install an SCP client**

Most Linux, Unix, and Apple computers include an SCP client by default. If yours doesn't, the OpenSSH project provides a free implementation of the full suite of SSH tools, including an SCP client. For more information, see <http://www.openssh.org>.

The following procedure steps you through using SCP to transfer a file. If you've already connected to the instance with SSH and have verified its fingerprints, you can start with the step that contains the SCP command (step 4).

To use SCP to transfer a file

1. Transfer a file to your instance using the instance's public DNS name. For example, if the name of the private key file is `my-key-pair`, the file to transfer is `SampleFile.txt`, the user name is `my-instance-user-name`, and the public DNS name of the instance is `my-instance-public-dns-name` or the IPv6 address is `my-instance-IPv6-address`, use one the following commands to copy the file to the `my-instance-user-name` home directory.
 - (Public DNS) To transfer a file using your instance's public DNS name, enter the following command.

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt my-instance-user-name@my-instance-public-dns-name:~
```

- (IPv6) Alternatively, if your instance has an IPv6 address, you can transfer a file using the instance's IPv6 address. The IPv6 address must be enclosed in square brackets ([]), which must be escaped (\).


```
scp -i /path/my-key-pair.pem /path/SampleFile.txt my-instance-user-name@[my-instance-IPv6-address]\:-
```

You see a response like the following:

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)' can't be established.  
RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.  
Are you sure you want to continue connecting (yes/no)?
```

2. (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you previously obtained in [\(Optional\) Get the instance fingerprint \(p. 565\)](#). If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.
3. Enter **yes**.

You see a response like the following:

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)  
to the list of known hosts.  
Sending file modes: C0644 20 SampleFile.txt  
Sink: C0644 20 SampleFile.txt  
SampleFile.txt 100% 20 0.0KB/s 00:00
```

If you receive a "bash: scp: command not found" error, you must first install `scp` on your Linux instance. For some operating systems, this is located in the `openssh-clients` package. For Amazon Linux variants, such as the Amazon ECS-optimized AMI, use the following command to install `scp`:

```
[ec2-user ~]$ sudo yum install -y openssh-clients
```

4. To transfer files in the other direction (from your Amazon EC2 instance to your local computer), reverse the order of the host parameters. For example, to transfer the `SampleFile.txt` file from your EC2 instance back to the home directory on your local computer as `SampleFile2.txt`, use one of the following commands on your local computer.
 - (Public DNS) To transfer a file using your instance's public DNS name, enter the following command.

```
scp -i /path/my-key-pair.pem my-instance-user-name@ec2-198-51-100-1.compute-1.amazonaws.com:~/SampleFile.txt ~/SampleFile2.txt
```

4. (IPv6) Alternatively, if your instance has an IPv6 address, to transfer files in the other direction using the instance's IPv6 address, enter the following command.

```
scp -i /path/my-key-pair.pem my-instance-user-name@[2001:db8:1234:1a00:9691:9503:25ad:1761]:~/SampleFile.txt ~/SampleFile2.txt
```

Uninstalling WSL

For information about uninstalling Windows Subsystem for Linux, see [How do I uninstall a WSL Distribution?](#).

Connecting to your Linux instance using Session Manager

Session Manager is a fully managed AWS Systems Manager capability that lets you manage your Amazon EC2 instances through an interactive one-click browser-based shell or through the AWS CLI. You can use Session Manager to start a session with an instance in your account. After the session is started, you can run bash commands as you would through any other connection type. For more information about Session Manager, see [AWS Systems Manager Session Manager](#) in the *AWS Systems Manager User Guide*.

Before attempting to connect to an instance using Session Manager, ensure that the necessary setup steps have been completed. For more information and instructions, see [Getting Started with Session Manager](#).

To connect to a Linux instance using Session Manager using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Connect**.
4. For **Connection method**, choose **Session Manager**.
5. Choose **Connect**.

Troubleshooting

If you receive an error that you're not authorized to perform one or more Systems Manager actions (`ssm:command-name`), then you must update your policies to allow you to start sessions from the Amazon EC2 console. For more information, see [Quickstart Default IAM Policies for Session Manager](#) in the *AWS Systems Manager User Guide*.

Stop and start your instance

You can stop and start your instance if it has an Amazon EBS volume as its root device. The instance retains its instance ID, but can change as described in the [Overview \(p. 588\)](#) section.

When you stop an instance, we shut it down. We don't charge usage for a stopped instance, or data transfer fees, but we do charge for the storage for any Amazon EBS volumes. Each time you start a

stopped instance we charge a minimum of one minute for usage. After one minute, we charge only for the seconds you use. For example, if you run an instance for 20 seconds and then stop it, we charge for a full one minute. If you run an instance for 3 minutes and 40 seconds, we charge for exactly 3 minutes and 40 seconds of usage.

While the instance is stopped, you can treat its root volume like any other volume, and modify it (for example, repair file system problems or update software). You just detach the volume from the stopped instance, attach it to a running instance, make your changes, detach it from the running instance, and then reattach it to the stopped instance. Make sure that you reattach it using the storage device name that's specified as the root device in the block device mapping for the instance.

If you decide that you no longer need an instance, you can terminate it. As soon as the state of an instance changes to `shutting-down` or `terminated`, we stop charging for that instance. For more information, see [Terminate your instance \(p. 607\)](#). If you'd rather hibernate the instance, see [Hibernate your Linux instance \(p. 591\)](#). For more information, see [Differences between reboot, stop, hibernate, and terminate \(p. 496\)](#).

Contents

- [Overview \(p. 588\)](#)
- [What happens when you stop an instance \(p. 589\)](#)
- [Stopping and starting your instances \(p. 589\)](#)
- [Modifying a stopped instance \(p. 590\)](#)
- [Troubleshooting \(p. 591\)](#)

Overview

You can only stop an Amazon EBS-backed instance. To verify the root device type of your instance, describe the instance and check whether the device type of its root volume is `ebs` (Amazon EBS-backed instance) or `instance-store` (instance store-backed instance). For more information, see [Determining the root device type of your AMI \(p. 100\)](#).

When you stop a running instance, the following happens:

- The instance performs a normal shutdown and stops running; its status changes to `stopping` and then `stopped`.
- Any Amazon EBS volumes remain attached to the instance, and their data persists.
- Any data stored in the RAM of the host computer or the instance store volumes of the host computer is gone.
- In most cases, the instance is migrated to a new underlying host computer when it's started (though in some cases, it remains on the current host).
- The instance retains its private IPv4 addresses and any IPv6 addresses when stopped and started. We release the public IPv4 address and assign a new one when you start it.
- The instance retains its associated Elastic IP addresses. You're charged for any Elastic IP addresses associated with a stopped instance. With EC2-Classic, an Elastic IP address is dissociated from your instance when you stop it. For more information, see [EC2-Classic \(p. 884\)](#).
- When you stop and start a Windows instance, the EC2Config service performs tasks on the instance, such as changing the drive letters for any attached Amazon EBS volumes. For more information about these defaults and how you can change them, see [Configuring a Windows instance using the EC2Config service](#) in the *Amazon EC2 User Guide for Windows Instances*.
- If your instance is in an Auto Scaling group, the Amazon EC2 Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. For more information, see [Health Checks for Auto Scaling Instances](#) in the *Amazon EC2 Auto Scaling User Guide*.

- When you stop a ClassicLink instance, it's unlinked from the VPC to which it was linked. You must link the instance to the VPC again after starting it. For more information about ClassicLink, see [ClassicLink \(p. 892\)](#).

For more information, see [Differences between reboot, stop, hibernate, and terminate \(p. 496\)](#).

You can modify the following attributes of an instance only when it is stopped:

- Instance type
- User data
- Kernel
- RAM disk

If you try to modify these attributes while the instance is running, Amazon EC2 returns the `IncorrectInstanceState` error.

What happens when you stop an instance

When an EC2 instance is stopped using the `stop-instances` command, the following is registered at the OS level:

- The API request will send a button press event to the guest.
- Various system services will be stopped as a result of the button press event. `systemd` handles a graceful shutdown of the system. Graceful shutdown is triggered by the ACPI shutdown button press event from the hypervisor.
- ACPI shutdown will be initiated.
- The instance will shut down when the graceful shutdown process exits. There is no configurable OS shutdown time.
- If the instance OS does not shut down cleanly within a few minutes, a hard shutdown is performed.

Stopping and starting your instances

You can stop and start your Amazon EBS-backed instance using the console or the command line.

By default, when you initiate a shutdown from an Amazon EBS-backed instance (using the `shutdown` or `poweroff` command), the instance stops. You can change this behavior so that it terminates instead. For more information, see [Changing the instance initiated shutdown behavior \(p. 610\)](#).

Using the `halt` command from an instance does not initiate a shutdown. If used, the instance does not terminate; instead, it places the CPU into `HLT` and the instance remains running.

New console

To stop and start an Amazon EBS-backed instance using the console

1. When you stop an instance, the data on any instance store volumes is erased. Before you stop an instance, verify that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.
2. In the navigation pane, choose **Instances** and select the instance.
3. Choose **Actions, Instance state, Stop instance**. If this option is disabled, either the instance is already stopped or its root device is an instance store volume.
4. When prompted for confirmation, choose **Stop**. It can take a few minutes for the instance to stop.

5. (Optional) While your instance is stopped, you can modify certain instance attributes. For more information, see [Modifying a stopped instance \(p. 590\)](#).
6. To start the stopped instance, select the instance, and choose **Actions, Instance state, Start instance**.
7. It can take a few minutes for the instance to enter the `running` state.

Old console

To stop and start an Amazon EBS-backed instance using the console

1. When you stop an instance, the data on any instance store volumes is erased. Before you stop an instance, verify that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.
2. In the navigation pane, choose **Instances** and select the instance.
3. Choose **Actions, Instance State, Stop**. If this option is disabled, either the instance is already stopped or its root device is an instance store volume.
4. When prompted for confirmation, choose **Yes, Stop**. It can take a few minutes for the instance to stop.
5. (Optional) While your instance is stopped, you can modify certain instance attributes. For more information, see [Modifying a stopped instance \(p. 590\)](#).
6. To start the stopped instance, select the instance, and choose **Actions, Instance State, Start**.
7. In the confirmation dialog box, choose **Yes, Start**. It can take a few minutes for the instance to enter the `running` state.

To stop and start an Amazon EBS-backed instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `stop-instances` and `start-instances` (AWS CLI)
- `Stop-EC2Instance` and `Start-EC2Instance` (AWS Tools for Windows PowerShell)

Modifying a stopped instance

You can change the instance type, user data, and EBS-optimization attributes of a stopped instance using the AWS Management Console or the command line interface. You can't use the AWS Management Console to modify the `DeleteOnTermination`, kernel, or RAM disk attributes.

To modify an instance attribute

- To change the instance type, see [Changing the instance type \(p. 294\)](#).
- To change the user data for your instance, see [Working with instance user data \(p. 672\)](#).
- To enable or disable EBS-optimization for your instance, see [Modifying EBS-Optimization \(p. 1156\)](#).
- To change the `DeleteOnTermination` attribute of the root volume for your instance, see [Updating the block device mapping of a running instance \(p. 1219\)](#). You are not required to stop the instance to change this attribute.

To modify an instance attribute using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Troubleshooting

If you have stopped your Amazon EBS-backed instance and it appears "stuck" in the stopping state, you can forcibly stop it. For more information, see [Troubleshooting stopping your instance \(p. 1254\)](#).

Hibernate your Linux instance

When you hibernate an instance, Amazon EC2 signals the operating system to perform hibernation (suspend-to-disk). Hibernation saves the contents from the instance memory (RAM) to your Amazon Elastic Block Store (Amazon EBS) root volume. Amazon EC2 persists the instance's EBS root volume and any attached EBS data volumes. When you start your instance:

- The EBS root volume is restored to its previous state
- The RAM contents are reloaded
- The processes that were previously running on the instance are resumed
- Previously attached data volumes are reattached and the instance retains its instance ID

You can hibernate an instance only if it's [enabled for hibernation \(p. 596\)](#) and it meets the [hibernation prerequisites \(p. 592\)](#).

If an instance or application takes a long time to bootstrap and build a memory footprint to become fully productive, you can use hibernation to pre-warm the instance. To pre-warm the instance, you:

1. Launch it with hibernation enabled.
2. Bring it to a desired state.
3. Hibernate it, ready to be resumed to the same state as needed.

You're not charged for instance usage for a hibernated instance when it is in the stopped state. You are charged for instance usage while the instance is in the stopping state, when the contents of the RAM are transferred to the EBS root volume. (This is different from when you [stop an instance \(p. 587\)](#) without hibernating it.) You're not charged for data transfer. However, you are charged for storage of any EBS volumes, including storage for the RAM contents.

If you no longer need an instance, you can terminate it at any time, including when it is in a stopped (hibernated) state. For more information, see [Terminate your instance \(p. 607\)](#).

Note

For information about using hibernation on Windows instances, see [Hibernate Your Windows Instance](#) in the *Amazon EC2 User Guide for Windows Instances*.

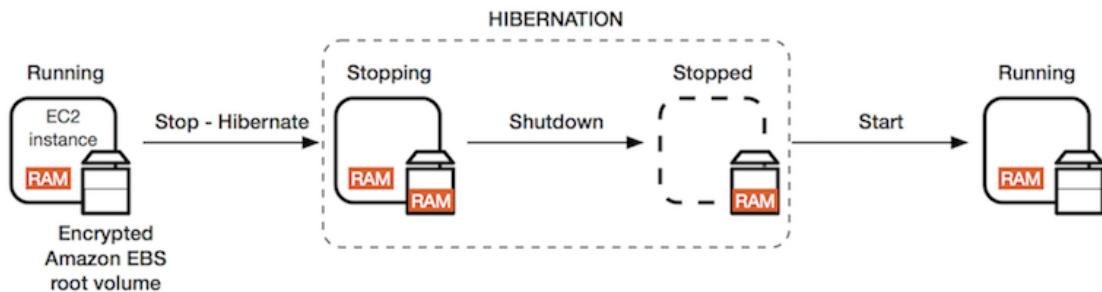
Contents

- [Overview of hibernation \(p. 592\)](#)
- [Hibernation prerequisites \(p. 592\)](#)
- [Limitations \(p. 593\)](#)
- [Configuring an existing AMI to support hibernation \(p. 594\)](#)
- [Enabling hibernation for an instance \(p. 596\)](#)
- [Disabling KASLR on an instance \(Ubuntu only\) \(p. 599\)](#)
- [Hibernating an instance \(p. 599\)](#)
- [Starting a hibernated instance \(p. 601\)](#)

- [Troubleshooting hibernation \(p. 602\)](#)

Overview of hibernation

The following diagram shows a basic overview of the hibernation process.



When you hibernate a running instance, the following happens:

- When you initiate hibernation, the instance moves to the stopping state. Amazon EC2 signals the operating system to perform hibernation (suspend-to-disk). The hibernation freezes all of the processes, saves the contents of the RAM to the EBS root volume, and then performs a regular shutdown.
- After the shutdown is complete, the instance moves to the stopped state.
- Any EBS volumes remain attached to the instance, and their data persists, including the saved contents of the RAM.
- In most cases, the instance is migrated to a new underlying host computer when it's started. This is also what happens when you stop and start an instance.
- When you start the instance, the instance boots up and the operating system reads in the contents of the RAM from the EBS root volume, before unfreezing processes to resume its state.
- The instance retains its private IPv4 addresses and any IPv6 addresses. When you start the instance, the instance continues to retain its private IPv4 addresses and any IPv6 addresses.
- Amazon EC2 releases the public IPv4 address. When you start the instance, Amazon EC2 assigns a new public IPv4 address to the instance.
- The instance retains its associated Elastic IP addresses. You're charged for any Elastic IP addresses associated with a hibernated instance. With EC2-Classic, an Elastic IP address is disassociated from your instance when you hibernate it. For more information, see [EC2-Classic \(p. 884\)](#).
- When you hibernate a ClassicLink instance, it's unlinked from the VPC to which it was linked. You must link the instance to the VPC again after starting it. For more information, see [ClassicLink \(p. 892\)](#).

For information about how hibernation differs from reboot, stop, and terminate, see [Differences between reboot, stop, hibernate, and terminate \(p. 496\)](#).

Hibernation prerequisites

To hibernate an instance, the following prerequisites must be in place:

- **Supported instance families** – C3, C4, C5, M3, M4, M5, M5a, R3, R4, R5, R5a, and T2.
- **Instance RAM size** – must be less than 150 GB.
- **Instance size** – not supported for bare metal instances.
- **Supported AMIs** (must be an HVM AMI that supports hibernation):
 - Amazon Linux 2 AMI released 2019.08.29 or later.

- Amazon Linux AMI 2018.03 released 2018.11.16 or later.
- Ubuntu 18.04 LTS - Bionic AMI released with serial number 20190722.1 or later.*
- Ubuntu 16.04 LTS - Xenial AMI.* ([Additional configuration](#) is required.)

*We recommend disabling KASLR on instances with Ubuntu 18.04 LTS - Bionic and Ubuntu 16.04 LTS - Xenial. For more information, see [Disabling KASLR on an instance \(Ubuntu only\) \(p. 599\)](#).

To configure your own AMI to support hibernation, see [Configuring an existing AMI to support hibernation \(p. 594\)](#).

Support for other versions of Ubuntu and other operating systems is coming soon.

For information about the supported AMIs for Windows, see [Hibernation prerequisites](#) in the *Amazon EC2 User Guide for Windows Instances*.

- **Root volume type** - must be an EBS volume, not an instance store volume.
- **Supported EBS volume types** - General Purpose SSD (gp2) or Provisioned IOPS SSD (io1 or io2). If you choose a Provisioned IOPS SSD (io1 or io2) volume type, to achieve optimum performance for hibernation, you must provision the EBS volume with the appropriate IOPS. For more information, see [Amazon EBS volume types \(p. 1023\)](#).
- **EBS root volume size** - must be large enough to store the RAM contents and accommodate your expected usage, for example, OS or applications. If you enable hibernation, space is allocated on the root volume at launch to store the RAM.
- **EBS root volume encryption** - To use hibernation, the root volume must be encrypted to ensure the protection of sensitive content that is in memory at the time of hibernation. When RAM data is moved to the EBS root volume, it is always encrypted. Encryption of the root volume is enforced at instance launch. Use one of the following three options to ensure that the root volume is an encrypted EBS volume:
 - EBS “single-step” encryption: You can launch encrypted EBS-backed EC2 instances from an unencrypted AMI and also enable hibernation at the same time. For more information, see [Using encryption with EBS-backed AMIs \(p. 157\)](#).
 - EBS encryption by default: You can enable EBS encryption by default to ensure all new EBS volumes created in your AWS account are encrypted. This way, you can enable hibernation for your instances without specifying encryption intent at instance launch. For more information, see [Encryption by default \(p. 1124\)](#).
 - Encrypted AMI: You can enable EBS encryption by using an encrypted AMI to launch your instance. If your AMI does not have an encrypted root snapshot, you can copy it to a new AMI and request encryption. For more information, see [Encrypt an unencrypted image during copy \(p. 162\)](#) and [Copying an AMI \(p. 167\)](#).
- **Enable hibernation at launch** - You cannot enable hibernation on an existing instance (running or stopped). For more information, see [Enabling hibernation for an instance \(p. 596\)](#).
- **Purchasing options** - This feature is available for On-Demand Instances and Reserved Instances. It is not available for Spot Instances. For more information, see [Hibernating interrupted Spot Instances \(p. 428\)](#).

Limitations

- The following actions are not supported for hibernation:
 - Changing the instance type or size of a hibernated instance
 - Creating snapshots or AMIs from instances for which hibernation is enabled
 - Creating snapshots or AMIs from hibernated instances
- You can't stop or hibernate instance store-backed instances.*
- You can't hibernate an instance that has more than 150 GB of RAM.

- You cannot hibernate an instance that is in an Auto Scaling group or used by Amazon ECS. If your instance is in an Auto Scaling group and you try to hibernate it, the Amazon EC2 Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. For more information, see [Health Checks for Auto Scaling Instances](#) in the *Amazon EC2 Auto Scaling User Guide*.
- We do not support keeping an instance hibernated for more than 60 days. To keep the instance for longer than 60 days, you must start the hibernated instance, stop the instance, and start it.
- We constantly update our platform with upgrades and security patches, which can conflict with existing hibernated instances. We notify you about critical updates that require a start for hibernated instances so that we can perform a shutdown or a reboot to apply the necessary upgrades and security patches.

*For C3 and R3 instances that are enabled for hibernation, do not use instance store volumes.

Configuring an existing AMI to support hibernation

To hibernate an instance that was launched using your own AMI, you must first configure your AMI to support hibernation. For more information, see [Updating instance software on your Amazon Linux instance \(p. 615\)](#).

If you use one of the [supported AMIs \(p. 592\)](#) (except Ubuntu 16.04 LTS), or if you create an AMI based on one of the supported AMIs, you do not need to configure it to support hibernation. These AMIs are preconfigured to support hibernation. To configure Ubuntu 16.04 LTS to support hibernation, you need to install the `linux-aws-hwe` kernel package version 4.15.0-1058-aws or later and the `ec2-hibinit-agent`. For the configuration steps, choose the **Ubuntu 16.04 - Xenial** tab below.

Amazon Linux 2

To configure an Amazon Linux 2 AMI to support hibernation

1. Update to the latest kernel to 4.14.138-114.102 or later using the following command.

```
[ec2-user ~]$ sudo yum update kernel
```

2. Install the `ec2-hibinit-agent` package from the repositories using the following command.

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

3. Reboot the instance using the following command.

```
[ec2-user ~]$ sudo reboot
```

4. Confirm that the kernel version is updated to 4.14.138-114.102 or later using the following command.

```
[ec2-user ~]$ uname -a
```

5. Stop the instance and create an AMI. For more information, see [Creating a Linux AMI from an instance \(p. 124\)](#).

Amazon Linux

To configure an Amazon Linux AMI to support hibernation

1. Update to the latest kernel to 4.14.77-70.59 or later using the following command.

```
[ec2-user ~]$ sudo yum update kernel
```

2. Install the ec2-hibinit-agent package from the repositories using the following command.

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

3. Reboot the instance using the following command.

```
[ec2-user ~]$ sudo reboot
```

4. Confirm that the kernel version is updated to 4.14.77-70.59 or greater using the following command.

```
[ec2-user ~]$ uname -a
```

5. Stop the instance and create an AMI. For more information, see [Creating a Linux AMI from an instance \(p. 124\)](#).

Ubuntu 18.04 - Bionic

To configure an Ubuntu 18.04 LTS AMI to support hibernation

1. Update to the latest kernel to 4.15.0-1044 or later using the following commands.

```
[ec2-user ~]$ sudo apt update  
[ec2-user ~]$ sudo apt dist-upgrade
```

2. Install the ec2-hibinit-agent package from the repositories using the following command.

```
[ec2-user ~]$ sudo apt install ec2-hibinit-agent
```

3. Reboot the instance using the following command.

```
[ec2-user ~]$ sudo reboot
```

4. Confirm that the kernel version is updated to 4.15.0-1044 or later using the following command.

```
[ec2-user ~]$ uname -a
```

Ubuntu 16.04 - Xenial

To configure an Ubuntu 16.04 LTS AMI to support hibernation

1. Update to the latest kernel to 4.15.0-1058-aws or later using the following commands.

```
[ec2-user ~]$ sudo apt update  
[ec2-user ~]$ sudo apt install linux-aws-hwe
```

Note

The linux-aws-hwe kernel package is fully supported by Canonical. The package will continue to receive regular updates until standard support for Ubuntu 16.04 LTS ends in April 2021, and will receive additional security updates until the Extended Security

Maintenance support ends in 2024. For more information, see [Amazon EC2 Hibernation for Ubuntu 16.04 LTS now available](#) on the Canonical Ubuntu Blog.

2. Install the `ec2-hibinit-agent` package from the repositories using the following command.

```
[ec2-user ~]$ sudo apt install ec2-hibinit-agent
```

3. Reboot the instance using the following command.

```
[ec2-user ~]$ sudo reboot
```

4. Confirm that the kernel version is updated to 4.15.0-1058-aws or later using the following command.

```
[ec2-user ~]$ uname -a
```

Enabling hibernation for an instance

To hibernate an instance, it must first be enabled for hibernation. To enable hibernation, you must do it while launching the instance.

Important

You can't enable or disable hibernation for an instance after you launch it.

Console

To enable hibernation using the console

1. Follow the [Launching an instance using the Launch Instance Wizard \(p. 499\)](#) procedure.
2. On the **Choose an Amazon Machine Image (AMI)** page, select an AMI that supports hibernation. For more information about supported AMIs, see [Hibernation prerequisites \(p. 592\)](#).
3. On the **Choose an Instance Type** page, select a supported instance type, and choose **Next: Configure Instance Details**. For information about supported instance types, see [Hibernation prerequisites \(p. 592\)](#).
4. On the **Configure Instance Details** page, for **Stop - Hibernate Behavior**, select the **Enable hibernation as an additional stop behavior** check box.
5. On the **Add Storage** page, for the root volume, specify the following information:
 - For **Size (GiB)**, enter the EBS root volume size. The volume must be large enough to store the RAM contents and accommodate your expected usage.
 - For **Volume Type**, select a supported EBS volume type (General Purpose SSD (gp2) or Provisioned IOPS SSD (io1 or io2)).
 - For **Encryption**, select the encryption key for the volume. If you enabled encryption by default in this AWS Region, the default encryption key is selected.

For more information about the prerequisites for the root volume, see [Hibernation prerequisites \(p. 592\)](#).

6. Continue as prompted by the wizard. When you've finished reviewing your options on the **Review Instance Launch** page, choose **Launch**. For more information, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#).

AWS CLI

To enable hibernation using the AWS CLI

Use the [run-instances](#) command to launch an instance. Specify the EBS root volume parameters using the `--block-device-mappings file://mapping.json` parameter, and enable hibernation using the `--hibernation-options Configured=true` parameter.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type m5.large \
    --block-device-mappings file://mapping.json \
    --hibernation-options Configured=true \
    --count 1 \
    --key-name MyKeyPair
```

Specify the following in `mapping.json`:

```
[ \
  { \
    "DeviceName": "/dev/xvda", \
    "Ebs": { \
      "VolumeSize": 30, \
      "VolumeType": "gp2", \
      "Encrypted": true \
    } \
  } \
]
```

Note

The value for `DeviceName` must match the root device name associated with the AMI. To find the root device name, use the [describe-images](#) command, as follows:

```
aws ec2 describe-images --image-id ami-0abcdef1234567890
```

If you enabled encryption by default in this AWS Region, you can omit `"Encrypted": true`.

PowerShell

To enable hibernation using the AWS Tools for Windows PowerShell

Use the [New-EC2Instance](#) command to launch an instance. Specify the EBS root volume by first defining the block device mapping, and then adding it to the command using the `-BlockDeviceMappings` parameter. Enable hibernation using the `-HibernationOptions_Configured $true` parameter.

```
PS C:\> $ebs_encrypt = New-Object Amazon.EC2.Model.BlockDeviceMapping
PS C:\> $ebs_encrypt.DeviceName = "/dev/xvda"
PS C:\> $ebs_encrypt.Ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
PS C:\> $ebs_encrypt.Ebs.VolumeSize = 30
PS C:\> $ebs_encrypt.Ebs.VolumeType = "gp2"
PS C:\> $ebs_encrypt.Ebs.Encrypted = $true

PS C:\> New-EC2Instance ` 
        -ImageId ami-0abcdef1234567890 ` 
        -InstanceType m5.large ` 
        -BlockDeviceMappings $ebs_encrypt ` 
        -HibernationOptions_Configured $true ` 
        -MinCount 1 ` 
        -MaxCount 1
```

-KeyName **MyKeyPair**

Note

The value for **DeviceName** must match the root device name associated with the AMI. To find the root device name, use the [Get-EC2Image](#) command, as follows:

```
Get-EC2Image -ImageId ami-Oabcdef1234567890
```

If you enabled encryption by default in this AWS Region, you can omit **Encrypted = \$true** from the block device mapping.

New console

To view if an instance is enabled for hibernation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and, on the **Details** tab, in the **Instance details** section, inspect **Stop-hibernate behavior. Enabled**. **Enabled** indicates that the instance is enabled for hibernation.

Old console

To view if an instance is enabled for hibernation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and, in the details pane, inspect **Stop - Hibernation behavior. Enabled**. **Enabled** indicates that the instance is enabled for hibernation.

AWS CLI

To view if an instance is enabled for hibernation using the AWS CLI

Use the [describe-instances](#) command and specify the `--filters "Name=hibernation-options.configured,Values=true"` parameter to filter instances that are enabled for hibernation.

```
aws ec2 describe-instances \
  --filters "Name=hibernation-options.configured,Values=true"
```

The following field in the output indicates that the instance is enabled for hibernation.

```
"HibernationOptions": {
    "Configured": true
}
```

PowerShell

To view if an instance is enabled for hibernation using the AWS Tools for Windows PowerShell

Use the [Get-EC2Instance](#) command and specify the `-Filter @{ Name="hibernation-options.configured"; Value="true" }` parameter to filter instances that are enabled for hibernation.

```
Get-EC2Instance ^  
    -Filter @{ Name="hibernation-options.configured"; Value="true" }
```

The output lists the EC2 instances that are enabled for hibernation.

Disabling KASLR on an instance (Ubuntu only)

To run hibernation on a newly launched instance with Ubuntu 16.04 LTS - Xenial or Ubuntu 18.04 LTS - Bionic released with serial 20190722.1 or later, we recommend disabling KASLR (Kernel Address Space Layout Randomization). On Ubuntu 16.04 LTS or Ubuntu 18.04 LTS, KASLR is enabled by default. KASLR is a standard Linux kernel security feature that helps to mitigate exposure to and ramifications of yet-undiscovered memory access vulnerabilities by randomizing the base address value of the kernel. With KASLR enabled, there is a possibility that the instance might not resume after it has been hibernated.

To learn more about KASLR, see [Ubuntu Features](#).

To disable KASLR on an instance launched with Ubuntu

1. Connect to your instance using SSH. For more information, see [Connecting to your Linux instance using SSH \(p. 566\)](#).
2. Open the `/etc/default/grub.d/50-cloudimg-settings.cfg` file in your editor of choice. Edit the `GRUB_CMDLINE_LINUX_DEFAULT` line to append the `nokaslr` option to its end, as shown in the following example.

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0 nvme_core.io_timeout=4294967295  
nokaslr"
```

3. Save the file and exit your editor.
4. Run the following command to rebuild the grub configuration.

```
[ec2-user ~]$ sudo update-grub
```

5. Reboot the instance.

```
[ec2-user ~]$ sudo reboot
```

6. Confirm that `nokaslr` has been added when running the following command.

```
[ec2-user ~]$ cat /proc/cmdline
```

The output of the command should include the `nokaslr` option.

Hibernating an instance

You can hibernate an instance if the instance is [enabled for hibernation \(p. 596\)](#) and meets the [hibernation prerequisites \(p. 592\)](#). If an instance cannot hibernate successfully, a normal shutdown occurs.

New console

To hibernate an Amazon EBS-backed instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Instances**.
3. Select an instance, and choose **Actions**, **Instance state**, **Hibernate instance**. If **Hibernate instance** is disabled, the instance is already hibernated or stopped, or it can't be hibernated. For more information, see [Hibernate prerequisites \(p. 592\)](#).
4. In the confirmation dialog box, choose **Hibernate**. It can take a few minutes for the instance to hibernate. The **Instance state** changes to **Stopping** while the instance is hibernating, and then **Stopped** when the instance has hibernated.

Old console

To hibernate an Amazon EBS-backed instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select an instance, and choose **Actions**, **Instance State**, **Stop - Hibernate**. If **Stop - Hibernate** is disabled, the instance is already hibernated or stopped, or it can't be hibernated. For more information, see [Hibernate prerequisites \(p. 592\)](#).
4. In the confirmation dialog box, choose **Yes, Stop - Hibernate**. It can take a few minutes for the instance to hibernate. The **Instance State** changes to **Stopping** while the instance is hibernating, and then **Stopped** when the instance has hibernated.

AWS CLI

To hibernate an Amazon EBS-backed instance using the AWS CLI

Use the [stop-instances](#) command and specify the `--hibernate` parameter.

```
aws ec2 stop-instances \
--instance-ids i-1234567890abcdef0 \
--hibernate
```

PowerShell

To hibernate an Amazon EBS-backed instance using the AWS Tools for Windows PowerShell

Use the [Stop-EC2Instance](#) command and specify the `-Hibernate $true` parameter.

```
Stop-EC2Instance ^
-InstanceId i-1234567890abcdef0 ^
-Hibernate $true
```

New console

To view if hibernation was initiated on an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and, on the **Details** tab, in the **Instance details** section, inspect **State transition message**. The message **Client.UserInitiatedHibernate: User initiated hibernate** indicates that hibernation was initiated on the instance.

Old console

To view if hibernation was initiated on an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and, in the details pane, inspect **State transition reason message**. The message **Client.UserInitiatedHibernate: User initiated hibernate** indicates that hibernation was initiated on the instance.

AWS CLI

To view if hibernation was initiated on an instance using the AWS CLI

Use the `describe-instances` command and specify the `state-reason-code` filter to see the instances on which hibernation was initiated.

```
aws ec2 describe-instances \
--filters "Name=state-reason-code,Values=Client.UserInitiatedHibernate"
```

The following field in the output indicates that hibernation was initiated on the instance.

```
"StateReason": {
    "Code": "Client.UserInitiatedHibernate"
}
```

PowerShell

To view if hibernation was initiated on an instance using the AWS Tools for Windows PowerShell

Use the `Get-EC2Instance` command and specify the `state-reason-code` filter to see the instances on which hibernation was initiated.

```
Get-EC2Instance ^
-Filter @{Name="state-reason-code";Value="Client.UserInitiatedHibernate"}
```

The output lists the EC2 instances on which hibernation was initiated.

Starting a hibernated instance

Start a hibernated instance by starting it in the same way that you would start a stopped instance.

New console

To start a hibernated instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select a hibernated instance, and choose **Actions, Instance state, Start instance**. It can take a few minutes for the instance to enter the `running` state. During this time, the instance [status checks \(p. 698\)](#) show the instance in a failed state until the instance has started.

Old console

To start a hibernated instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select a hibernated instance, and choose **Actions**, **Instance State**, **Start**. It can take a few minutes for the instance to enter the **running** state. During this time, the instance [status checks \(p. 698\)](#) show the instance in a failed state until the instance has started.

AWS CLI

To start a hibernated instance using the AWS CLI

Use the [start-instances](#) command.

```
aws ec2 start-instances \
--instance-ids i-1234567890abcdef0
```

PowerShell

To start a hibernated instance using the AWS Tools for Windows PowerShell

Use the [Start-EC2Instance](#) command.

```
Start-EC2Instance ^
-InstanceId i-1234567890abcdef0
```

Troubleshooting hibernation

Use this information to help diagnose and fix issues that you might encounter when hibernating an instance.

Can't hibernate immediately after launch

If you try to hibernate an instance too quickly after you've launched it, you get an error.

You must wait for about two minutes after launch before hibernating.

Takes too long to transition from stopping to stopped, and memory state not restored after start

If it takes a long time for your hibernating instance to transition from the **stopping** state to **stopped**, and if the memory state is not restored after you start, this could indicate that hibernation was not properly configured.

Check the instance system log and look for messages that are related to hibernation. To access the system log, [connect \(p. 563\)](#) to the instance or use the [get-console-output](#) command. Find the log lines from the `hibinit-agent`. If the log lines indicate a failure or the log lines are missing, there was most likely a failure configuring hibernation at launch.

For example, the following message indicates that the instance root volume is not large enough:
`hibinit-agent: Insufficient disk space. Cannot create setup for hibernation.`
`Please allocate a larger root device.`

If the last log line from the `hibinit-agent` is `hibinit-agent: Running: swapoff /swap`, hibernation was successfully configured.

If you do not see any logs from these processes, your AMI might not support hibernation. For information about supported AMIs, see [Hibernation prerequisites \(p. 592\)](#). If you used your own AMI, make sure that you followed the instructions for [Configuring an existing AMI to support hibernation \(p. 594\)](#).

Instance "stuck" in the stopping state

If you hibernated your instance and it appears "stuck" in the stopping state, you can forcibly stop it. For more information, see [Troubleshooting stopping your instance \(p. 1254\)](#).

Reboot your instance

An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance. When you reboot an instance, it keeps its public DNS name (IPv4), private IPv4 address, IPv6 address (if applicable), and any data on its instance store volumes.

Rebooting an instance doesn't start a new instance billing period (with a minimum one-minute charge), unlike stopping and starting your instance.

We might schedule your instance for a reboot for necessary maintenance, such as to apply updates that require a reboot. No action is required on your part; we recommend that you wait for the reboot to occur within its scheduled window. For more information, see [Scheduled events for your instances \(p. 704\)](#).

We recommend that you use the Amazon EC2 console, a command line tool, or the Amazon EC2 API to reboot your instance instead of running the operating system reboot command from your instance. If you use the Amazon EC2 console, a command line tool, or the Amazon EC2 API to reboot your instance, we perform a hard reboot if the instance does not cleanly shut down within a few minutes. If you use AWS CloudTrail, then using Amazon EC2 to reboot your instance also creates an API record of when your instance was rebooted.

New console

To reboot an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, Instance state, Reboot instance**.
4. Choose **Reboot** when prompted for confirmation. The instance remains in the running state.

Old console

To reboot an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, Instance State, Reboot**.

4. Choose **Yes, Reboot** when prompted for confirmation. The instance remains in the running state.

To reboot an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [reboot-instances \(AWS CLI\)](#)
- [Restart-EC2Instance \(AWS Tools for Windows PowerShell\)](#)

Instance retirement

An instance is scheduled to be retired when AWS detects irreparable failure of the underlying hardware that hosts the instance. When an instance reaches its scheduled retirement date, it is stopped or terminated by AWS.

- If your instance root device is an Amazon EBS volume, the instance is stopped, and you can start it again at any time. Starting the stopped instance migrates it to new hardware.
- If your instance root device is an instance store volume, the instance is terminated, and cannot be used again.

For more information about the types of instance events, see [Scheduled events for your instances \(p. 704\)](#).

Contents

- [Identifying instances scheduled for retirement \(p. 604\)](#)
- [Actions to take for EBS-backed instances scheduled for retirement \(p. 605\)](#)
- [Actions to take for instance-store backed instances scheduled for retirement \(p. 606\)](#)

Identifying instances scheduled for retirement

If your instance is scheduled for retirement, you receive an email prior to the event with the instance ID and retirement date. You can also check for instances that are scheduled for retirement using the Amazon EC2 console or the command line.

Important

If an instance is scheduled for retirement, we recommend that you take action as soon as possible because the instance might be unreachable. (The email notification you receive states the following: "Due to this degradation your instance could already be unreachable.") For more information about the recommended action you should take, see [Check if your instance is reachable](#).

Ways to identify instances scheduled for retirement

- [Email notification \(p. 604\)](#)
- [Console identification \(p. 605\)](#)

Email notification

If your instance is scheduled for retirement, you receive an email prior to the event with the instance ID and retirement date.

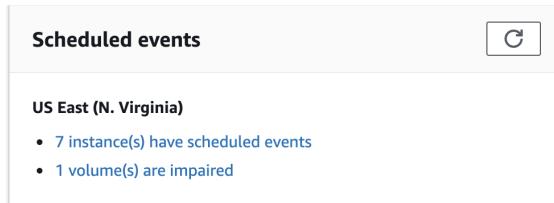
The email is sent to the address that's associated with your account. It's the same email address that you use to log in to the AWS Management Console. To update the contact information for your account, go to the [Account Settings](#) page.

Console identification

If you use an email account that you do not check regularly for instance retirement notifications, you can use the Amazon EC2 console or the command line to determine if any of your instances are scheduled for retirement.

To identify instances scheduled for retirement using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **EC2 Dashboard**. Under **Scheduled events**, you can see the events that are associated with your Amazon EC2 instances and volumes, organized by Region.



3. If you have an instance with a scheduled event listed, select its link below the Region name to go to the **Events** page.
4. The **Events** page lists all resources that have events associated with them. To view instances that are scheduled for retirement, select **Instance resources** from the first filter list, and then **Instance stop or retirement** from the second filter list.
5. If the filter results show that an instance is scheduled for retirement, select it, and note the date and time in the **Start time** field in the details pane. This is your instance retirement date.

To identify instances scheduled for retirement using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-instance-status](#) (AWS CLI)
- [Get-EC2InstanceState](#) (AWS Tools for Windows PowerShell)

Actions to take for EBS-backed instances scheduled for retirement

To preserve the data on your retiring instance, you can perform one of the following actions. It's important that you take this action before the instance retirement date to prevent unforeseen downtime and data loss.

If you are not sure whether your instance is backed by EBS or instance store, see [Determining the root device type of your instance \(p. 22\)](#).

Check if your instance is reachable

When you are notified that your instance is scheduled for retirement, we recommend that you take the following action as soon as possible:

- Check if your instance is reachable by either [connecting \(p. 563\)](#) to or pinging your instance.

- If your instance is reachable, you should plan to stop/start your instance at an appropriate time before the scheduled retirement date, when the impact is minimal. For more information about stopping and starting your instance, and what to expect when your instance is stopped, such as the effect on public, private, and Elastic IP addresses that are associated with your instance, see [Stop and start your instance \(p. 587\)](#). Note that data on instance store volumes is lost when you stop and start your instance.
- If your instance is unreachable, you should take immediate action and perform a [stop/start \(p. 587\)](#) to recover your instance.
- Alternatively, if you want to [terminate \(p. 607\)](#) your instance, plan to do so as soon as possible so that you stop incurring charges for the instance.

Create a backup of your instance

Create an EBS-backed AMI from your instance so that you have a backup. To ensure data integrity, stop the instance before you create the AMI. You can wait for the scheduled retirement date when the instance is stopped, or stop the instance yourself before the retirement date. You can start the instance again at any time. For more information, see [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#).

Launch a replacement instance

After you create an AMI from your instance, you can use the AMI to launch a replacement instance. From the Amazon EC2 console, select your new AMI and then choose **Actions, Launch**. Follow the wizard to launch your instance. For more information about each step in the wizard, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#).

Actions to take for instance-store backed instances scheduled for retirement

To preserve the data on your retiring instance, you can perform one of the following actions. It's important that you take this action before the instance retirement date to prevent unforeseen downtime and data loss.

Warning

If your instance store-backed instance passes its retirement date, it is terminated and you cannot recover the instance or any data that was stored on it. Regardless of the root device of your instance, the data on instance store volumes is lost when the instance is retired, even if the volumes are attached to an EBS-backed instance.

Check if your instance is reachable

When you are notified that your instance is scheduled for retirement, we recommend that you take the following action as soon as possible:

- Check if your instance is reachable by either [connecting \(p. 563\)](#) to or pinging your instance.
- If your instance is unreachable, there is likely very little that can be done to recover your instance. For more information, see [Troubleshooting an unreachable instance \(p. 1277\)](#). AWS will terminate your instance on the scheduled retirement date, so, for an unreachable instance, you can immediately [terminate \(p. 607\)](#) the instance yourself.

Launch a replacement instance

Create an instance store-backed AMI from your instance using the AMI tools, as described in [Creating an instance store-backed Linux AMI \(p. 127\)](#). From the Amazon EC2 console, select your new AMI and then choose **Actions, Launch**. Follow the wizard to launch your instance. For more information about each step in the wizard, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#).

Convert your instance to an EBS-backed instance

Transfer your data to an EBS volume, take a snapshot of the volume, and then create AMI from the snapshot. You can launch a replacement instance from your new AMI. For more information, see [Converting your instance store-backed AMI to an Amazon EBS-backed AMI \(p. 138\)](#).

Terminate your instance

You can delete your instance when you no longer need it. This is referred to as *terminating* your instance. As soon as the state of an instance changes to `shutting-down` or `terminated`, you stop incurring charges for that instance.

You can't connect to or start an instance after you've terminated it. However, you can launch additional instances using the same AMI. If you'd rather stop and start your instance, or hibernate it, see [Stop and start your instance \(p. 587\)](#) or [Hibernate your Linux instance \(p. 591\)](#). For more information, see [Differences between reboot, stop, hibernate, and terminate \(p. 496\)](#).

Contents

- [Instance termination \(p. 607\)](#)
- [What happens when you terminate an instance \(p. 608\)](#)
- [Terminating an instance \(p. 608\)](#)
- [Enabling termination protection \(p. 609\)](#)
- [Changing the instance initiated shutdown behavior \(p. 610\)](#)
- [Preserving Amazon EBS volumes on instance termination \(p. 610\)](#)
- [Troubleshooting \(p. 612\)](#)

Instance termination

After you terminate an instance, it remains visible in the console for a short while, and then the entry is automatically deleted. You cannot delete the terminated instance entry yourself. After an instance is terminated, resources such as tags and volumes are gradually disassociated from the instance and may no longer be visible on the terminated instance after a short while.

When an instance terminates, the data on any instance store volumes associated with that instance is deleted.

By default, Amazon EBS root device volumes are automatically deleted when the instance terminates. However, by default, any additional EBS volumes that you attach at launch, or any EBS volumes that you attach to an existing instance persist even after the instance terminates. This behavior is controlled by the volume's `DeleteOnTermination` attribute, which you can modify. For more information, see [Preserving Amazon EBS volumes on instance termination \(p. 610\)](#).

You can prevent an instance from being terminated accidentally by someone using the AWS Management Console, the CLI, and the API. This feature is available for both Amazon EC2 instance store-backed and Amazon EBS-backed instances. Each instance has a `DisableApiTermination` attribute with the default value of `false` (the instance can be terminated through Amazon EC2). You can modify this instance attribute while the instance is running or stopped (in the case of Amazon EBS-backed instances). For more information, see [Enabling termination protection \(p. 609\)](#).

You can control whether an instance should stop or terminate when shutdown is initiated from the instance using an operating system command for system shutdown. For more information, see [Changing the instance initiated shutdown behavior \(p. 610\)](#).

If you run a script on instance termination, your instance might have an abnormal termination, because we have no way to ensure that shutdown scripts run. Amazon EC2 attempts to shut an instance down cleanly and run any system shutdown scripts; however, certain events (such as hardware failure) may prevent these system shutdown scripts from running.

What happens when you terminate an instance

When an EC2 instance is terminated using the `terminate-instances` command, the following is registered at the OS level:

- The API request will send a button press event to the guest.
- Various system services will be stopped as a result of the button press event. `systemd` handles a graceful shutdown of the system. Graceful shutdown is triggered by the ACPI shutdown button press event from the hypervisor.
- ACPI shutdown will be initiated.
- The instance will shut down when the graceful shutdown process exits. There is no configurable OS shutdown time.

Terminating an instance

You can terminate an instance using the AWS Management Console or the command line.

By default, when you initiate a shutdown from an Amazon EBS-backed instance (using the `shutdown` or `poweroff` commands), the instance stops. The `halt` command does not initiate a shutdown. If used, the instance does not terminate; instead, it places the CPU into HALT and the instance remains running.

New console

To terminate an instance using the console

1. Before you terminate an instance, verify that you won't lose any data by checking that your Amazon EBS volumes won't be deleted on termination and that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.
2. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
3. In the navigation pane, choose **Instances**.
4. Select the instance, and choose **Actions**, **Instance state**, **Terminate instance**.
5. Choose **Terminate** when prompted for confirmation.

Old console

To terminate an instance using the console

1. Before you terminate an instance, verify that you won't lose any data by checking that your Amazon EBS volumes won't be deleted on termination and that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.
2. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
3. In the navigation pane, choose **Instances**.
4. Select the instance, and choose **Actions**, **Instance State**, **Terminate**.
5. Choose **Yes, Terminate** when prompted for confirmation.

To terminate an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [terminate-instances \(AWS CLI\)](#)
- [Stop-EC2Instance \(AWS Tools for Windows PowerShell\)](#)

Enabling termination protection

By default, you can terminate your instance using the Amazon EC2 console, command line interface, or API. To prevent your instance from being accidentally terminated using Amazon EC2, you can enable *termination protection* for the instance. The `DisableApiTermination` attribute controls whether the instance can be terminated using the console, CLI, or API. By default, termination protection is disabled for your instance. You can set the value of this attribute when you launch the instance, while the instance is running, or while the instance is stopped (for Amazon EBS-backed instances).

The `DisableApiTermination` attribute does not prevent you from terminating an instance by initiating shutdown from the instance (using an operating system command for system shutdown) when the `InstanceInitiatedShutdownBehavior` attribute is set. For more information, see [Changing the instance initiated shutdown behavior \(p. 610\)](#).

Limitations

You can't enable termination protection for Spot Instances—a Spot Instance is terminated when the Spot price exceeds the amount you're willing to pay for Spot Instances. However, you can prepare your application to handle Spot Instance interruptions. For more information, see [Spot Instance interruptions \(p. 426\)](#).

The `DisableApiTermination` attribute does not prevent Amazon EC2 Auto Scaling from terminating an instance. For instances in an Auto Scaling group, use the following Amazon EC2 Auto Scaling features instead of Amazon EC2 termination protection:

- To prevent instances that are part of an Auto Scaling group from terminating on scale in, use instance protection. For more information, see [Instance Protection](#) in the *Amazon EC2 Auto Scaling User Guide*.
- To prevent Amazon EC2 Auto Scaling from terminating unhealthy instances, suspend the `ReplaceUnhealthy` process. For more information, see [Suspending and Resuming Scaling Processes](#) in the *Amazon EC2 Auto Scaling User Guide*.
- To specify which instances Amazon EC2 Auto Scaling should terminate first, choose a termination policy. For more information, see [Customizing the Termination Policy](#) in the *Amazon EC2 Auto Scaling User Guide*.

To enable termination protection for an instance at launch time

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch Instance** and follow the directions in the wizard.
3. On the **Configure Instance Details** page, select the **Enable termination protection** check box.

To enable termination protection for a running or stopped instance

1. Select the instance, and choose **Actions, Instance Settings, Change Termination Protection**.
2. Choose **Yes, Enable**.

To disable termination protection for a running or stopped instance

1. Select the instance, and choose **Actions, Instance Settings, Change Termination Protection**.
2. Choose **Yes, Disable**.

To enable or disable termination protection using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Changing the instance initiated shutdown behavior

By default, when you initiate a shutdown from an Amazon EBS-backed instance (using a command such as **shutdown** or **poweroff**), the instance stops (Note that **halt** does not issue a **poweroff** command and, if used, the instance will not terminate; instead, it will place the CPU into HLT and the instance will remain running). You can change this behavior using the `InstanceInitiatedShutdownBehavior` attribute for the instance so that it terminates instead. You can update this attribute while the instance is running or stopped.

You can update the `InstanceInitiatedShutdownBehavior` attribute using the Amazon EC2 console or the command line. The `InstanceInitiatedShutdownBehavior` attribute only applies when you perform a shutdown from the operating system of the instance itself; it does not apply when you stop an instance using the `StopInstances` API or the Amazon EC2 console.

To change the shutdown behavior of an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. Choose **Actions**, **Instance settings**, **Change shutdown behavior**. The current behavior is selected.
5. To change the behavior, select **Stop** or **Terminate** from **Shutdown behavior** and then choose **Apply**.

To change the shutdown behavior of an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Preserving Amazon EBS volumes on instance termination

When an instance terminates, Amazon EC2 uses the value of the `DeleteOnTermination` attribute for each attached Amazon EBS volume to determine whether to preserve or delete the volume.

The default value for the `DeleteOnTermination` attribute differs depending on whether the volume is the root volume of the instance or a non-root volume attached to the instance.

Root volume

By default, the `DeleteOnTermination` attribute for the root volume of an instance is set to `true`. Therefore, the default is to delete the root volume of the instance when the instance terminates. The `DeleteOnTermination` attribute can be set by the creator of an AMI as well as by the person who launches an instance. When the attribute is changed by the creator of an AMI or by the person who launches an instance, the new setting overrides the original AMI default setting. We recommend that you verify the default setting for the `DeleteOnTermination` attribute after you launch an instance with an AMI.

Non-root volume

By default, when you [attach a non-root EBS volume to an instance \(p. 1042\)](#), its `DeleteOnTermination` attribute is set to `false`. Therefore, the default is to preserve these volumes. After the instance terminates, you can take a snapshot of the preserved volume or attach it to another instance. You must delete a volume to avoid incurring further charges. For more information, see [Deleting an Amazon EBS volume \(p. 1060\)](#).

To verify the value of the `DeleteOnTermination` attribute for an EBS volume that is in use, look at the instance's block device mapping. For more information, see [Viewing the EBS volumes in an instance block device mapping \(p. 1219\)](#).

You can change the value of the `DeleteOnTermination` attribute for a volume when you launch the instance or while the instance is running.

Examples

- [Changing the root volume to persist at launch using the console \(p. 611\)](#)
- [Changing the root volume to persist at launch using the command line \(p. 611\)](#)
- [Changing the root volume of a running instance to persist using the command line \(p. 612\)](#)

Changing the root volume to persist at launch using the console

Using the console, you can change the `DeleteOnTermination` attribute when you launch an instance. To change this attribute for a running instance, you must use the command line.

To change the root volume of an instance to persist at launch using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the console dashboard, select **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose an AMI and choose **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, deselect the **Delete On Termination** check box for the root volume.
6. Complete the remaining wizard pages, and then choose **Launch**.

You can verify the setting by viewing details for the root device volume on the instance's details pane. Next to **Block devices**, choose the entry for the root device volume. By default, **Delete on termination** is `True`. If you change the default behavior, **Delete on termination** is `False`.

Changing the root volume to persist at launch using the command line

When you launch an EBS-backed instance, you can use one of the following commands to change the root device volume to persist. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [run-instances \(AWS CLI\)](#)
- [New-EC2Instance \(AWS Tools for Windows PowerShell\)](#)

For example, add the following option to your `run-instances` command:

```
--block-device-mappings file://mapping.json
```

Specify the following in `mapping.json`:

```
[  
  {  
    "DeviceName": "/dev/sda1",  
    "Ebs": {  
      "DeleteOnTermination": false,  
      "SnapshotId": "snap-1234567890abcdef0",  
      "VolumeType": "gp2"  
    }  
  }  
]
```

Changing the root volume of a running instance to persist using the command line

You can use one of the following commands to change the root device volume of a running EBS-backed instance to persist. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

For example, use the following command:

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --block-device-mappings  
file://mapping.json
```

Specify the following in `mapping.json`:

```
[  
  {  
    "DeviceName": "/dev/sda1",  
    "Ebs": {  
      "DeleteOnTermination": false  
    }  
  }  
]
```

Troubleshooting

If you terminate your instance and another instance starts, most likely you have configured automatic scaling through a feature like EC2 Fleet or Amazon EC2 Auto Scaling.

If your instance is in the `shutting-down` state for longer than usual, it should be cleaned up (terminated) by automated processes within the Amazon EC2 service. For more information, see [Delayed instance termination \(p. 1256\)](#).

Recover your instance

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. Terminated instances cannot be recovered. A recovered instance is identical to the original instance, including the instance ID, private IP addresses, Elastic IP addresses, and all instance metadata. If the impaired instance is in a placement group, the recovered instance runs in the placement group. For more information about using Amazon CloudWatch alarms to recover an instance, see [Adding recover actions to Amazon CloudWatch alarms \(p. 742\)](#). To troubleshoot issues with instance recovery failures, see [Troubleshooting instance recovery failures \(p. 613\)](#).

When the `StatusCheckFailed_System` alarm is triggered, and the recover action is initiated, you will be notified by the Amazon SNS topic that you selected when you created the alarm and associated the recover action. During instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is lost. When the process is complete, information is published to the SNS topic you've configured for the alarm. Anyone who is subscribed to this SNS topic will receive an email notification that includes the status of the recovery attempt and any further instructions. You will notice an instance reboot on the recovered instance.

Examples of problems that cause system status checks to fail include:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host that impact network reachability

If your instance has a public IPv4 address, it retains the public IPv4 address after recovery.

Requirements

The recover action is supported only on instances with the following characteristics:

- Uses one of the following instance types: A1, C3, C4, C5, C5a, C5n, C6g, Inf1, M3, M4, M5, M5a, M5n, M6g, P3, R3, R4, R5, R5a, R5n, R6g, T2, T3, T3a, T4g, X1, or X1e
- Runs in a virtual private cloud (VPC)
- Uses default or dedicated instance tenancy
- Has only EBS volumes (do not configure instance store volumes)

Troubleshooting instance recovery failures

The following issues can cause automatic recovery of your instance to fail:

- Temporary, insufficient capacity of replacement hardware.
- The instance has an attached instance store storage, which is an unsupported configuration for automatic instance recovery.
- There is an ongoing Service Health Dashboard event that prevented the recovery process from successfully executing. Refer to <http://status.aws.amazon.com/> for the latest service availability information.
- The instance has reached the maximum daily allowance of three recovery attempts.

The automatic recovery process attempts to recover your instance for up to three separate failures per day. If the instance system status check failure persists, we recommend that you manually stop and start the instance. For more information, see [Stop and start your instance \(p. 587\)](#).

Your instance may subsequently be retired if automatic recovery fails and a hardware degradation is determined to be the root cause for the original system status check failure.

Configuring your Amazon Linux instance

After you have successfully launched and logged into your Amazon Linux instance, you can make changes to it. There are many different ways you can configure an instance to meet the needs of a specific application. The following are some common tasks to help get you started.

Contents

- [Common configuration scenarios \(p. 614\)](#)
- [Managing software on your Amazon Linux instance \(p. 614\)](#)
- [Managing user accounts on your Amazon Linux instance \(p. 620\)](#)
- [Processor state control for your EC2 instance \(p. 622\)](#)
- [Setting the time for your Linux instance \(p. 628\)](#)
- [Optimizing CPU options \(p. 632\)](#)
- [Changing the hostname of your Amazon Linux instance \(p. 648\)](#)
- [Setting up dynamic DNS on Your Amazon Linux instance \(p. 651\)](#)
- [Running commands on your Linux instance at launch \(p. 653\)](#)
- [Instance metadata and user data \(p. 658\)](#)

Common configuration scenarios

The base distribution of Amazon Linux contains many software packages and utilities that are required for basic server operations. However, many more software packages are available in various software repositories, and even more packages are available for you to build from source code. For more information on installing and building software from these locations, see [Managing software on your Amazon Linux instance \(p. 614\)](#).

Amazon Linux instances come pre-configured with an `ec2-user` account, but you may want to add other user accounts that do not have super-user privileges. For more information on adding and removing user accounts, see [Managing user accounts on your Amazon Linux instance \(p. 620\)](#).

The default time configuration for Amazon Linux instances uses Amazon Time Sync Service to set the system time on an instance. The default time zone is UTC. For more information on setting the time zone for an instance or using your own time server, see [Setting the time for your Linux instance \(p. 628\)](#).

If you have your own network with a domain name registered to it, you can change the hostname of an instance to identify itself as part of that domain. You can also change the system prompt to show a more meaningful name without changing the hostname settings. For more information, see [Changing the hostname of your Amazon Linux instance \(p. 648\)](#). You can configure an instance to use a dynamic DNS service provider. For more information, see [Setting up dynamic DNS on Your Amazon Linux instance \(p. 651\)](#).

When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common configuration tasks and even run scripts after the instance starts. You can pass two types of user data to Amazon EC2: cloud-init directives and shell scripts. For more information, see [Running commands on your Linux instance at launch \(p. 653\)](#).

Managing software on your Amazon Linux instance

The base distribution of Amazon Linux contains many software packages and utilities that are required for basic server operations. However, many more software packages are available in various software repositories, and even more packages are available for you to build from source code.

Contents

- [Updating instance software on your Amazon Linux instance \(p. 615\)](#)
- [Adding repositories on an Amazon Linux instance \(p. 616\)](#)
- [Finding software packages on an Amazon Linux instance \(p. 617\)](#)
- [Installing software packages on an Amazon Linux instance \(p. 618\)](#)

- [Preparing to compile software on an Amazon Linux instance \(p. 619\)](#)

It is important to keep software up-to-date. Many packages in a Linux distribution are updated frequently to fix bugs, add features, and protect against security exploits. For more information, see [Updating instance software on your Amazon Linux instance \(p. 615\)](#).

By default, Amazon Linux instances launch with the following repositories enabled:

- Amazon Linux 2: `amzn2-core` and `amzn2extra-docker`
- Amazon Linux AMI: `amzn-main` and `amzn-updates`

While there are many packages available in these repositories that are updated by Amazon Web Services, there may be a package that you wish to install that is contained in another repository. For more information, see [Adding repositories on an Amazon Linux instance \(p. 616\)](#). For help finding packages in enabled repositories, see [Finding software packages on an Amazon Linux instance \(p. 617\)](#). For information about installing software on an Amazon Linux instance, see [Installing software packages on an Amazon Linux instance \(p. 618\)](#).

Not all software is available in software packages stored in repositories; some software must be compiled on an instance from its source code. For more information, see [Preparing to compile software on an Amazon Linux instance \(p. 619\)](#).

Amazon Linux instances manage their software using the yum package manager. The yum package manager can install, remove, and update software, as well as manage all of the dependencies for each package. Debian-based Linux distributions, like Ubuntu, use the `apt-get` command and `dpkg` package manager, so the `yum` examples in the following sections do not work for those distributions.

Updating instance software on your Amazon Linux instance

It is important to keep software up-to-date. Many packages in a Linux distribution are updated frequently to fix bugs, add features, and protect against security exploits. When you first launch and connect to an Amazon Linux instance, you may see a message asking you to update software packages for security purposes. This section shows how to update an entire system, or just a single package.

Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

To update all packages on an Amazon Linux instance

1. (Optional) Start a `screen` session in your shell window. Sometimes you may experience a network interruption that can disconnect the SSH connection to your instance. If this happens during a long software update, it can leave the instance in a recoverable, although confused state. A `screen` session allows you to continue running the update even if your connection is interrupted, and you can reconnect to the session later without problems.
 - a. Execute the `screen` command to begin the session.

```
[ec2-user ~]$ screen
```

- b. If your session is disconnected, log back into your instance and list the available screens.

```
[ec2-user ~]$ screen -ls
There is a screen on:
  17793.pts-0.ip-12-34-56-78 (Detached)
  1 Socket in /var/run/screen/S-ec2-user.
```

- c. Reconnect to the screen using the **screen -r** command and the process ID from the previous command.

```
[ec2-user ~]$ screen -r 17793
```

- d. When you are finished using **screen**, use the **exit** command to close the session.

```
[ec2-user ~]$ exit  
[screen is terminating]
```

2. Run the **yum update** command. Optionally, you can add the **--security** flag to apply only security updates.

```
[ec2-user ~]$ sudo yum update
```

3. Review the packages listed, type **y**, and press Enter to accept the updates. Updating all of the packages on a system can take several minutes. The **yum** output shows the status of the update while it is running.
4. (Optional) Reboot your instance to ensure that you are using the latest packages and libraries from your update; kernel updates are not loaded until a reboot occurs. Updates to any glibc libraries should also be followed by a reboot. For updates to packages that control services, it may be sufficient to restart the services to pick up the updates, but a system reboot ensures that all previous package and library updates are complete.

To update a single package on an Amazon Linux instance

Use this procedure to update a single package (and its dependencies) and not the entire system.

1. Run the **yum update** command with the name of the package you would like to update.

```
[ec2-user ~]$ sudo yum update openssl
```

2. Review the package information listed, type **y**, and press Enter to accept the update or updates. Sometimes there will be more than one package listed if there are package dependencies that must be resolved. The **yum** output shows the status of the update while it is running.
3. (Optional) Reboot your instance to ensure that you are using the latest packages and libraries from your update; kernel updates are not loaded until a reboot occurs. Updates to any glibc libraries should also be followed by a reboot. For updates to packages that control services, it may be sufficient to restart the services to pick up the updates, but a system reboot ensures that all previous package and library updates are complete.

Adding repositories on an Amazon Linux instance

By default, Amazon Linux instances launch with two repositories enabled: `amzn-main` and `amzn-updates`. While there are many packages available in these repositories that are updated by Amazon Web Services, there may be a package that you wish to install that is contained in another repository.

Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

To install a package from a different repository with **yum**, you need to add the repository information to the `/etc/yum.conf` file or to its own `repository.repo` file in the `/etc/yum.repos.d` directory. You can do this manually, but most yum repositories provide their own `repository.repo` file at their repository URL.

To determine what yum repositories are already installed

- List the installed yum repositories with the following command:

```
[ec2-user ~]$ yum repolist all
```

The resulting output lists the installed repositories and reports the status of each. Enabled repositories display the number of packages they contain.

To add a yum repository to /etc/yum.repos.d

1. Find the location of the .repo file. This will vary depending on the repository you are adding. In this example, the .repo file is at <https://www.example.com/repository.repo>.
2. Add the repository with the **yum-config-manager** command.

```
[ec2-user ~]$ sudo yum-config-manager --add-repo https://  
www.example.com/repository.repo  
Loaded plugins: priorities, update-motd, upgrade-helper  
adding repo from: https://www.example.com/repository.repo  
grabbing file https://www.example.com/repository.repo to /etc/  
yum.repos.d/repository.repo  
repository.repo | 4.0 kB     00:00  
repo saved to /etc/yum.repos.d/repository.repo
```

After you install a repository, you must enable it as described in the next procedure.

To enable a yum repository in /etc/yum.repos.d

- Use the **yum-config-manager** command with the --enable *repository* flag. The following command enables the Extra Packages for Enterprise Linux (EPEL) repository from the Fedora project. By default, this repository is present in /etc/yum.repos.d on Amazon Linux AMI instances, but it is not enabled.

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

Note

To enable the EPEL repository on Amazon Linux 2, use the following command:

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-  
release-latest-7.noarch.rpm
```

For information on enabling the EPEL repository on other distributions, such as Red Hat and CentOS, see the EPEL documentation at <https://fedoraproject.org/wiki/EPEL>.

Finding software packages on an Amazon Linux instance

You can use the **yum search** command to search the descriptions of packages that are available in your configured repositories. This is especially helpful if you don't know the exact name of the package you want to install. Simply append the keyword search to the command; for multiple word searches, wrap the search query with quotation marks.

Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

```
[ec2-user ~]$ sudo yum search "find"
```

The following is example output for Amazon Linux 2.

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
=====
N/S matched: find =====
findutils.x86_64 : The GNU versions of find utilities (find and xargs)
gedit-plugin-findinfiles.x86_64 : gedit findinfiles plugin
ocaml-findlib-devel.x86_64 : Development files for ocaml-findlib
perl-File-Find-Rule.noarch : Perl module implementing an alternative interface to
    File::Find
robotfindskitten.x86_64 : A game/zen simulation. You are robot. Your job is to find kitten.
mlocate.x86_64 : An utility for finding files by name
ocaml-findlib.x86_64 : Objective CAML package manager and build helper
perl-Devel-Cycle.noarch : Find memory cycles in objects
perl-Devel-EnforceEncapsulation.noarch : Find access violations to blessed objects
perl-File-Find-Rule-Perl.noarch : Common rules for searching for Perl things
perl-File-HomeDir.noarch : Find your home and other directories on any platform
perl-IPC-Cmd.noarch : Finding and running system commands made easy
perl-Perl-MinimumVersion.noarch : Find a minimum required version of perl for Perl code
texlive-xesearch.noarch : A string finder for XeTeX
valgrind.x86_64 : Tool for finding memory management bugs in programs
valgrind.i686 : Tool for finding memory management bugs in programs
```

The following is example output for Amazon Linux.

```
Loaded plugins: priorities, security, update-motd, upgrade-helper
=====
N/S Matched: find =====
findutils.x86_64 : The GNU versions of find utilities (find and xargs)
perl-File-Find-Rule.noarch : Perl module implementing an alternative interface to
    File::Find
perl-Module-Find.noarch : Find and use installed modules in a (sub)category
libpuzzle.i686 : Library to quickly find visually similar images (gif, png, jpg)
libpuzzle.x86_64 : Library to quickly find visually similar images (gif, png, jpg)
mlocate.x86_64 : An utility for finding files by name
```

Multiple word search queries in quotation marks only return results that match the exact query. If you don't see the expected package, simplify your search to one keyword and then scan the results. You can also try keyword synonyms to broaden your search.

For more information about packages for Amazon Linux 2 and Amazon Linux, see the following:

- [Package repository \(p. 178\)](#)
- [Extras library \(Amazon Linux 2\) \(p. 180\)](#)

Installing software packages on an Amazon Linux instance

The yum package manager is a great tool for installing software, because it can search all of your enabled repositories for different software packages and also handle any dependencies in the software installation process.

Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

To install a package from a repository, use the **yum install *package*** command, replacing *package* with the name of the software to install. For example, to install the **links** text-based web browser, enter the following command.

```
[ec2-user ~]$ sudo yum install links
```

You can also use **yum install** to install RPM package files that you have downloaded from the Internet. To do this, simply append the path name of an RPM file to the installation command instead of a repository package name.

```
[ec2-user ~]$ sudo yum install my-package.rpm
```

Preparing to compile software on an Amazon Linux instance

There is a wealth of open-source software available on the Internet that has not been pre-compiled and made available for download from a package repository. You may eventually discover a software package that you need to compile yourself, from its source code. For your system to be able to compile software, you need to install several development tools, such as **make**, **gcc**, and **autoconf**.

Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

Because software compilation is not a task that every Amazon EC2 instance requires, these tools are not installed by default, but they are available in a package group called "Development Tools" that is easily added to an instance with the **yum groupinstall** command.

```
[ec2-user ~]$ sudo yum groupinstall "Development Tools"
```

Software source code packages are often available for download (from web sites such as <https://github.com/> and <http://sourceforge.net/>) as a compressed archive file, called a tarball. These tarballs will usually have the `.tar.gz` file extension. You can decompress these archives with the **tar** command.

```
[ec2-user ~]$ tar -xzf software.tar.gz
```

After you have decompressed and unarchived the source code package, you should look for a `README` or `INSTALL` file in the source code directory that can provide you with further instructions for compiling and installing the source code.

To retrieve source code for Amazon Linux packages

Amazon Web Services provides the source code for maintained packages. You can download the source code for any installed packages with the **yumdownloader --source** command.

- Run the **yumdownloader --source package** command to download the source code for `package`. For example, to download the source code for the `htop` package, enter the following command.

```
[ec2-user ~]$ yumdownloader --source htop
Loaded plugins: priorities, update-motd, upgrade-helper
Enabling amzn-updates-source repository
Enabling amzn-main-source repository
amzn-main-source
| 1.9 kB  00:00:00
amzn-updates-source
| 1.9 kB  00:00:00
(1/2): amzn-updates-source/latest/primary_db
| 52 kB  00:00:00
(2/2): amzn-main-source/latest/primary_db
| 734 kB  00:00:00
htop-1.0.1-2.3.amzn1.src.rpm
```

The location of the source RPM is in the directory from which you ran the command.

Managing user accounts on your Amazon Linux instance

Each Linux instance launches with a default Linux system user account. The default user name is determined by the AMI that was specified when you launched the instance.

- For Amazon Linux 2 or the Amazon Linux AMI, the user name is `ec2-user`.
- For a CentOS AMI, the user name is `centos`.
- For a Debian AMI, the user name is `admin`.
- For a Fedora AMI, the user name is `ec2-user` or `fedora`.
- For a RHEL AMI, the user name is `ec2-user` or `root`.
- For a SUSE AMI, the user name is `ec2-user` or `root`.
- For an Ubuntu AMI, the user name is `ubuntu`.
- Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.

Note

Linux system users should not be confused with AWS Identity and Access Management (IAM) users. For more information, see [IAM Users and Groups](#) in the *IAM User Guide*.

Contents

- [Considerations \(p. 620\)](#)
- [Creating a user account \(p. 620\)](#)
- [Removing a user account \(p. 622\)](#)

Considerations

Using the default user account is adequate for many applications. However, you may choose to add user accounts so that individuals can have their own files and workspaces. Furthermore, creating user accounts for new users is much more secure than granting multiple (possibly inexperienced) users access to the default user account, because the default user account can cause a lot of damage to a system when used improperly. For more information, see [Tips for Securing Your EC2 Instance](#).

To enable users SSH access to your EC2 instance using a Linux system user account, you must share the SSH key with the user. Alternatively, you can use EC2 Instance Connect to provide access to users without the need to share and manage SSH keys. For more information, see [Connecting to your Linux instance using EC2 Instance Connect \(p. 569\)](#).

Creating a user account

First create the user account, and then add the SSH public key that allows the user to connect to and log into the instance.

To create a user account

1. [Create a new key pair \(p. 986\)](#). You must provide the `.pem` file to the user for whom you are creating the user account. They must use this file to connect to the instance.
2. Retrieve the public key from the key pair that you created in the previous step.

```
$ ssh-keygen -y -f /path_to_key_pair/key-pair-name.pem
```

The command returns the public key, as shown in the following example.

```
ssh-rsa  
AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706Vhz2ITxCi  
+PnDSUaw+WNQn/mZphTk/a/gu8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/  
d6RJhJOI0iBXrlsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/  
cQk+0FzzqaeJAAHco+CY/5WrUBkrHmJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi  
+z7wB3RbBQoQzd8v7yeb70z1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

3. Connect to the instance.
4. Use the **adduser** command to create the user account and add it to the system (with an entry in the `/etc/passwd` file). The command also creates a group and a home directory for the account. In this example, the user account is named `newuser`.
 - Amazon Linux and Amazon Linux 2

```
[ec2-user ~]$ sudo adduser newuser
```

- Ubuntu

Include the `--disabled-password` parameter to create the user account without a password.

```
[ubuntu ~]$ sudo adduser newuser --disabled-password
```

5. Switch to the new account so that the directory and file that you create will have the proper ownership.

```
[ec2-user ~]$ sudo su - newuser
```

The prompt changes from `ec2-user` to `newuser` to indicate that you have switched the shell session to the new account.

6. Add the SSH public key to the user account. First create a directory in the user's home directory for the SSH key file, then create the key file, and finally paste the public key into the key file, as described in the following sub-steps.
 - a. Create a `.ssh` directory in the `newuser` home directory and change its file permissions to 700 (only the owner can read, write, or open the directory).

```
[newuser ~]$ mkdir .ssh
```

```
[newuser ~]$ chmod 700 .ssh
```

Important

Without these exact file permissions, the user will not be able to log in.

- b. Create a file named `authorized_keys` in the `.ssh` directory and change its file permissions to 600 (only the owner can read or write to the file).

```
[newuser ~]$ touch .ssh/authorized_keys
```

```
[newuser ~]$ chmod 600 .ssh/authorized_keys
```

Important

Without these exact file permissions, the user will not be able to log in.

- c. Open the `authorized_keys` file using your favorite text editor (such as `vim` or `nano`).

```
[newuser ~]$ nano .ssh/authorized_keys
```

Paste the public key that you retrieved in **Step 2** into the file and save the changes.

Important

Ensure that you paste the public key in one continuous line. The public key must not be split over multiple lines.

The user should now be able to log into the `newuser` account on your instance, using the private key that corresponds to the public key that you added to the `authorized_keys` file. For more information about the different methods of connecting to a Linux instance, see [Connect to your Linux instance \(p. 563\)](#).

Removing a user account

If a user account is no longer needed, you can remove that account so that it can no longer be used.

Use the `userdel` command to remove the user account from the system. When you specify the `-r` parameter, the user's home directory and mail spool are deleted. To keep the user's home directory and mail spool, omit the `-r` parameter.

```
[ec2-user ~]$ sudo userdel -r olduser
```

Processor state control for your EC2 instance

C-states control the sleep levels that a core can enter when it is idle. C-states are numbered starting with C0 (the shallowest state where the core is totally awake and executing instructions) and go to C6 (the deepest idle state where a core is powered off). P-states control the desired performance (in CPU frequency) from a core. P-states are numbered starting from P0 (the highest performance setting where the core is allowed to use Intel Turbo Boost Technology to increase frequency if possible), and they go from P1 (the P-state that requests the maximum baseline frequency) to P15 (the lowest possible frequency).

The following instance types provide the ability for an operating system to control processor C-states and P-states:

- General purpose: m4.10xlarge | m4.16xlarge | m5.metal | m5d.metal
- Compute optimized: c4.8xlarge | c5.metal | c5n.metal
- Memory optimized: r4.8xlarge | r4.16xlarge | r5.metal | r5d.metal | u-6tb1.metal | u-9tb1.metal | u-12tb1.metal | u-18tb1.metal | u-24tb1.metal | x1.16xlarge | x1.32xlarge | x1e.8xlarge | x1e.16xlarge | x1e.32xlarge | z1d.metal
- Storage optimized: d2.8xlarge | i3.8xlarge | i3.16xlarge | i3.metal | i3en.metal | h1.8xlarge | h1.16xlarge
- Accelerated computing: f1.16xlarge | g3.16xlarge | g4dn.metal | p2.16xlarge | p3.16xlarge

The following instance types provide the ability for an operating system to control processor C-states:

- General purpose: m5.12xlarge | m5.24xlarge | m5d.12xlarge | m5d.24xlarge | m5n.12xlarge | m5n.24xlarge | m5dn.12xlarge | m5dn.24xlarge

- Compute optimized: c5.9xlarge | c5.12xlarge | c5.18xlarge | c5.24xlarge | c5a.24xlarge | c5ad.24xlarge | c5d.9xlarge | c5d.12xlarge | c5d.18xlarge | c5d.24xlarge | c5n.9xlarge | c5n.18xlarge
- Memory optimized: r5.12xlarge | r5.24xlarge | r5d.12xlarge | r5d.24xlarge | r5n.12xlarge | r5n.24xlarge | r5dn.12xlarge | r5dn.24xlarge | z1d.6xlarge | z1d.12xlarge
- Storage optimized: i3en.12xlarge | i3en.24xlarge
- Accelerated computing: inf1.24xlarge | p3dn.24xlarge

AWS Graviton processors have built-in power saving modes and operate at a fixed frequency. Therefore, they do not provide the ability for the operating system to control C-states and P-states.

You might want to change the C-state or P-state settings to increase processor performance consistency, reduce latency, or tune your instance for a specific workload. The default C-state and P-state settings provide maximum performance, which is optimal for most workloads. However, if your application would benefit from reduced latency at the cost of higher single- or dual-core frequencies, or from consistent performance at lower frequencies as opposed to bursty Turbo Boost frequencies, consider experimenting with the C-state or P-state settings that are available to these instances.

The following sections describe the different processor state configurations and how to monitor the effects of your configuration. These procedures were written for, and apply to Amazon Linux; however, they may also work for other Linux distributions with a Linux kernel version of 3.9 or newer. For more information about other Linux distributions and processor state control, see your system-specific documentation.

Note

The examples on this page use the **turbostat** utility (which is available on Amazon Linux by default) to display processor frequency and C-state information, and the **stress** command (which can be installed by running **sudo yum install -y stress**) to simulate a workload.

If the output does not display the C-state information, include the **--debug** option in the command (**sudo turbostat --debug stress <options>**).

Contents

- [Highest performance with maximum Turbo Boost frequency \(p. 623\)](#)
- [High performance and low latency by limiting deeper C-states \(p. 624\)](#)
- [Baseline performance with the lowest variability \(p. 626\)](#)

Highest performance with maximum Turbo Boost frequency

This is the default processor state control configuration for the Amazon Linux AMI, and it is recommended for most workloads. This configuration provides the highest performance with lower variability. Allowing inactive cores to enter deeper sleep states provides the thermal headroom required for single or dual core processes to reach their maximum Turbo Boost potential.

The following example shows a c4.8xlarge instance with two cores actively performing work reaching their maximum processor Turbo Boost frequency.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [30680] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [30680] successful run completed in 10s
pk cor CPU %c0 GHz TSC SMI %c1 %c3 %c6 %c7 %pc2 %pc3 %pc6 %pc7
Pkg_W RAM_W PKG_% RAM_%
      5.54 3.44 2.90   0   9.18   0.00  85.28   0.00   0.00   0.00   0.00   0.00
 94.04 32.70 54.18  0.00
  0    0    0.12 3.26 2.90   0   3.61   0.00  96.27   0.00   0.00   0.00   0.00
 48.12 18.88 26.02  0.00
```

```

0 0 18 0.12 3.26 2.90 0 3.61
0 1 1 0.12 3.26 2.90 0 4.11 0.00 95.77 0.00
0 1 19 0.13 3.27 2.90 0 4.11
0 2 2 0.13 3.28 2.90 0 4.45 0.00 95.42 0.00
0 2 20 0.11 3.27 2.90 0 4.47
0 3 3 0.05 3.42 2.90 0 99.91 0.00 0.05 0.00
0 3 21 97.84 3.45 2.90 0 2.11
...
1 1 10 0.06 3.33 2.90 0 99.88 0.01 0.06 0.00
1 1 28 97.61 3.44 2.90 0 2.32
...
10.002556 sec

```

In this example, vCPUs 21 and 28 are running at their maximum Turbo Boost frequency because the other cores have entered the C6 sleep state to save power and provide both power and thermal headroom for the working cores. vCPUs 3 and 10 (each sharing a processor core with vCPUs 21 and 28) are in the C1 state, waiting for instruction.

In the following example, all 18 cores are actively performing work, so there is no headroom for maximum Turbo Boost, but they are all running at the "all core Turbo Boost" speed of 3.2 GHz.

```

[ec2-user ~]$ sudo turbostat stress -c 36 -t 10
stress: info: [30685] dispatching hogs: 36 cpu, 0 io, 0 vm, 0 hdd
stress: info: [30685] successful run completed in 10s
pk cor CPU %c0 GHz TSC SMI %c1 %c3 %c6 %c7 %pc2 %pc3 %pc6 %pc7
Pkg_W RAM_W PKG_% RAM_%
         99.27 3.20 2.90 0 0.26 0.00 0.47 0.00 0.00 0.00 0.00 0.00
228.59 31.33 199.26 0.00
0 0 0 99.08 3.20 2.90 0 0.27 0.01 0.64 0.00 0.00 0.00 0.00
114.69 18.55 99.32 0.00
0 0 18 98.74 3.20 2.90 0 0.62
0 1 1 99.14 3.20 2.90 0 0.09 0.00 0.76 0.00
0 1 19 98.75 3.20 2.90 0 0.49
0 2 2 99.07 3.20 2.90 0 0.10 0.02 0.81 0.00
0 2 20 98.73 3.20 2.90 0 0.44
0 3 3 99.02 3.20 2.90 0 0.24 0.00 0.74 0.00
0 3 21 99.13 3.20 2.90 0 0.13
0 4 4 99.26 3.20 2.90 0 0.09 0.00 0.65 0.00
0 4 22 98.68 3.20 2.90 0 0.67
0 5 5 99.19 3.20 2.90 0 0.08 0.00 0.73 0.00
0 5 23 98.58 3.20 2.90 0 0.69
0 6 6 99.01 3.20 2.90 0 0.11 0.00 0.89 0.00
0 6 24 98.72 3.20 2.90 0 0.39
...

```

High performance and low latency by limiting deeper C-states

C-states control the sleep levels that a core may enter when it is inactive. You may want to control C-states to tune your system for latency versus performance. Putting cores to sleep takes time, and although a sleeping core allows more headroom for another core to boost to a higher frequency, it takes time for that sleeping core to wake back up and perform work. For example, if a core that is assigned to handle network packet interrupts is asleep, there may be a delay in servicing that interrupt. You can configure the system to not use deeper C-states, which reduces the processor reaction latency, but that in turn also reduces the headroom available to other cores for Turbo Boost.

A common scenario for disabling deeper sleep states is a Redis database application, which stores the database in system memory for the fastest possible query response time.

To limit deeper sleep states on Amazon Linux 2

1. Open the /etc/default/grub file with your editor of choice.

```
[ec2-user ~]$ sudo vim /etc/default/grub
```

2. Edit the GRUB_CMDLINE_LINUX_DEFAULT line and add the intel_idle.max_cstate=1 option to set C1 as the deepest C-state for idle cores.

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0  
biosdevname=0 nvme_core.io_timeout=4294967295 intel_idle.max_cstate=1"  
GRUB_TIMEOUT=0
```

3. Save the file and exit your editor.
4. Run the following command to rebuild the boot configuration.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. Reboot your instance to enable the new kernel option.

```
[ec2-user ~]$ sudo reboot
```

To limit deeper sleep states on Amazon Linux AMI

1. Open the /boot/grub/grub.conf file with your editor of choice.

```
[ec2-user ~]$ sudo vim /boot/grub/grub.conf
```

2. Edit the kernel line of the first entry and add the intel_idle.max_cstate=1 option to set C1 as the deepest C-state for idle cores.

```
# created by imagebuilder  
default=0  
timeout=1  
hiddenmenu  
  
title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)  
root (hd0,0)  
kernel /boot/vmlinuz-3.14.26-24.46.amzn1.x86_64 root=LABEL=/ console=ttyS0  
intel_idle.max_cstate=1  
initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
```

3. Save the file and exit your editor.
4. Reboot your instance to enable the new kernel option.

```
[ec2-user ~]$ sudo reboot
```

The following example shows a c4.8xlarge instance with two cores actively performing work at the "all core Turbo Boost" core frequency.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10  
stress: info: [5322] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd  
stress: info: [5322] successful run completed in 10s  
pk cor CPU %c0 GHz TSC SMI %c1 %c3 %c6 %c7 %pc2 %pc3 %pc6 %pc7  
Pkg_W RAM_W PKG_% RAM_%  
      5.56 3.20 2.90  0 94.44  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  
131.90 31.11 199.47  0.00  
 0   0   0.03 2.08 2.90  0 99.97  0.00  0.00  0.00  0.00  0.00  0.00  
67.23 17.11 99.76  0.00
```

0	0	18	0.01	1.93	2.90	0	99.99			
0	1	1	0.02	1.96	2.90	0	99.98	0.00	0.00	0.00
0	1	19	99.70	3.20	2.90	0	0.30			
...										
1	1	10	0.02	1.97	2.90	0	99.98	0.00	0.00	0.00
1	1	28	99.67	3.20	2.90	0	0.33			
1	2	11	0.04	2.63	2.90	0	99.96	0.00	0.00	0.00
1	2	29	0.02	2.11	2.90	0	99.98			
...										

In this example, the cores for vCPUs 19 and 28 are running at 3.2 GHz, and the other cores are in the C1 C-state, awaiting instruction. Although the working cores are not reaching their maximum Turbo Boost frequency, the inactive cores will be much faster to respond to new requests than they would be in the deeper C6 C-state.

Baseline performance with the lowest variability

You can reduce the variability of processor frequency with P-states. P-states control the desired performance (in CPU frequency) from a core. Most workloads perform better in P0, which requests Turbo Boost. But you may want to tune your system for consistent performance rather than bursty performance that can happen when Turbo Boost frequencies are enabled.

Intel Advanced Vector Extensions (AVX or AVX2) workloads can perform well at lower frequencies, and AVX instructions can use more power. Running the processor at a lower frequency, by disabling Turbo Boost, can reduce the amount of power used and keep the speed more consistent. For more information about optimizing your instance configuration and workload for AVX, see <http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/performance-xeon-e5-v3-advanced-vector-extensions-paper.pdf>.

This section describes how to limit deeper sleep states and disable Turbo Boost (by requesting the P1 P-state) to provide low-latency and the lowest processor speed variability for these types of workloads.

To limit deeper sleep states and disable Turbo Boost on Amazon Linux 2

1. Open the /etc/default/grub file with your editor of choice.

```
[ec2-user ~]$ sudo vim /etc/default/grub
```

2. Edit the GRUB_CMDLINE_LINUX_DEFAULT line and add the intel_idle.max_cstate=1 option to set C1 as the deepest C-state for idle cores.

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
biosdevname=0 nvme_core.io_timeout=4294967295 intel_idle.max_cstate=1
GRUB_TIMEOUT=0
```

3. Save the file and exit your editor.
4. Run the following command to rebuild the boot configuration.

```
[ec2-user ~]$ grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. Reboot your instance to enable the new kernel option.

```
[ec2-user ~]$ sudo reboot
```

6. When you need the low processor speed variability that the P1 P-state provides, execute the following command to disable Turbo Boost.

```
[ec2-user ~]$ sudo sh -c "echo 1 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

- When your workload is finished, you can re-enable Turbo Boost with the following command.

```
[ec2-user ~]$ sudo sh -c "echo 0 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

To limit deeper sleep states and disable Turbo Boost on Amazon Linux AMI

- Open the /boot/grub/grub.conf file with your editor of choice.

```
[ec2-user ~]$ sudo vim /boot/grub/grub.conf
```

- Edit the kernel line of the first entry and add the intel_idle.max_cstate=1 option to set C1 as the deepest C-state for idle cores.

```
# created by imagebuilder
default=0
timeout=1
hiddenmenu

title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-3.14.26-24.46.amzn1.x86_64 root=LABEL=/ console=ttyS0
    intel_idle.max_cstate=1
initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
```

- Save the file and exit your editor.
- Reboot your instance to enable the new kernel option.

```
[ec2-user ~]$ sudo reboot
```

- When you need the low processor speed variability that the P1 P-state provides, execute the following command to disable Turbo Boost.

```
[ec2-user ~]$ sudo sh -c "echo 1 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

- When your workload is finished, you can re-enable Turbo Boost with the following command.

```
[ec2-user ~]$ sudo sh -c "echo 0 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

The following example shows a c4.8xlarge instance with two vCPUs actively performing work at the baseline core frequency, with no Turbo Boost.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [5389] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [5389] successful run completed in 10s
pk cor CPU %c0 GHz TSC SMI %c1 %c3 %c6 %c7 %pc2 %pc3 %pc6 %pc7
Pkg_W RAM_W PKG_% RAM%
      5.59 2.90 2.90   0  94.41  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
128.48 33.54 200.00  0.00
0   0   0   0.04 2.90 2.90   0  99.96  0.00  0.00  0.00  0.00  0.00  0.00
65.33 19.02 100.00  0.00
0   0   18  0.04 2.90 2.90   0  99.96
0   1   1   0.05 2.90 2.90   0  99.95  0.00  0.00  0.00
0   1   19  0.04 2.90 2.90   0  99.96
0   2   2   0.04 2.90 2.90   0  99.96  0.00  0.00  0.00
0   2   20  0.04 2.90 2.90   0  99.96
0   3   3   0.05 2.90 2.90   0  99.95  0.00  0.00  0.00
0   3   21  99.95 2.90 2.90   0  0.05
```

```
...
1   1   28  99.92 2.90    0   0.08
1   2   11  0.06 2.90 2.90    0  99.94    0.00   0.00   0.00
1   2   29  0.05 2.90 2.90    0  99.95
```

The cores for vCPUs 21 and 28 are actively performing work at the baseline processor speed of 2.9 GHz, and all inactive cores are also running at the baseline speed in the C1 C-state, ready to accept instructions.

Setting the time for your Linux instance

A consistent and accurate time reference is crucial for many server tasks and processes. Most system logs include a time stamp that you can use to determine when problems occur and in what order the events take place. If you use the AWS CLI or an AWS SDK to make requests from your instance, these tools sign requests on your behalf. If your instance's date and time are not set correctly, the date in the signature may not match the date of the request, and AWS rejects the request.

Amazon provides the Amazon Time Sync Service, which is accessible from all EC2 instances, and is also used by other AWS services. This service uses a fleet of satellite-connected and atomic reference clocks in each Region to deliver accurate current time readings of the Coordinated Universal Time (UTC) global standard through Network Time Protocol (NTP). The Amazon Time Sync Service automatically smooths any leap seconds that are added to UTC.

The Amazon Time Sync Service is available through NTP at the 169.254.169.123 IP address for any instance running in a VPC. Your instance does not require access to the internet, and you do not have to configure your security group rules or your network ACL rules to allow access. The latest versions of Amazon Linux 2 and Amazon Linux AMIs synchronize with the Amazon Time Sync Service by default.

Use the following procedures to configure the Amazon Time Sync Service on your instance using the `chrony` client. Alternatively, you can use external NTP sources. For more information about NTP and public time sources, see <http://www.ntp.org/>. An instance needs access to the internet for the external NTP time sources to work.

Topics

- [Configuring the Amazon Time Sync Service on Amazon Linux AMI \(p. 628\)](#)
- [Configuring the Amazon Time Sync Service on Ubuntu \(p. 630\)](#)
- [Configuring the Amazon Time Sync Service on SUSE Linux \(p. 631\)](#)
- [Changing the time zone on Amazon Linux \(p. 631\)](#)

Configuring the Amazon Time Sync Service on Amazon Linux AMI

Note

On Amazon Linux 2, the default `chrony` configuration is already set up to use the Amazon Time Sync Service IP address.

With the Amazon Linux AMI, you must edit the `chrony` configuration file to add a server entry for the Amazon Time Sync Service.

To configure your instance to use the Amazon Time Sync Service

1. Connect to your instance and uninstall the NTP service.

```
[ec2-user ~]$ sudo yum erase 'ntp*'
```

2. Install the chrony package.

```
[ec2-user ~]$ sudo yum install chrony
```

3. Open the /etc/chrony.conf file using a text editor (such as **vim** or **nano**). Verify that the file includes the following line:

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

If the line is present, then the Amazon Time Sync Service is already configured and you can go to the next step. If not, add the line after any other `server` or `pool` statements that are already present in the file, and save your changes.

4. Restart the chrony daemon (`chronyd`).

```
[ec2-user ~]$ sudo service chronyd restart
```

```
Starting chronyd: [ OK ]
```

Note

On RHEL and CentOS (up to version 6), the service name is `chrony` instead of `chronyd`.

5. Use the `chkconfig` command to configure `chronyd` to start at each system boot.

```
[ec2-user ~]$ sudo chkconfig chronyd on
```

6. Verify that `chrony` is using the 169.254.169.123 IP address to synchronize the time.

```
[ec2-user ~]$ chronyc sources -v
```

```
210 Number of sources = 7

.-- Source mode '^' = server, '=' = peer, '#' = local clock.
/- Source state '*' = current synced, '+' = combined , '-' = not combined,
| / '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||                                     .- xxxx [ yyyy ] +/- zzzz
||          Reachability register (octal) -. | xxxx = adjusted offset,
||          Log2(Polling interval) --. | | yyyy = measured offset,
||                                     \ | zzzz = estimated error.
||                                     | |
MS Name/IP address      Stratum Poll Reach LastRx Last sample
=====
^* 169.254.169.123        3   6    17    43   -30us[ -226us] +/-  287us
^- ec2-12-34-231-12.eu-west>  2   6    17    43   -388us[ -388us] +/-   11ms
^- tshirt.heanet.ie       1   6    17    44   +178us[ +25us] +/- 1959us
^? tbag.heanet.ie        0   6     0     -    +0ns[ +0ns] +/-    0ns
^? bray.walcz.net        0   6     0     -    +0ns[ +0ns] +/-    0ns
^? 2a05:d018:c43:e312:ce77:> 0   6     0     -    +0ns[ +0ns] +/-    0ns
^? 2a05:d018:dab:2701:b70:b> 0   6     0     -    +0ns[ +0ns] +/-    0ns
```

In the output that's returned, `^*` indicates the preferred time source.

7. Verify the time synchronization metrics that are reported by `chrony`.

```
[ec2-user ~]$ chronyc tracking
```

```
Reference ID      : A9FEA97B (169.254.169.123)
```

```
Stratum      : 4
Ref time (UTC) : Wed Nov 22 13:18:34 2017
System time   : 0.000000626 seconds slow of NTP time
Last offset    : +0.002852759 seconds
RMS offset     : 0.002852759 seconds
Frequency      : 1.187 ppm fast
Residual freq  : +0.020 ppm
Skew           : 24.388 ppm
Root delay     : 0.000504752 seconds
Root dispersion : 0.001112565 seconds
Update interval : 64.4 seconds
Leap status    : Normal
```

Configuring the Amazon Time Sync Service on Ubuntu

You must edit the `chrony` configuration file to add a server entry for the Amazon Time Sync Service.

To configure your instance to use the Amazon Time Sync Service

1. Connect to your instance and use `apt` to install the `chrony` package.

```
ubuntu:~$ sudo apt install chrony
```

Note

If necessary, update your instance first by running `sudo apt update`.

2. Open the `/etc/chrony/chrony.conf` file using a text editor (such as `vim` or `nano`). Add the following line before any other `server` or `pool` statements that are already present in the file, and save your changes:

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

3. Restart the `chrony` service.

```
ubuntu:~$ sudo /etc/init.d/chrony restart
```

```
[ ok ] Restarting chrony (via systemctl): chrony.service.
```

4. Verify that `chrony` is using the 169.254.169.123 IP address to synchronize the time.

```
ubuntu:~$ chronyc sources -v
```

```
210 Number of sources = 7

.-- Source mode '^' = server, '=' = peer, '#' = local clock.
/.- Source state '*' = current synced, '+' = combined , '-' = not combined,
| / '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||   .- xxxx [ yyyy ] +/- zzzz
||   Reachability register (octal) -.          |   xxxx = adjusted offset,
||   Log2(Polling interval) --.                |   yyyy = measured offset,
||   \                                |   zzzz = estimated error.
||   \                                \
MS Name/IP address      Stratum Poll Reach LastRx Last sample
=====
^* 169.254.169.123          3   6    17    12    +15us[ +57us] +/- 320us
^- tbag.heanet.ie          1   6    17    13   -3488us[-3446us] +/- 1779us
^- ec2-12-34-231-12.eu-west- 2   6    17    13    +893us[ +935us] +/- 7710us
```

```
^? 2a05:d018:c43:e312:ce77:6      0   6      0   10y    +0ns[ +0ns] +/- 0ns
^? 2a05:d018:d34:9000:d8c6:5      0   6      0   10y    +0ns[ +0ns] +/- 0ns
^? tshirt.heanet.ie               0   6      0   10y    +0ns[ +0ns] +/- 0ns
^? bray.walcz.net                0   6      0   10y    +0ns[ +0ns] +/- 0ns
```

In the output that's returned, ^* indicates the preferred time source.

- Verify the time synchronization metrics that are reported by chrony.

```
ubuntu:~$ chronyc tracking
```

```
Reference ID      : 169.254.169.123 (169.254.169.123)
Stratum          : 4
Ref time (UTC)   : Wed Nov 29 07:41:57 2017
System time      : 0.000000011 seconds slow of NTP time
Last offset      : +0.000041659 seconds
RMS offset       : 0.000041659 seconds
Frequency        : 10.141 ppm slow
Residual freq   : +7.557 ppm
Skew             : 2.329 ppm
Root delay       : 0.000544 seconds
Root dispersion  : 0.000631 seconds
Update interval  : 2.0 seconds
Leap status      : Normal
```

Configuring the Amazon Time Sync Service on SUSE Linux

Install chrony from <https://software.opensuse.org/package/chrony>.

Open the /etc/chrony.conf file using a text editor (such as **vim** or **nano**). Verify that the file contains the following line:

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

If this line is not present, add it. Comment out any other server or pool lines. Open yast and enable the chrony service.

Changing the time zone on Amazon Linux

Amazon Linux instances are set to the UTC (Coordinated Universal Time) time zone by default. You can change the time on an instance to the local time or to another time zone in your network.

Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

To change the time zone on an instance

- Identify the time zone to use on the instance. The /usr/share/zoneinfo directory contains a hierarchy of time zone data files. Browse the directory structure at that location to find a file for your time zone.

```
[ec2-user ~]$ ls /usr/share/zoneinfo
Africa      Chile     GB      Indian      Mideast    posixrules  US
America    CST6CDT  GB-Eire  Iran        MST        PRC        UTC
Antarctica Cuba     GMT     iso3166.tab MST7MDT   PST8PDT   WET
Arctic     EET      GMTO    Israel     Navajo    right     W-SU
```

...

Some of the entries at this location are directories (such as America), and these directories contain time zone files for specific cities. Find your city (or a city in your time zone) to use for the instance.

2. Update the /etc/sysconfig/clock file with the new time zone. In this example, we use the time zone data file for Los Angeles, /usr/share/zoneinfo/America/Los_Angeles.
 - a. Open the /etc/sysconfig/clock file with your favorite text editor (such as **vim** or **nano**). You need to use **sudo** with your editor command because /etc/sysconfig/clock is owned by root.

```
[ec2-user ~]$ sudo nano /etc/sysconfig/clock
```

- b. Locate the ZONE entry, and change it to the time zone file (omitting the /usr/share/zoneinfo section of the path). For example, to change to the Los Angeles time zone, change the ZONE entry to the following:

```
ZONE="America/Los_Angeles"
```

Note

Do not change the UTC=true entry to another value. This entry is for the hardware clock, and does not need to be adjusted when you're setting a different time zone on your instance.

- c. Save the file and exit the text editor.
3. Create a symbolic link between /etc/localtime and the time zone file so that the instance finds the time zone file when it references local time information.

```
[ec2-user ~]$ sudo ln -sf /usr/share/zoneinfo/America/Los_Angeles /etc/localtime
```

4. Reboot the system to pick up the new time zone information in all services and applications.

```
[ec2-user ~]$ sudo reboot
```

5. (Optional) Confirm that the current time zone is updated to the new time zone by using the **date** command. The current time zone appears in the output. In the following example, the current time zone is PDT, which refers to the Los Angeles time zone.

```
[ec2-user ~]$ date
Sun Aug 16 05:45:16 PDT 2020
```

Optimizing CPU options

Amazon EC2 instances support multithreading, which enables multiple threads to run concurrently on a single CPU core. Each thread is represented as a virtual CPU (vCPU) on the instance. An instance has a default number of CPU cores, which varies according to instance type. For example, an m5.xlarge instance type has two CPU cores and two threads per core by default—four vCPUs in total.

Note

Each vCPU is a thread of a CPU core, except for T2 instances and instances powered by AWS Graviton2 processors.

In most cases, there is an Amazon EC2 instance type that has a combination of memory and number of vCPUs to suit your workloads. However, you can specify the following CPU options to optimize your instance for specific workloads or business needs:

- **Number of CPU cores:** You can customize the number of CPU cores for the instance. You might do this to potentially optimize the licensing costs of your software with an instance that has sufficient amounts of RAM for memory-intensive workloads but fewer CPU cores.
- **Threads per core:** You can disable multithreading by specifying a single thread per CPU core. You might do this for certain workloads, such as high performance computing (HPC) workloads.

You can specify these CPU options during instance launch. There is no additional or reduced charge for specifying CPU options. You're charged the same as instances that are launched with default CPU options.

Contents

- [Rules for specifying CPU options \(p. 633\)](#)
- [CPU cores and threads per CPU core per instance type \(p. 633\)](#)
- [Specifying CPU options for your instance \(p. 646\)](#)
- [Viewing the CPU options for your instance \(p. 647\)](#)

Rules for specifying CPU options

To specify the CPU options for your instance, be aware of the following rules:

- CPU options can only be specified during instance launch and cannot be modified after launch.
- When you launch an instance, you must specify both the number of CPU cores and threads per core in the request. For example requests, see [Specifying CPU options for your instance \(p. 646\)](#).
- The number of vCPUs for the instance is the number of CPU cores multiplied by the threads per core. To specify a custom number of vCPUs, you must specify a valid number of CPU cores and threads per core for the instance type. You cannot exceed the default number of vCPUs for the instance. For more information, see [CPU cores and threads per CPU core per instance type \(p. 633\)](#).
- To disable multithreading, specify one thread per core.
- When you [change the instance type \(p. 294\)](#) of an existing instance, the CPU options automatically change to the default CPU options for the new instance type.
- The specified CPU options persist after you stop, start, or reboot an instance.

CPU cores and threads per CPU core per instance type

The following tables list the instance types that support specifying CPU options. For each type, the table shows the default and supported number of CPU cores and threads per core.

Accelerated computing instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
f1.2xlarge	8	4	2	1, 2, 3, 4	1, 2
f1.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
f1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
g3.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
g3.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
g3.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
g3s.xlarge	4	2	2	1, 2	1, 2
g4dn.xlarge	4	2	2	1, 2	1, 2
g4dn.2xlarge	8	4	2	1, 2, 3, 4	1, 2
g4dn.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
g4dn.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
g4dn.12xlarge	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
g4dn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
inf1.xlarge	4	2	2	2	1, 2
inf1.2xlarge	8	4	2	2, 4	1, 2
inf1.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
inf1.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
p2.xlarge	4	2	2	1, 2	1, 2
p2.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
p2.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
p3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
p3.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
p3.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
p3dn.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

Compute optimized instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
c4.large	2	1	2	1	1, 2
c4.xlarge	4	2	2	1, 2	1, 2
c4.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c4.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
c4.8xlarge	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
c5.large	2	1	2	1	1, 2
c5.xlarge	4	2	2	2	1, 2
c5.2xlarge	8	4	2	2, 4	1, 2
c5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
c5.9xlarge	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
c5.12xlarge	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
c5.18xlarge	72	36	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36	1, 2
c5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
c5a.large	2	1	2	1	1, 2
c5a.xlarge	4	2	2	1, 2	1, 2
c5a.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c5a.4xlarge	16	8	2	1, 2, 3, 4, 8	1, 2
c5a.8xlarge	32	16	2	1, 2, 3, 4, 8, 12, 16	1, 2
c5a.12xlarge	48	24	2	1, 2, 3, 4, 8, 12, 16, 20, 24	1, 2
c5a.16xlarge	64	32	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1, 2
c5a.24xlarge	96	48	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48	1, 2
c5ad.large	2	1	2	1	1, 2
c5ad.xlarge	4	2	2	1, 2	1, 2
c5ad.2xlarge	8	4	2	1, 2, 3, 4	1, 2
c5ad.4xlarge	16	8	2	1, 2, 3, 4, 8	1, 2
c5ad.8xlarge	32	16	2	1, 2, 3, 4, 8, 12, 16	1, 2
c5ad.12xlarge	48	24	2	1, 2, 3, 4, 8, 12, 16, 20, 24	1, 2
c5ad.16xlarge	64	32	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
c5ad.24xlarge	96	48	2	1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48	1, 2
c5d.large	2	1	2	1	1, 2
c5d.xlarge	4	2	2	2	1, 2
c5d.2xlarge	8	4	2	2, 4	1, 2
c5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
c5d.9xlarge	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
c5d.12xlarge	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
c5d.18xlarge	72	36	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36	1, 2
c5d.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
c5n.large	2	1	2	1	1, 2
c5n.xlarge	4	2	2	2	1, 2
c5n.2xlarge	8	4	2	2, 4	1, 2
c5n.4xlarge	16	8	2	2, 4, 6, 8	1, 2
c5n.9xlarge	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
c5n.18xlarge	72	36	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36	1, 2

General purpose instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
m5.large	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
m5.xlarge	4	2	2	2	1, 2
m5.2xlarge	8	4	2	2, 4	1, 2
m5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
m5a.large	2	1	2	1	1, 2
m5a.xlarge	4	2	2	2	1, 2
m5a.2xlarge	8	4	2	2, 4	1, 2
m5a.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5a.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m5a.12xlarge	48	24	2	6, 12, 18, 24	1, 2
m5a.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5a.24xlarge	96	48	2	12, 18, 24, 36, 48	1, 2
m5ad.large	2	1	2	1	1, 2
m5ad.xlarge	4	2	2	2	1, 2
m5ad.2xlarge	8	4	2	2, 4	1, 2
m5ad.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5ad.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
m5ad.12xlarge	48	24	2	6, 12, 18, 24	1, 2
m5ad.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5ad.24xlarge	96	48	2	12, 18, 24, 36, 48	1, 2
m5d.large	2	1	2	1	1, 2
m5d.xlarge	4	2	2	2	1, 2
m5d.2xlarge	8	4	2	2, 4	1, 2
m5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5d.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5d.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
m5dn.large	2	1	2	1	1, 2
m5dn.xlarge	4	2	2	2	1, 2
m5dn.2xlarge	8	4	2	2, 4	1, 2
m5dn.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5dn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m5dn.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5dn.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
m5dn.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
m5n.large	2	1	2	1	1, 2
m5n.xlarge	4	2	2	2	1, 2
m5n.2xlarge	8	4	2	2, 4	1, 2
m5n.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5n.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
m5n.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5n.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5n.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
t3.nano	2	1	2	1	1, 2
t3.micro	2	1	2	1	1, 2
t3.small	2	1	2	1	1, 2
t3.medium	2	1	2	1	1, 2
t3.large	2	1	2	1	1, 2
t3.xlarge	4	2	2	2	1, 2
t3.2xlarge	8	4	2	2, 4	1, 2
t3a.nano	2	1	2	1	1, 2
t3a.micro	2	1	2	1	1, 2
t3a.small	2	1	2	1	1, 2
t3a.medium	2	1	2	1	1, 2
t3a.large	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
t3a.xlarge	4	2	2	2	1, 2
t3a.2xlarge	8	4	2	2, 4	1, 2

Memory optimized instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
r4.large	2	1	2	1	1, 2
r4.xlarge	4	2	2	1, 2	1, 2
r4.2xlarge	8	4	2	1, 2, 3, 4	1, 2
r4.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
r4.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
r4.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5.large	2	1	2	1	1, 2
r5.xlarge	4	2	2	2	1, 2
r5.2xlarge	8	4	2	2, 4	1, 2
r5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r5.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
r5a.large	2	1	2	1	1, 2
r5a.xlarge	4	2	2	2	1, 2
r5a.2xlarge	8	4	2	2, 4	1, 2
r5a.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5a.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5a.12xlarge	48	24	2	6, 12, 18, 24	1, 2
r5a.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5a.24xlarge	96	48	2	12, 18, 24, 36, 48	1, 2
r5ad.large	2	1	2	1	1, 2
r5ad.xlarge	4	2	2	2	1, 2
r5ad.2xlarge	8	4	2	2, 4	1, 2
r5ad.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5ad.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5ad.12xlarge	48	24	2	6, 12, 18, 24	1, 2
r5ad.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5ad.24xlarge	96	48	2	12, 18, 24, 36, 48	1, 2
r5d.large	2	1	2	1	1, 2
r5d.xlarge	4	2	2	2	1, 2
r5d.2xlarge	8	4	2	2, 4	1, 2
r5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
r5d.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5d.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
r5dn.large	2	1	2	1	1, 2
r5dn.xlarge	4	2	2	2	1, 2
r5dn.2xlarge	8	4	2	2, 4	1, 2
r5dn.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5dn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5dn.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r5dn.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5dn.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
r5n.large	2	1	2	1	1, 2
r5n.xlarge	4	2	2	2	1, 2
r5n.2xlarge	8	4	2	2, 4	1, 2
r5n.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5n.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
r5n.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
r5n.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5n.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
x1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
x1.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
x1e.xlarge	4	2	2	1, 2	1, 2
x1e.2xlarge	8	4	2	1, 2, 3, 4	1, 2
x1e.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
x1e.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
x1e.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
x1e.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
z1d.large	2	1	2	1	1, 2
z1d.xlarge	4	2	2	2	1, 2
z1d.2xlarge	8	4	2	2, 4	1, 2
z1d.3xlarge	12	6	2	2, 4, 6	1, 2
z1d.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
<code>z1d.12xlarge</code>	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

Storage optimized instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
<code>d2.xlarge</code>	4	2	2	1, 2	1, 2
<code>d2.2xlarge</code>	8	4	2	1, 2, 3, 4	1, 2
<code>d2.4xlarge</code>	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
<code>d2.8xlarge</code>	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
<code>h1.2xlarge</code>	8	4	2	1, 2, 3, 4	1, 2
<code>h1.4xlarge</code>	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
<code>h1.8xlarge</code>	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
<code>h1.16xlarge</code>	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
<code>i3.large</code>	2	1	2	1	1, 2
<code>i3.xlarge</code>	4	2	2	1, 2	1, 2
<code>i3.2xlarge</code>	8	4	2	1, 2, 3, 4	1, 2
<code>i3.4xlarge</code>	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
<code>i3.8xlarge</code>	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
<code>i3.16xlarge</code>	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
<code>i3en.large</code>	2	1	2	1	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
i3en.xlarge	4	2	2	2	1, 2
i3en.2xlarge	8	4	2	2, 4	1, 2
i3en.3xlarge	12	6	2	2, 4, 6	1, 2
i3en.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
i3en.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
i3en.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

Specifying CPU options for your instance

You can specify CPU options during instance launch. The following examples are for an `r4.4xlarge` instance type, which has the following [default values \(p. 641\)](#):

- Default CPU cores: 8
- Default threads per core: 2
- Default vCPUs: 16 (8 * 2)
- Valid number of CPU cores: 1, 2, 3, 4, 5, 6, 7, 8
- Valid number of threads per core: 1, 2

Disabling multithreading

To disable multithreading, specify one thread per core.

To disable multithreading during instance launch (console)

1. Follow the [Launching an instance using the Launch Instance Wizard \(p. 499\)](#) procedure.
2. On the **Configure Instance Details** page, for **CPU options**, choose **Specify CPU options**.
3. For **Core count**, choose the number of required CPU cores. In this example, to specify the default CPU core count for an `r4.4xlarge` instance, choose 8.
4. To disable multithreading, for **Threads per core**, choose 1.
5. Continue as prompted by the wizard. When you've finished reviewing your options on the **Review Instance Launch** page, choose **Launch**. For more information, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#).

To disable multithreading during instance launch (AWS CLI)

Use the [run-instances](#) AWS CLI command and specify a value of 1 for `ThreadsPerCore` for the `--cpu-options` parameter. For `CoreCount`, specify the number of CPU cores. In this example, to specify the default CPU core count for an `r4.4xlarge` instance, specify a value of 8.

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options "CoreCount=8,ThreadsPerCore=1" --key-name MyKeyPair
```

Specifying a custom number of vCPUs

You can customize the number of CPU cores and threads per core for the instance.

To specify a custom number of vCPUs during instance launch (console)

The following example launches an `r4.4xlarge` instance with six vCPUs.

1. Follow the [Launching an instance using the Launch Instance Wizard \(p. 499\)](#) procedure.
2. On the **Configure Instance Details** page, for **CPU options**, choose **Specify CPU options**.
3. To get six vCPUs, specify three CPU cores and two threads per core, as follows:
 - For **Core count**, choose **3**.
 - For **Threads per core**, choose **2**.
4. Continue as prompted by the wizard. When you've finished reviewing your options on the **Review Instance Launch** page, choose **Launch**. For more information, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#).

To specify a custom number of vCPUs during instance launch (AWS CLI)

The following example launches an `r4.4xlarge` instance with six vCPUs.

Use the [run-instances](#) AWS CLI command and specify the number of CPU cores and number of threads in the `--cpu-options` parameter. You can specify three CPU cores and two threads per core to get six vCPUs.

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options "CoreCount=3,ThreadsPerCore=2" --key-name MyKeyPair
```

Alternatively, specify six CPU cores and one thread per core (disable multithreading) to get six vCPUs:

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options "CoreCount=6,ThreadsPerCore=1" --key-name MyKeyPair
```

Viewing the CPU options for your instance

You can view the CPU options for an existing instance in the Amazon EC2 console or by describing the instance using the AWS CLI.

New console

To view the CPU options for an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances** and select the instance.
3. On the **Details** tab, under **Host and placement group**, find **Number of vCPUs**.
4. To view core count and threads per core, choose the value for **Number of vCPUs**.

Old console

To view the CPU options for an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances** and select the instance.
3. Choose **Description** and find **Number of vCPUs**.
4. To view core count and threads per core, choose the value for **Number of vCPUs**.

To view the CPU options for an instance (AWS CLI)

Use the [describe-instances](#) command.

```
aws ec2 describe-instances --instance-ids i-123456789abcde123
```

```
...
    "Instances": [
        {
            "Monitoring": {
                "State": "disabled"
            },
            "PublicDnsName": "ec2-198-51-100-5.eu-central-1.compute.amazonaws.com",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "EbsOptimized": false,
            "LaunchTime": "2018-05-08T13:40:33.000Z",
            "PublicIpAddress": "198.51.100.5",
            "PrivateIpAddress": "172.31.2.206",
            "ProductCodes": [],
            "VpcId": "vpc-1a2b3c4d",
            "CpuOptions": {
                "CoreCount": 34,
                "ThreadsPerCore": 1
            },
            "StateTransitionReason": "",
            ...
        }
    ]
...
```

In the output that's returned, the `CoreCount` field indicates the number of cores for the instance. The `ThreadsPerCore` field indicates the number of threads per core.

Alternatively, connect to your instance and use a tool such as `lscpu` to view the CPU information for your instance.

You can use AWS Config to record, assess, audit, and evaluate configuration changes for instances, including terminated instances. For more information, see [Getting Started with AWS Config](#) in the *AWS Config Developer Guide*.

Changing the hostname of your Amazon Linux instance

When you launch an instance, it is assigned a hostname that is a form of the private, internal IPv4 address. A typical Amazon EC2 private DNS name looks something like this: `ip-12-34-56-78.us-`

`west-2.compute.internal`, where the name consists of the internal domain, the service (in this case, `compute`), the region, and a form of the private IPv4 address. Part of this hostname is displayed at the shell prompt when you log into your instance (for example, `ip-12-34-56-78`). Each time you stop and restart your Amazon EC2 instance (unless you are using an Elastic IP address), the public IPv4 address changes, and so does your public DNS name, system hostname, and shell prompt.

Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

Changing the system hostname

If you have a public DNS name registered for the IP address of your instance (such as `webserver.mydomain.com`), you can set the system hostname so your instance identifies itself as a part of that domain. This also changes the shell prompt so that it displays the first portion of this name instead of the hostname supplied by AWS (for example, `ip-12-34-56-78`). If you do not have a public DNS name registered, you can still change the hostname, but the process is a little different.

To change the system hostname to a public DNS name

Follow this procedure if you already have a public DNS name registered.

- For Amazon Linux 2: Use the `hostnamectl` command to set your hostname to reflect the fully qualified domain name (such as `webserver.mydomain.com`).

```
[ec2-user ~]$ sudo hostnamectl set-hostname webserver.mydomain.com
```

- For Amazon Linux AMI: On your instance, open the `/etc/sysconfig/network` configuration file in your favorite text editor and change the `HOSTNAME` entry to reflect the fully qualified domain name (such as `webserver.mydomain.com`).

```
HOSTNAME=webserver.mydomain.com
```

- Reboot the instance to pick up the new hostname.

```
[ec2-user ~]$ sudo reboot
```

Alternatively, you can reboot using the Amazon EC2 console (on the **Instances** page, choose **Actions, Instance State, Reboot**).

- Log into your instance and verify that the hostname has been updated. Your prompt should show the new hostname (up to the first ".") and the `hostname` command should show the fully-qualified domain name.

```
[ec2-user@webserver ~]$ hostname  
webserver.mydomain.com
```

To change the system hostname without a public DNS name

- For Amazon Linux 2: Use the `hostnamectl` command to set your hostname to reflect the desired system hostname (such as `webserver`).

```
[ec2-user ~]$ sudo hostnamectl set-hostname webserver.localdomain
```

- For Amazon Linux AMI: On your instance, open the `/etc/sysconfig/network` configuration file in your favorite text editor and change the `HOSTNAME` entry to reflect the desired system hostname (such as `webserver`).

```
HOSTNAME=webserver.localdomain
```

2. Open the /etc/hosts file in your favorite text editor and change the entry beginning with **127.0.0.1** to match the example below, substituting your own hostname.

```
127.0.0.1 webserver.localdomain webserver localhost4 localhost4.localdomain4
```

3. Reboot the instance to pick up the new hostname.

```
[ec2-user ~]$ sudo reboot
```

Alternatively, you can reboot using the Amazon EC2 console (on the **Instances** page, choose **Actions, Instance State, Reboot**).

4. Log into your instance and verify that the hostname has been updated. Your prompt should show the new hostname (up to the first ".") and the **hostname** command should show the fully-qualified domain name.

```
[ec2-user@webserver ~]$ hostname  
webserver.localdomain
```

Changing the shell prompt without affecting the hostname

If you do not want to modify the hostname for your instance, but you would like to have a more useful system name (such as **webserver**) displayed than the private name supplied by AWS (for example, ip-12-34-56-78), you can edit the shell prompt configuration files to display your system nickname instead of the hostname.

To change the shell prompt to a host nickname

1. Create a file in /etc/profile.d that sets the environment variable called **NICKNAME** to the value you want in the shell prompt. For example, to set the system nickname to **webserver**, run the following command.

```
[ec2-user ~]$ sudo sh -c 'echo "export NICKNAME=webserver" > /etc/profile.d/prompt.sh'
```

2. Open the /etc/bashrc (Red Hat) or /etc/bash.bashrc (Debian/Ubuntu) file in your favorite text editor (such as **vim** or **nano**). You need to use **sudo** with the editor command because /etc/bashrc and /etc/bash.bashrc are owned by **root**.
3. Edit the file and change the shell prompt variable (**PS1**) to display your nickname instead of the hostname. Find the following line that sets the shell prompt in /etc/bashrc or /etc/bash.bashrc (several surrounding lines are shown below for context; look for the line that starts with ["\$PS1"]):

```
# Turn on checkwinsize
shopt -s checkwinsize
[ "$PS1" = "\s-\v\$ " ] && PS1="\u@\h \w\$ "
# You might want to have e.g. tty in prompt (e.g. more virtual machines)
# and console windows
```

Change the **\h** (the symbol for hostname) in that line to the value of the **NICKNAME** variable.

```
# Turn on checkwinsize
shopt -s checkwinsize
```

```
[ "$PS1" = "\s-\v\\\$ " ] && PS1=[\u@\$NICKNAME \w]\$\n# You might want to have e.g. tty in prompt (e.g. more virtual machines)\n# and console windows
```

4. (Optional) To set the title on shell windows to the new nickname, complete the following steps.

- a. Create a file named /etc/sysconfig/bash-prompt-xterm.

```
[ec2-user ~]$ sudo touch /etc/sysconfig/bash-prompt-xterm
```

- b. Make the file executable using the following command.

```
[ec2-user ~]$ sudo chmod +x /etc/sysconfig/bash-prompt-xterm
```

- c. Open the /etc/sysconfig/bash-prompt-xterm file in your favorite text editor (such as **vim** or **nano**). You need to use **sudo** with the editor command because /etc/sysconfig/bash-prompt-xterm is owned by root.
 - d. Add the following line to the file.

```
echo -ne "\033]0;${USER}@${NICKNAME}: ${PWD/#$HOME/~}\007"
```

5. Log out and then log back in to pick up the new nickname value.

Changing the hostname on other Linux distributions

The procedures on this page are intended for use with Amazon Linux only. For more information about other Linux distributions, see their specific documentation and the following articles:

- [How do I assign a static hostname to a private Amazon EC2 instance running RHEL 7 or Centos 7?](#)

Setting up dynamic DNS on Your Amazon Linux instance

When you launch an EC2 instance, it is assigned a public IP address and a public DNS (Domain Name System) name that you can use to reach it from the Internet. Because there are so many hosts in the Amazon Web Services domain, these public names must be quite long for each name to remain unique. A typical Amazon EC2 public DNS name looks something like this: ec2-12-34-56-78.us-west-2.compute.amazonaws.com, where the name consists of the Amazon Web Services domain, the service (in this case, compute), the region, and a form of the public IP address.

Dynamic DNS services provide custom DNS host names within their domain area that can be easy to remember and that can also be more relevant to your host's use case; some of these services are also free of charge. You can use a dynamic DNS provider with Amazon EC2 and configure the instance to update the IP address associated with a public DNS name each time the instance starts. There are many different providers to choose from, and the specific details of choosing a provider and registering a name with them are outside the scope of this guide.

Important

This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

To use dynamic DNS with Amazon EC2

1. Sign up with a dynamic DNS service provider and register a public DNS name with their service. This procedure uses the free service from [noip.com/free](#) as an example.

2. Configure the dynamic DNS update client. After you have a dynamic DNS service provider and a public DNS name registered with their service, point the DNS name to the IP address for your instance. Many providers (including [noip.com](#)) allow you to do this manually from your account page on their website, but many also support software update clients. If an update client is running on your EC2 instance, your dynamic DNS record is updated each time the IP address changes, as after a shutdown and restart. In this example, you install the noip2 client, which works with the service provided by [noip.com](#).
 - a. Enable the Extra Packages for Enterprise Linux (EPEL) repository to gain access to the noip2 client.

Note

Amazon Linux instances have the GPG keys and repository information for the EPEL repository installed by default; however, Red Hat and CentOS instances must first install the epel-release package before you can enable the EPEL repository. For more information and to download the latest version of this package, see <https://fedoraproject.org/wiki/EPEL>.

- For Amazon Linux 2:

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

- For Amazon Linux AMI:

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

- b. Install the noip package.

```
[ec2-user ~]$ sudo yum install -y noip
```

- c. Create the configuration file. Enter the login and password information when prompted and answer the subsequent questions to configure the client.

```
[ec2-user ~]$ sudo noip2 -C
```

3. Enable the noip service.

- For Amazon Linux 2:

```
[ec2-user ~]$ sudo systemctl enable noip.service
```

- For Amazon Linux AMI:

```
[ec2-user ~]$ sudo chkconfig noip on
```

4. Start the noip service.

- For Amazon Linux 2:

```
[ec2-user ~]$ sudo systemctl start noip.service
```

- For Amazon Linux AMI:

```
[ec2-user ~]$ sudo service noip start
```

This command starts the client, which reads the configuration file (`/etc/no-ip2.conf`) that you created earlier and updates the IP address for the public DNS name that you chose.

5. Verify that the update client has set the correct IP address for your dynamic DNS name. Allow a few minutes for the DNS records to update, and then try to connect to your instance using SSH with the public DNS name that you configured in this procedure.

Running commands on your Linux instance at launch

When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts. You can pass two types of user data to Amazon EC2: shell scripts and cloud-init directives. You can also pass this data into the launch wizard as plain text, as a file (this is useful for launching instances using the command line tools), or as base64-encoded text (for API calls).

If you are interested in more complex automation scenarios, consider using AWS CloudFormation and AWS OpsWorks. For more information, see the [AWS CloudFormation User Guide](#) and the [AWS OpsWorks User Guide](#).

For information about running commands on your Windows instance at launch, see [Running Commands on Your Windows Instance at Launch](#) and [Managing Windows Instance Configuration](#) in the *Amazon EC2 User Guide for Windows Instances*.

In the following examples, the commands from the [Install a LAMP Web Server on Amazon Linux 2 \(p. 36\)](#) are converted to a shell script and a set of cloud-init directives that executes when the instance launches. In each example, the following tasks are executed by the user data:

- The distribution software packages are updated.
- The necessary web server, `php`, and `mariadb` packages are installed.
- The `httpd` service is started and turned on via `systemctl`.
- The `ec2-user` is added to the `apache` group.
- The appropriate ownership and file permissions are set for the web directory and the files contained within it.
- A simple web page is created to test the web server and PHP engine.

Contents

- [Prerequisites \(p. 653\)](#)
- [User data and shell scripts \(p. 654\)](#)
- [User data and the console \(p. 654\)](#)
- [User data and cloud-init directives \(p. 655\)](#)
- [User data and the AWS CLI \(p. 657\)](#)

Prerequisites

The following examples assume that your instance has a public DNS name that is reachable from the Internet. For more information, see [Step 1: Launch an instance \(p. 31\)](#). You must also configure your security group to allow SSH (port 22), HTTP (port 80), and HTTPS (port 443) connections. For more information about these prerequisites, see [Setting up with Amazon EC2 \(p. 26\)](#).

Also, these instructions are intended for use with Amazon Linux 2, and the commands and directives may not work for other Linux distributions. For more information about other distributions, such as their support for cloud-init, see their specific documentation.

User data and shell scripts

If you are familiar with shell scripting, this is the easiest and most complete way to send instructions to an instance at launch. Adding these tasks at boot time adds to the amount of time it takes to boot the instance. You should allow a few minutes of extra time for the tasks to complete before you test that the user script has finished successfully.

Important

By default, user data scripts and cloud-init directives run only during the boot cycle when you first launch an instance. You can update your configuration to ensure that your user data scripts and cloud-init directives run every time you restart your instance. For more information, see [How can I execute user data with every restart of my EC2 instance?](#) in the AWS Knowledge Center.

User data shell scripts must start with the `#!` characters and the path to the interpreter you want to read the script (commonly `/bin/bash`). For a great introduction on shell scripting, see [the BASH Programming HOW-TO](#) at the Linux Documentation Project (tldp.org).

Scripts entered as user data are executed as the `root` user, so do not use the `sudo` command in the script. Remember that any files you create will be owned by `root`; if you need non-root users to have file access, you should modify the permissions accordingly in the script. Also, because the script is not run interactively, you cannot include commands that require user feedback (such as `yum update` without the `-y` flag).

If you use an AWS API, including the AWS CLI, in a user data script, you must use an instance profile when launching the instance. An instance profile provides the appropriate AWS credentials required by the user data script to execute the API call. For more information, see [Using instance profiles](#) in the IAM User Guide. The permissions you assign to the IAM role depend on which services you are calling with the API. For more information, see [IAM roles for Amazon EC2](#).

The cloud-init output log file (`/var/log/cloud-init-output.log`) captures console output so it is easy to debug your scripts following a launch if the instance does not behave the way you intended.

When a user data script is processed, it is copied to and executed from `/var/lib/cloud/instances/instance-id/`. The script is not deleted after it is run. Be sure to delete the user data scripts from `/var/lib/cloud/instances/instance-id/` before you create an AMI from the instance. Otherwise, the script will exist in this directory on any instance launched from the AMI.

User data and the console

You can specify instance user data when you launch the instance. If the root volume of the instance is an EBS volume, you can also stop the instance and update its user data.

Specify instance user data at launch

Follow the procedure for launching an instance at [Launching an instance using the Launch Instance Wizard \(p. 499\)](#), but when you get to the section called ["Step 3: Configure Instance Details" \(p. 500\)](#) in that procedure, copy your shell script in the **User data** field, and then complete the launch procedure.

In the example script below, the script creates and configures our web server.

```
#!/bin/bash
yum update -y
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
yum install -y httpd mariadb-server
systemctl start httpd
systemctl enable httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
```

```
find /var/www -type f -exec chmod 0644 {} \;
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

Allow enough time for the instance to launch and execute the commands in your script, and then check to see that your script has completed the tasks that you intended.

For our example, in a web browser, enter the URL of the PHP test file the script created. This URL is the public DNS address of your instance followed by a forward slash and the file name.

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

You should see the PHP information page. If you are unable to see the PHP information page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic. For more information, see [Adding rules to a security group \(p. 1006\)](#).

(Optional) If your script did not accomplish the tasks you were expecting it to, or if you just want to verify that your script completed without errors, examine the cloud-init output log file at `/var/log/cloud-init-output.log` and look for error messages in the output.

For additional debugging information, you can create a Mime multipart archive that includes a cloud-init data section with the following directive:

```
output : { all : '| tee -a /var/log/cloud-init-output.log' }
```

This directive sends command output from your script to `/var/log/cloud-init-output.log`. For more information about cloud-init data formats and creating Mime multi part archive, see [cloud-init Formats](#).

View and update the instance user data

To modify instance user data

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, Instance State, Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

4. When prompted for confirmation, choose **Yes, Stop**. It can take a few minutes for the instance to stop.
5. With the instance still selected, choose **Actions, Instance Settings, View/Change User Data**. You can't change the user data if the instance is running, but you can view it.
6. In the **View/Change User Data** dialog box, update the user data, and then choose **Save**.
7. Restart the instance. The new user data is visible on your instance after you restart it; however, user data scripts are not executed.

User data and cloud-init directives

The cloud-init package configures specific aspects of a new Amazon Linux instance when it is launched; most notably, it configures the `.ssh/authorized_keys` file for the `ec2-user` so you can log in with your own private key. For more information, see [cloud-init \(p. 181\)](#).

The cloud-init user directives can be passed to an instance at launch the same way that a script is passed, although the syntax is different. For more information about cloud-init, go to <http://cloudinit.readthedocs.org/en/latest/index.html>.

Important

By default, user data scripts and cloud-init directives run only during the boot cycle when you first launch an instance. You can update your configuration to ensure that your user data scripts and cloud-init directives run every time you restart your instance. For more information, see [How can I execute user data with every restart of my EC2 instance?](#) in the AWS Knowledge Center.

Adding these tasks at boot time adds to the amount of time it takes to boot an instance. You should allow a few minutes of extra time for the tasks to complete before you test that your user data directives have completed.

To pass cloud-init directives to an instance with user data

1. Follow the procedure for launching an instance at [Launching an instance using the Launch Instance Wizard \(p. 499\)](#), but when you get to the section called “Step 3: Configure Instance Details” (p. 500) in that procedure, enter your cloud-init directive text in the **User data** field, and then complete the launch procedure.

In the example below, the directives create and configure a web server on Amazon Linux 2. The `#cloud-config` line at the top is required in order to identify the commands as cloud-init directives.

```
#cloud-config
repo_update: true
repo_upgrade: all

packages:
- httpd
- mariadb-server

runcmd:
- [ sh, -c, "amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2" ]
- systemctl start httpd
- sudo systemctl enable httpd
- [ sh, -c, "usermod -a -G apache ec2-user" ]
- [ sh, -c, "chown -R ec2-user:apache /var/www" ]
- chmod 2775 /var/www
- [ find, /var/www, -type, d, -exec, chmod, 2775, {}, \; ]
- [ find, /var/www, -type, f, -exec, chmod, 0664, {}, \; ]
- [ sh, -c, 'echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php' ]
```

2. Allow enough time for the instance to launch and execute the directives in your user data, and then check to see that your directives have completed the tasks you intended.

For our example, in a web browser, enter the URL of the PHP test file the directives created. This URL is the public DNS address of your instance followed by a forward slash and the file name.

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

You should see the PHP information page. If you are unable to see the PHP information page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic. For more information, see [Adding rules to a security group \(p. 1006\)](#).

3. (Optional) If your directives did not accomplish the tasks you were expecting them to, or if you just want to verify that your directives completed without errors, examine the output log file at `/var/log/cloud-init-output.log` and look for error messages in the output. For additional debugging information, you can add the following line to your directives:

```
output : { all : '| tee -a /var/log/cloud-init-output.log' }
```

This directive sends **runcmd** output to `/var/log/cloud-init-output.log`.

User data and the AWS CLI

You can use the AWS CLI to specify, modify, and view the user data for your instance. For information about viewing user data from your instance using instance metadata, see [Retrieve instance user data \(p. 673\)](#).

On Windows, you can use the AWS Tools for Windows PowerShell instead of using the AWS CLI. For more information, see [User Data and the Tools for Windows PowerShell](#) in the *Amazon EC2 User Guide for Windows Instances*.

Example: Specify user data at launch

To specify user data when you launch your instance, use the [run-instances](#) command with the `--user-data` parameter. With **run-instances**, the AWS CLI performs base64 encoding of the user data for you.

The following example shows how to specify a script as a string on the command line:

```
aws ec2 run-instances --image-id ami-abcd1234 --count 1 --instance-type m3.medium \
--key-name my-key-pair --subnet-id subnet-abcd1234 --security-group-ids sg-abcd1234 \
--user-data echo user data
```

The following example shows how to specify a script using a text file. Be sure to use the `file://` prefix to specify the file.

```
aws ec2 run-instances --image-id ami-abcd1234 --count 1 --instance-type m3.medium \
--key-name my-key-pair --subnet-id subnet-abcd1234 --security-group-ids sg-abcd1234 \
--user-data file:///my_script.txt
```

The following is an example text file with a shell script.

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

Example: Modify the user data of a stopped instance

You can modify the user data of a stopped instance using the [modify-instance-attribute](#) command. With **modify-instance-attribute**, the AWS CLI does not perform base64 encoding of the user data for you.

- On a **Linux** computer, use the `base64` command to encode the user data.

```
base64 my_script.txt >my_script_base64.txt
```

- On a **Windows** computer, use the `certutil` command to encode the user data. Before you can use this file with the AWS CLI, you must remove the first (BEGIN CERTIFICATE) and last (END CERTIFICATE) lines.

```
certutil -encode my_script.txt my_script_base64.txt
notepad my_script_base64.txt
```

Use the `--attribute` and `--value` parameters to use the encoded text file to specify the user data. Be sure to use the `file://` prefix to specify the file.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --attribute userData --value file://my_script_base64.txt
```

Example: Clear the user data of a stopped instance

To delete the existing user data, use the [modify-instance-attribute](#) command as follows:

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --user-data Value=
```

Example: View user data

To retrieve the user data for an instance, use the [describe-instance-attribute](#) command. With [describe-instance-attribute](#), the AWS CLI does not perform base64 decoding of the user data for you.

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute userData
```

The following is example output with the user data base64 encoded.

```
{  
    "UserData": {  
        "Value":  
            "IyEvYmluL2Jhc2gKeXVtIHVwZGF0ZSAtOpzZXJ2aN1IGH0dHBkIHN0YXJ0CmNoa2NvbmcZpZyBodHRwZCBvbg=="  
    },  
    "InstanceId": "i-1234567890abcdef0"  
}
```

- On a **Linux** computer, use the `--query` option to get the encoded user data and the `base64` command to decode it.

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute userData --output text --query "UserData.Value" | base64 --decode
```

- On a **Windows** computer, use the `--query` option to get the coded user data and the `certutil` command to decode it. Note that the encoded output is stored in a file and the decoded output is stored in another file.

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute userData --output text --query "UserData.Value" >my_output.txt  
certutil -decode my_output.txt my_output_decoded.txt  
type my_output_decoded.txt
```

The following is example output.

```
#!/bin/bash  
yum update -y  
service httpd start  
chkconfig httpd on
```

Instance metadata and user data

Instance metadata is data about your instance that you can use to configure or manage the running instance. Instance metadata is divided into categories, for example, host name, events, and security groups.

You can also use instance metadata to access *user data* that you specified when launching your instance. For example, you can specify parameters for configuring your instance, or include a simple script. You

can build generic AMIs and use user data to modify the configuration files supplied at launch time. For example, if you run web servers for various small businesses, they can all use the same generic AMI and retrieve their content from the Amazon S3 bucket that you specify in the user data at launch. To add a new customer at any time, create a bucket for the customer, add their content, and launch your AMI with the unique bucket name provided to your code in the user data. If you launch more than one instance at the same time, the user data is available to all instances in that reservation. Each instance that is part of the same reservation has a unique `ami-launch-index` number, allowing you to write code that controls what to do. For example, the first host might elect itself as the original node in a cluster. For a detailed AMI launch example, see [Example: AMI launch index value \(p. 674\)](#).

EC2 instances can also include *dynamic data*, such as an instance identity document that is generated when the instance is launched. For more information, see [Dynamic data categories \(p. 683\)](#).

Important

Although you can only access instance metadata and user data from within the instance itself, the data is not protected by authentication or cryptographic methods. Anyone who has direct access to the instance, and potentially any software running on the instance, can view its metadata. Therefore, you should not store sensitive data, such as passwords or long-lived encryption keys, as user data.

Contents

- [Configuring the instance metadata service \(p. 659\)](#)
- [Retrieving instance metadata \(p. 665\)](#)
- [Working with instance user data \(p. 672\)](#)
- [Retrieving dynamic data \(p. 674\)](#)
- [Example: AMI launch index value \(p. 674\)](#)
- [Instance metadata categories \(p. 677\)](#)
- [Instance identity documents \(p. 684\)](#)

Configuring the instance metadata service

You can access instance metadata from a running instance using one of the following methods:

- Instance Metadata Service Version 1 (IMDSv1) – a request/response method
- Instance Metadata Service Version 2 (IMDSv2) – a session-oriented method

By default, you can use either IMDSv1 or IMDSv2, or both. The instance metadata service distinguishes between IMDSv1 and IMDSv2 requests based on whether, for any given request, either the `PUT` or `GET` headers, which are unique to IMDSv2, are present in that request.

You can configure the instance metadata service on each instance such that local code or users must use IMDSv2. When you specify that IMDSv2 must be used, IMDSv1 no longer works. For more information, see [Configuring the instance metadata options \(p. 662\)](#).

To retrieve instance metadata, see [Retrieving instance metadata \(p. 665\)](#).

How Instance Metadata Service Version 2 works

IMDSv2 uses session-oriented requests. With session-oriented requests, you create a session token that defines the session duration, which can be a minimum of one second and a maximum of six hours. During the specified duration, you can use the same session token for subsequent requests. After the specified duration expires, you must create a new session token to use for future requests.

The following example uses a Linux shell script and IMDSv2 to retrieve the top-level instance metadata items. The example command:

- Creates a session token lasting six hours (21,600 seconds) using the `PUT` request
- Stores the session token header in a variable named `TOKEN`
- Requests the top-level metadata items using the token

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/
```

After you've created a token, you can reuse it until it expires. In the following example command, which gets the ID of the AMI used to launch the instance, the token that is stored in `$TOKEN` in the previous example is reused.

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/ami-id
```

When you use IMDSv2 to request instance metadata, the request must include the following:

1. Use a `PUT` request to initiate a session to the instance metadata service. The `PUT` request returns a token that must be included in subsequent `GET` requests to the instance metadata service. The token is required to access metadata using IMDSv2.
2. Include the token in all `GET` requests to the instance metadata service. When token usage is set to `required`, requests without a valid token or with an expired token receive a `401 - Unauthorized` HTTP error code. For information about changing the token usage requirement, see [modify-instance-metadata-options](#) in the *AWS CLI Command Reference*.
 - The token is an instance-specific key. The token is not valid on other EC2 instances and will be rejected if you attempt to use it outside of the instance on which it was generated.
 - The `PUT` request must include a header that specifies the time to live (TTL) for the token, in seconds, up to a maximum of six hours (21,600 seconds). The token represents a logical session. The TTL specifies the length of time that the token is valid and, therefore, the duration of the session.
 - After a token expires, to continue accessing instance metadata, you must create a new session using another `PUT`.
 - You can choose to reuse a token or create a new token with every request. For a small number of requests, it might be easier to generate and immediately use a token each time you need to access the instance metadata service. But for efficiency, you can specify a longer duration for the token and reuse it rather than having to write a `PUT` request every time you need to request instance metadata. There is no practical limit on the number of concurrent tokens, each representing its own session. IMDSv2 is, however, still constrained by normal instance metadata service connection and throttling limits. For more information, see [Throttling \(p. 671\)](#).

HTTP `GET` and `HEAD` methods are allowed in IMDSv2 instance metadata requests. `PUT` requests are rejected if they contain an `X-Forwarded-For` header.

By default, the response to `PUT` requests has a response hop limit (time to live) of 1 at the IP protocol level. You can adjust the hop limit using the `modify-instance-metadata-options` command if you need to make it larger. For example, you might need a larger hop limit for backward compatibility with container services running on the instance. For more information, see [modify-instance-metadata-options](#) in the *AWS CLI Command Reference*.

Transitioning to using Instance Metadata Service Version 2

Use of Instance Metadata Service Version 2 (IMDSv2) is optional. Instance Metadata Service Version 1 (IMDSv1) will continue to be supported indefinitely. If you choose to migrate to using IMDSv2, we recommend that you use the following tools and transition path.

Tools for helping with the transition to IMDSv2

If your software uses IMDSv1, use the following tools to help reconfigure your software to use IMDSv2.

- **AWS software:** The latest versions of the AWS SDKs and CLIs support IMDSv2. To use IMDSv2, make sure that your EC2 instances have the latest versions of the AWS SDKs and CLIs. For information about updating the CLI, see [Upgrading to the latest version of the AWS CLI](#) in the *AWS Command Line Interface User Guide*.
- **CloudWatch:** IMDSv2 uses token-backed sessions, while IMDSv1 does not. The `MetadataNoToken` CloudWatch metric tracks the number of calls to the instance metadata service that are using IMDSv1. By tracking this metric to zero, you can determine if and when all of your software has been upgraded to use IMDSv2. For more information, see [Instance metrics \(p. 718\)](#).
- **Updates to EC2 APIs and CLIs:** For existing instances, you can use the [modify-instance-metadata-options](#) CLI command (or the `ModifyInstanceMetadataOptions` API) to require the use of IMDSv2. For new instances, you can use the [run-instances](#) CLI command (or the `RunInstances` API) and the `metadata-options` parameter to launch new instances that require the use of IMDSv2.

To require the use of IMDSv2 on all new instances launched by Auto Scaling groups, your Auto Scaling groups can use either a launch template or a launch configuration. When you [create a launch template](#) or [create a launch configuration](#), you must configure the `MetadataOptions` parameters to require the use IMDSv2. After you configure the launch template or launch configuration, the Auto Scaling group launches new instances using the new launch template or launch configuration, but existing instances are not affected.

Use the [modify-instance-metadata-options](#) CLI command (or the `ModifyInstanceMetadataOptions` API) to require the use of IMDSv2 on the existing instances, or terminate the instances and the Auto Scaling group will launch new replacement instances with the instance metadata options settings that are defined in the launch template or launch configuration.

For Auto Scaling groups that use launch configurations, you can [replace the launch configurations with launch templates](#).

- **IAM policies and SCPs:** You can use an IAM condition to enforce that IAM users can't launch an instance unless it uses IMDSv2. You can also use IAM conditions to enforce that IAM users can't modify running instances to re-enable IMDSv1, and to enforce that the instance metadata service is available on the instance.

The `ec2:MetadataHttpTokens`, `ec2:MetadataHttpPutResponseHopLimit`, and `ec2:MetadataHttpEndpoint` IAM condition keys can be used to control the use of the `RunInstances` and the [ModifyInstanceMetadataOptions](#) API and corresponding CLI. If a policy is created, and a parameter in the API call does not match the state specified in the policy using the condition key, the API or CLI call fails with an `UnauthorizedOperation` response. These condition keys can be used either in IAM policies or AWS Organizations service control policies (SCPs).

Furthermore, you can choose an additional layer of protection to enforce the change from IMDSv1 to IMDSv2. At the access management layer with respect to the APIs called via EC2 Role credentials, you can use a new condition key in either IAM policies or AWS Organizations service control policies (SCPs). Specifically, by using the policy condition key `ec2:RoleDelivery` with a value of `2.0` in your IAM policies, API calls made with EC2 Role credentials obtained from IMDSv1 will receive an `UnauthorizedOperation` response. The same thing can be achieved more broadly with that condition required by an SCP. This ensures that credentials delivered via IMDSv1 cannot actually be used to call APIs because any API calls not matching the specified condition will receive an `UnauthorizedOperation` error. For example IAM policies, see [Working with instance metadata \(p. 963\)](#). For more information, see [Service Control Policies](#) in the *AWS Organizations User Guide*.

Recommended path to requiring IMDSv2 access

Using the above tools, we recommend that you follow this path for transitioning to IMDSv2:

Step 1: At the start

Update the SDKs, CLIs, and your software that use Role credentials on their EC2 instances to IMDSv2-compatible versions. For information about updating the CLI, see [Upgrading to the latest version of the AWS CLI in the AWS Command Line Interface User Guide](#).

Then, change your software that directly accesses instance metadata (in other words, that does not use an SDK) using the IMDSv2 requests.

Step 2: During the transition

Track your transition progress by using the CloudWatch metric `MetadataNoToken`. This metric shows the number of calls to the instance metadata service that are using IMDSv1 on your instances. For more information, see [Instance metrics \(p. 718\)](#).

Step 3: When everything is ready on all instances

Everything is ready on all instances when the CloudWatch metric `MetadataNoToken` records zero IMDSv1 usage. At this stage, you can do the following:

- For existing instances: You can require IMDSv2 use through the [modify-instance-metadata-options](#) command. You can make these changes on running instances; you do not need to restart your instances.
- For new instances: When launching a new instance, you can do one of the following:
 - In the Amazon EC2 console launch instance wizard, set **Metadata accessible** to **Enabled** and **Metadata version** to **V2**. For more information, see [Step 3: Configure Instance Details \(p. 500\)](#).
 - Use the [run-instances](#) command to specify that only IMDSv2 is to be used.

Updating instance metadata options for existing instances is available only through the API or AWS CLI. It is currently not available in the Amazon EC2 console. For more information, see [Configuring the instance metadata options \(p. 662\)](#).

Step 4: When all of your instances are transitioned to IMDSv2

The `ec2:MetadataHttpTokens`, `ec2:MetadataHttpPutResponseHopLimit`, and `ec2:MetadataHttpEndpoint` IAM condition keys can be used to control the use of the [RunInstances](#) and the [ModifyInstanceMetadataOptions](#) API and corresponding CLI. If a policy is created, and a parameter in the API call does not match the state specified in the policy using the condition key, the API or CLI call fails with an `UnauthorizedOperation` response. For example IAM policies, see [Working with instance metadata \(p. 963\)](#).

Configuring the instance metadata options

Instance metadata options allow you to configure new or existing instances to do the following:

- Require the use of IMDSv2 when requesting instance metadata
- Specify the `PUT` response hop limit
- Turn off access to instance metadata

You can also use IAM condition keys in an IAM policy or SCP to do the following:

- Allow an instance to launch only if it's configured to require the use of IMDSv2
- Restrict the number of allowed hops

- Turn off access to instance metadata

You can configure instance metadata options when launching new instances from the Amazon EC2 console. For more information, see [Step 3: Configure Instance Details \(p. 500\)](#).

To configure the instance metadata options on new or existing instances, you can use the AWS SDK or AWS CLI. For more information, see [run-instances](#) and [modify-instance-metadata-options](#) in the *AWS CLI Command Reference*.

Note

You should proceed cautiously and conduct careful testing before making any changes. Take note of the following:

- If you enforce the use of IMDSv2, applications or agents that use IMDSv1 for instance metadata access will break.
- If you turn off all access to instance metadata, applications or agents that rely on instance metadata access to function will break.

Topics

- [Configuring instance metadata options for new instances \(p. 663\)](#)
- [Configuring instance metadata options for existing instances \(p. 664\)](#)

Configuring instance metadata options for new instances

You can require the use of IMDSv2 on an instance when you launch it. You can also create an IAM policy that prevents users from launching new instances unless they require IMDSv2 on the new instance.

Console

To require the use of IMDSv2 on a new instance

- When launching a new instance in the Amazon EC2 console, select the following options on the **Configure Instance Details** page:
 - Under **Advanced Details**, for **Metadata accessible**, select **Enabled**.
 - For **Metadata version**, select **V2 (token required)**.

For more information, see [Step 3: Configure Instance Details \(p. 500\)](#).

AWS CLI

To require the use of IMDSv2 on a new instance

The following [run-instances](#) example launches a `c3.large` instance with `--metadata-options` set to `HttpTokens=required`. When you specify a value for `HttpTokens`, you must also set `HttpEndpoint` to `enabled`. Because the secure token header is set to `required` for metadata retrieval requests, this opts in the instance to require using IMDSv2 when requesting instance metadata.

```
aws ec2 run-instances
  --image-id ami-0abcdef1234567890
  --instance-type c3.large
  ...
  --metadata-options "HttpEndpoint=enabled,HttpTokens=required"
```

To enforce the use of IMDSv2 on all new instances

To ensure that IAM users can only launch instances that require the use of IMDSv2 when requesting instance metadata, you can specify that the condition to require IMDSv2 must be met before an instance can be launched. For the example IAM policy, see [Working with instance metadata \(p. 963\)](#).

Console

To turn off access to instance metadata

- To ensure that access to your instance metadata is turned off, regardless of which version of the instance metadata service you are using, launch the instance in the Amazon EC2 console with the following option selected on the **Configure Instance Details** page:
 - Under **Advanced Details**, for **Metadata accessible**, select **Disabled**.

For more information, see [Step 3: Configure Instance Details \(p. 500\)](#).

AWS CLI

To turn off access to instance metadata

To ensure that access to your instance metadata is turned off, regardless of which version of the instance metadata service you are using, launch the instance with `--metadata-options` set to `HttpEndpoint=disabled`. You can turn access on later by using the [modify-instance-metadata-options](#) command.

```
aws ec2 run-instances
  --image-id ami-0abcdef1234567890
  --instance-type c3.large
  ...
  --metadata-options "HttpEndpoint=disabled"
```

Configuring instance metadata options for existing instances

You can require the use IMDSv2 on an existing instance. You can also change the PUT response hop limit and turn off access to instance metadata on an existing instance. You can also create an IAM policy that prevents users from modifying the instance metadata options on an existing instance.

To require the use of IMDSv2

You can opt in to require that IMDSv2 is used when requesting instance metadata. Use the [modify-instance-metadata-options](#) CLI command and set the `http-tokens` parameter to `required`. When you specify a value for `http-tokens`, you must also set `http-endpoint` to `enabled`.

```
aws ec2 modify-instance-metadata-options \
  --instance-id i-1234567898abcdef0 \
  --http-tokens required \
  --http-endpoint enabled
```

To change the PUT response hop limit

For existing instances, you can change the settings of the PUT response hop limit. Use the [modify-instance-metadata-options](#) CLI command and set the `http-put-response-hop-limit` parameter to the required number of hops. In the following example, the hop limit is set to 3. Note that when specifying a value for `http-put-response-hop-limit`, you must also set `http-endpoint` to `enabled`.

```
aws ec2 modify-instance-metadata-options \
  --instance-id i-1234567898abcdef0 \
```

```
--http-put-response-hop-limit 3 \
--http-endpoint enabled
```

To restore the use of IMDSv1 on an instance using IMDSv2

You can use the [modify-instance-metadata-options](#) CLI command with `http-tokens` set to `optional` to restore the use of IMDSv1 when requesting instance metadata.

```
aws ec2 modify-instance-metadata-options \
--instance-id i-1234567898abcdef0 \
--http-tokens optional \
--http-endpoint enabled
```

To turn off access to instance metadata

You can turn off access to your instance metadata by disabling the HTTP endpoint of the instance metadata service, regardless of which version of the instance metadata service you are using. You can reverse this change at any time by enabling the HTTP endpoint. Use the [modify-instance-metadata-options](#) CLI command and set the `http-endpoint` parameter to `disabled`.

```
aws ec2 modify-instance-metadata-options \
--instance-id i-1234567898abcdef0 \
--http-endpoint disabled
```

To control the use of `modify-instance-metadata-options`

To control which IAM users can modify the instance metadata options, specify a policy that prevents all users other than users with a specified role to use the [ModifyInstanceMetadataOptions](#) API. For the example IAM policy, see [Working with instance metadata \(p. 963\)](#).

Retrieving instance metadata

Because your instance metadata is available from your running instance, you do not need to use the Amazon EC2 console or the AWS CLI. This can be helpful when you're writing scripts to run from your instance. For example, you can access the local IP address of your instance from instance metadata to manage a connection to an external application.

Instance metadata is divided into categories. For a description of each instance metadata category, see [Instance metadata categories \(p. 677\)](#).

To view all categories of instance metadata from within a running instance, use the following URI.

```
http://169.254.169.254/latest/meta-data/
```

The IP address 169.254.169.254 is a link-local address and is valid only from the instance. For more information, see [Link-local address](#) on Wikipedia.

Note that you are not billed for HTTP requests used to retrieve instance metadata and user data.

The command format is different, depending on whether you use IMDSv1 or IMDSv2. By default, you can use both instance metadata services. To require the use of IMDSv2, see [Configuring the instance metadata service \(p. 659\)](#).

You can use a tool such as cURL, as shown in the following example.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
```

```
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
```

You can also download the [Instance Metadata Query tool](#), which allows you to query the instance metadata using Instance Metadata Service Version 1 without having to enter the full URI or category names.

Responses and error messages

All instance metadata is returned as text (HTTP content type `text/plain`).

A request for a specific metadata resource returns the appropriate value, or a `404 – Not Found` HTTP error code if the resource is not available.

A request for a general metadata resource (the URI ends with a `/`) returns a list of available resources, or a `404 – Not Found` HTTP error code if there is no such resource. The list items are on separate lines, terminated by line feeds (ASCII 10).

For requests made using Instance Metadata Service Version 2, the following HTTP error codes can be returned:

- `400 – Missing or Invalid Parameters` – The `PUT` request is not valid.
- `401 – Unauthorized` – The `GET` request uses an invalid token. The recommended action is to generate a new token.
- `403 – Forbidden` – The request is not allowed or the instance metadata service is turned off.

Examples of retrieving instance metadata

Examples

- [Get the available versions of the instance metadata \(p. 666\)](#)
- [Get the top-level metadata items \(p. 667\)](#)
- [Get the list of available public keys \(p. 669\)](#)
- [Show the formats in which public key 0 is available \(p. 670\)](#)
- [Get public key 0 \(in the OpenSSH key format\) \(p. 670\)](#)
- [Get the subnet ID for an instance \(p. 671\)](#)

Get the available versions of the instance metadata

This example gets the available versions of the instance metadata. These versions do not necessarily correlate with an Amazon EC2 API version. The earlier versions are available to you in case you have scripts that rely on the structure and information present in a previous version.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/1.0
```

```
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
2016-06-30
2016-09-02
latest
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
2016-06-30
2016-09-02
latest
```

Get the top-level metadata items

This example gets the top-level metadata items. For more information, see [Instance metadata categories \(p. 677\)](#).

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
instance-action
instance-id
```

```
instance-life-cycle
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

The following examples get the values of some of the top-level metadata items that were obtained in the preceding example. The IMDSv2 requests use the stored token that was created in the preceding example command, assuming it has not expired.

IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
latest/meta-data/ami-id
ami-0abcdef1234567890
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/reservation-id  
r-0efghijk987654321
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/reservation-id  
r-0efghijk987654321
```

IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/local-hostname  
ip-10-251-50-12.ec2.internal
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-hostname  
ip-10-251-50-12.ec2.internal
```

IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/public-hostname  
ec2-203-0-113-25.compute-1.amazonaws.com
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-hostname  
ec2-203-0-113-25.compute-1.amazonaws.com
```

Get the list of available public keys

This example gets the list of available public keys.

IMDSv2

```
[ec2-user ~]$ `curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \  
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/public-keys/  
0=my-public-key
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/
```

```
0=my-public-key
```

Show the formats in which public key 0 is available

This example shows the formats in which public key 0 is available.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/public-keys/0/
openssh-key
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/
openssh-key
```

Get public key 0 (in the OpenSSH key format)

This example gets public key 0 (in the OpenSSH key format).

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCACFICQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMx
CzAJBgNVBAgTAldBMRawDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRawDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvCNQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvCNQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJ1lJ00zbhNY5f6GuoEDmPfJ10ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCACFICQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMx
CzAJBgNVBAgTAldBMRawDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRawDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvCNQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvCNQEFBQADgYEAtCu4
```

```
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb  
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp378OD8uTs7fLvjx79LjSTb  
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

Get the subnet ID for an instance

This example gets the subnet ID for an instance.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" `&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id  
subnet-be9b61d7
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id  
subnet-be9b61d7
```

Throttling

We throttle queries to the instance metadata service on a per-instance basis, and we place limits on the number of simultaneous connections from an instance to the instance metadata service.

If you're using the instance metadata service to retrieve AWS security credentials, avoid querying for credentials during every transaction or concurrently from a high number of threads or processes, as this might lead to throttling. Instead, we recommend that you cache the credentials until they start approaching their expiry time.

If you are throttled while accessing the instance metadata service, retry your query with an exponential backoff strategy.

Limiting instance metadata service access

You can consider using local firewall rules to disable access from some or all processes to the instance metadata service.

Using iptables to limit access

The following example uses Linux iptables and its `owner` module to prevent the Apache webserver (based on its default installation user ID of `apache`) from accessing 169.254.169.254. It uses a *deny rule* to reject all instance metadata requests (whether IMDSv1 or IMDSv2) from any process running as that user.

```
$ sudo iptables --append OUTPUT --proto tcp --destination 169.254.169.254 --match owner --uid-owner apache --jump REJECT
```

Or, you can consider only allowing access to particular users or groups, by using *allow rules*. Allow rules might be easier to manage from a security perspective, because they require you to make a decision about what software needs access to instance metadata. If you use *allow rules*, it's less likely you will accidentally allow software to access the metadata service (that you did not intend to have access) if you later change the software or configuration on an instance. You can also combine group usage with

allow rules, so that you can add and remove users from a permitted group without needing to change the firewall rule.

The following example prevents access to the instance metadata service by all processes, except for processes running in the user account `trustworthy-user`.

```
$ sudo iptables --append OUTPUT --proto tcp --destination 169.254.169.254 --match owner !  
--uid-owner trustworthy-user --jump REJECT
```

Note

- To use local firewall rules, you need to adapt the preceding example commands to suit your needs.
- By default, iptables rules are not persistent across system reboots. They can be made to be persistent by using OS features, not described here.
- The iptables owner module only matches group membership if the group is the primary group of a given local user. Other groups are not matched.

Using PF or IPFW to limit access

If you are using FreeBSD or OpenBSD, you can also consider using PF or IPFW. The following examples limit access to the instance metadata service to just the root user.

PF

```
$ block out inet proto tcp from any to 169.254.169.254
```

```
$ pass out inet proto tcp from any to 169.254.169.254 user root
```

IPFW

```
$ allow tcp from any to 169.254.169.254 uid root
```

```
$ deny tcp from any to 169.254.169.254
```

Note

The order of the PF and IPFW commands matter. PF defaults to last matching rule and IPFW defaults to first matching rule.

Working with instance user data

When working with instance user data, keep the following in mind:

- User data must be base64-encoded. The Amazon EC2 console can perform the base64-encoding for you or accept base64-encoded input.
- User data is limited to 16 KB, in raw form, before it is base64-encoded. The size of a string of length n after base64-encoding is $\text{ceil}(n/3)*4$.
- User data must be base64-decoded when you retrieve it. If you retrieve the data using instance metadata or the console, it's decoded for you automatically.
- User data is treated as opaque data: what you give is what you get back. It is up to the instance to be able to interpret it.
- If you stop an instance, modify its user data, and start the instance, the updated user data is not executed when you start the instance.

Specify instance user data at launch

You can specify user data when you launch an instance. For more information, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#) and [Running commands on your Linux instance at launch \(p. 653\)](#).

Modify instance user data

You can modify user data for an instance in the stopped state if the root volume is an EBS volume. For more information, see [View and update the instance user data \(p. 655\)](#).

Retrieve instance user data

To retrieve user data from within a running instance, use the following URI.

```
http://169.254.169.254/latest/user-data
```

A request for user data returns the data as it is (content type `application/octet-stream`).

This example returns user data that was provided as comma-separated text.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

This example returns user data that was provided as a script.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/user-data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/user-data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

To retrieve user data for an instance from your own computer, see [User data and the AWS CLI \(p. 657\)](#)

Retrieving dynamic data

To retrieve dynamic data from within a running instance, use the following URI.

```
http://169.254.169.254/latest/dynamic/
```

This example shows how to retrieve the high-level instance identity categories.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/dynamic/instance-identity/
rsa2048
pkcs7
document
signature
dsa2048
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/
rsa2048
pkcs7
document
signature
dsa2048
```

For more information about dynamic data and examples of how to retrieve it, see [Instance identity documents \(p. 684\)](#).

Example: AMI launch index value

This example demonstrates how you can use both user data and instance metadata to configure your instances.

Alice wants to launch four instances of her favorite database AMI, with the first acting as the original instance and the remaining three acting as replicas. When she launches them, she wants to add user data about the replication strategy for each replica. She is aware that this data will be available to all four instances, so she needs to structure the user data in a way that allows each instance to recognize which parts are applicable to it. She can do this using the `ami-launch-index` instance metadata value, which will be unique for each instance.

Here is the user data that Alice has constructed.

```
replicate-every=1min | replicate-every=5min | replicate-every=10min
```

The `replicate-every=1min` data defines the first replica's configuration, `replicate-every=5min` defines the second replica's configuration, and so on. Alice decides to provide this data as an ASCII string with a pipe symbol (|) delimiting the data for the separate instances.

Alice launches four instances using the `run-instances` command, specifying the user data.

```
aws ec2 run-instances \
```

```
--image-id ami-0abcdef1234567890 \
--count 4 \
--instance-type t2.micro \
--user-data "replicate-every=1min | replicate-every=5min | replicate-every=10min"
```

After they're launched, all instances have a copy of the user data and the common metadata shown here:

- AMI ID: ami-0abcdef1234567890
- Reservation ID: r-1234567890abcabc0
- Public keys: none
- Security group name: default
- Instance type: t2.micro

However, each instance has certain unique metadata.

Instance 1

Metadata	Value
instance-id	i-1234567890abcdef0
ami-launch-index	0
public-hostname	ec2-203-0-113-25.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-hostname	ip-10-251-50-12.ec2.internal
local-ipv4	10.251.50.35

Instance 2

Metadata	Value
instance-id	i-0598c7d356eba48d7
ami-launch-index	1
public-hostname	ec2-67-202-51-224.compute-1.amazonaws.com
public-ipv4	67.202.51.224
local-hostname	ip-10-251-50-36.ec2.internal
local-ipv4	10.251.50.36

Instance 3

Metadata	Value
instance-id	i-0ee992212549ce0e7
ami-launch-index	2
public-hostname	ec2-67-202-51-225.compute-1.amazonaws.com

Metadata	Value
public-ipv4	67.202.51.225
local-hostname	ip-10-251-50-37.ec2.internal
local-ipv4	10.251.50.37

Instance 4

Metadata	Value
instance-id	i-1234567890abcdef0
ami-launch-index	3
public-hostname	ec2-67-202-51-226.compute-1.amazonaws.com
public-ipv4	67.202.51.226
local-hostname	ip-10-251-50-38.ec2.internal
local-ipv4	10.251.50.38

Alice can use the `ami-launch-index` value to determine which portion of the user data is applicable to a particular instance.

- She connects to one of the instances, and retrieves the `ami-launch-index` for that instance to ensure it is one of the replicas:

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/meta-data/api/token"
-H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/ami-launch-index
2
```

For the following steps, the IMDSv2 requests use the stored token from the preceding IMDSv2 command, assuming the token has not expired.

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-launch-index
2
```

- She saves the `ami-launch-index` as a variable.

IMDSv2

```
[ec2-user ~]$ ami_launch_index=`curl -H "X-aws-ec2-metadata-token: $TOKEN" -v
http://169.254.169.254/latest/meta-data/ami-launch-index`
```

IMDSv1

```
[ec2-user ~]$ ami_launch_index=`curl http://169.254.169.254/latest/meta-data/ami-
launch-index`
```

- She saves the user data as a variable.

IMDSv2

```
[ec2-user ~]$ user_data=`curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/user-data`
```

IMDSv1

```
[ec2-user ~]$ user_data=`curl http://169.254.169.254/latest/user-data`
```

- Finally, Alice uses the `cut` command to extract the portion of the user data that is applicable to that instance.

IMDSv2

```
[ec2-user ~]$ echo $user_data | cut -d"|" -f"$ami_launch_index"replicate-every=5min
```

IMDSv1

```
[ec2-user ~]$ echo $user_data | cut -d"|" -f"$ami_launch_index"replicate-every=5min
```

Instance metadata categories

The following table lists the categories of instance metadata.

Note

When Amazon EC2 releases a new instance metadata category, the instance metadata for the new category might not be available for existing instances. To ensure that the instance metadata is available for an existing instance, you need to [stop and then start \(p. 587\)](#) the instance.

Important

Some of the category names in the following table are placeholders for data that is unique to your instance. For example, `mac` represents the MAC address for the network interface. You must replace the placeholders with the actual values.

Data	Description	Release date
ami-id	The AMI ID used to launch the instance.	Version 1.0
ami-launch-index	If you started more than one instance at the same time, this value indicates the order in which the instance was launched. The value of the first instance launched is 0.	Version 1.0
ami-manifest-path	The path to the AMI manifest file in Amazon S3. If you used an Amazon EBS-backed AMI to launch the instance, the returned result is unknown.	Version 1.0
ancestor-ami-ids	The AMI IDs of any instances that were rebundled to create this AMI. This value will only exist if the	2007-10-10

Data	Description	Release date
	AMI manifest file contained an <code>ancestor-amis</code> key.	
<code>block-device-mapping/ami</code>	The virtual device that contains the root/boot file system.	2007-12-15
<code>block-device-mapping/ebs N</code>	The virtual devices associated with any Amazon EBS volumes. Amazon EBS volumes are only available in metadata if they were present at launch time or when the instance was last started. The <i>N</i> indicates the index of the Amazon EBS volume (such as <code>ebs1</code> or <code>ebs2</code>).	2007-12-15
<code>block-device-mapping/ephemeral N</code>	The virtual devices for any non-NVMe instance store volumes. The <i>N</i> indicates the index of each volume. The number of instance store volumes in the block device mapping might not match the actual number of instance store volumes for the instance. The instance type determines the number of instance store volumes that are available to an instance. If the number of instance store volumes in a block device mapping exceeds the number available to an instance, the additional instance store volumes are ignored.	2007-12-15
<code>block-device-mapping/root</code>	The virtual devices or partitions associated with the root devices or partitions on the virtual device, where the root (/ or C:) file system is associated with the given instance.	2007-12-15
<code>block-device-mapping/swap</code>	The virtual devices associated with swap. Not always present.	2007-12-15
<code>elastic-gpus/associations/<i>elastic-gpu-id</i></code>	If there is an Elastic GPU attached to the instance, contains a JSON string with information about the Elastic GPU, including its ID and connection information.	2016-11-30
<code>elastic-inference/associations/<i>ei-a-id</i></code>	If there is an Elastic Inference accelerator attached to the instance, contains a JSON string with information about the Elastic Inference accelerator, including its ID and type.	2018-11-29

Data	Description	Release date
events/maintenance/history	If there are completed or canceled maintenance events for the instance, contains a JSON string with information about the events. For more information, see To view event history about completed or canceled events (p. 708) .	2018-08-17
events/maintenance/scheduled	If there are active maintenance events for the instance, contains a JSON string with information about the events. For more information, see Viewing scheduled events (p. 704) .	2018-08-17
hostname	The private IPv4 DNS hostname of the instance. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0).	Version 1.0
iam/info	If there is an IAM role associated with the instance, contains information about the last time the instance profile was updated, including the instance's LastUpdated date, InstanceProfileArn, and InstanceProfileId. Otherwise, not present.	2012-01-12
iam/security-credentials/ <i>role-name</i>	If there is an IAM role associated with the instance, <i>role-name</i> is the name of the role, and <i>role-name</i> contains the temporary security credentials associated with the role (for more information, see Retrieving security credentials from instance metadata (p. 975)). Otherwise, not present.	2012-01-12
identity-credentials/ec2/info	[Internal use only] Information about the credentials in identity-credentials/ec2/security-credentials/ec2-instance. These credentials are used by AWS features such as EC2 Instance Connect, and do not have any additional AWS API permissions or privileges beyond identifying the instance.	2018-05-23

Data	Description	Release date
<code>identity-credentials/ec2/security-credentials/ec2-instance</code>	[Internal use only] Credentials that allow on-instance software to identify itself to AWS to support features such as EC2 Instance Connect. These credentials do not have any additional AWS API permissions or privileges.	2018-05-23
<code>instance-action</code>	Notifies the instance that it should reboot in preparation for bundling. Valid values: <code>none</code> <code>shutdown</code> <code>bundle-pending</code> .	2008-09-01
<code>instance-id</code>	The ID of this instance.	Version 1.0
<code>instance-life-cycle</code>	The purchasing option of this instance. For more information, see Instance purchasing options (p. 303) .	2019-10-01
<code>instance-type</code>	The type of instance. For more information, see Instance types (p. 200) .	2007-08-29
<code>kernel-id</code>	The ID of the kernel launched with this instance, if applicable.	2008-02-01
<code>local-hostname</code>	The private IPv4 DNS hostname of the instance. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0).	2007-01-19
<code>local-ipv4</code>	The private IPv4 address of the instance. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0).	Version 1.0
<code>mac</code>	The instance's media access control (MAC) address. In cases where multiple network interfaces are present, this refers to the eth0 device (the device for which the device number is 0).	2011-01-01
<code>metrics/vhostmd</code>	No longer available.	2011-05-01
<code>network/interfaces/macs/mac/device-number</code>	The unique device number associated with that interface. The device number corresponds to the device name; for example, a <code>device-number</code> of 2 is for the eth2 device. This category corresponds to the <code>DeviceIndex</code> and <code>device-index</code> fields that are used by the Amazon EC2 API and the EC2 commands for the AWS CLI.	2011-01-01

Data	Description	Release date
<code>network/interfaces/macs/mac/interface-id</code>	The ID of the network interface.	2011-01-01
<code>network/interfaces/macs/mac/ipv4-associations/public-ip</code>	The private IPv4 addresses that are associated with each public IP address and assigned to that interface.	2011-01-01
<code>network/interfaces/macs/mac/ipv6s</code>	The IPv6 addresses associated with the interface. Returned only for instances launched into a VPC.	2016-06-30
<code>network/interfaces/macs/mac/local-hostname</code>	The interface's local hostname.	2011-01-01
<code>network/interfaces/macs/mac/local-ipv4s</code>	The private IPv4 addresses associated with the interface.	2011-01-01
<code>network/interfaces/macs/mac/mac</code>	The instance's MAC address.	2011-01-01
<code>network/interfaces/macs/mac/owner-id</code>	The ID of the owner of the network interface. In multiple-interface environments, an interface can be attached by a third party, such as Elastic Load Balancing. Traffic on an interface is always billed to the interface owner.	2011-01-01
<code>network/interfaces/macs/mac/public-hostname</code>	The interface's public DNS (IPv4). This category is only returned if the <code>enableDnsHostnames</code> attribute is set to true. For more information, see Using DNS with Your VPC .	2011-01-01
<code>network/interfaces/macs/mac/public-ipv4s</code>	The public IP address or Elastic IP addresses associated with the interface. There may be multiple IPv4 addresses on an instance.	2011-01-01
<code>network/interfaces/macs/mac/security-groups</code>	Security groups to which the network interface belongs.	2011-01-01
<code>network/interfaces/macs/mac/security-group-ids</code>	The IDs of the security groups to which the network interface belongs.	2011-01-01
<code>network/interfaces/macs/mac/subnet-id</code>	The ID of the subnet in which the interface resides.	2011-01-01
<code>network/interfaces/macs/mac/subnet-ipv4-cidr-block</code>	The IPv4 CIDR block of the subnet in which the interface resides.	2011-01-01
<code>network/interfaces/macs/mac/subnet-ipv6-cidr-blocks</code>	The IPv6 CIDR block of the subnet in which the interface resides.	2016-06-30
<code>network/interfaces/macs/mac/vpc-id</code>	The ID of the VPC in which the interface resides.	2011-01-01

Data	Description	Release date
network/interfaces/macs/mac/vpc-ipv4-cidr-block	The primary IPv4 CIDR block of the VPC.	2011-01-01
network/interfaces/macs/mac/vpc-ipv4-cidr-blocks	The IPv4 CIDR blocks for the VPC.	2016-06-30
network/interfaces/macs/mac/vpc-ipv6-cidr-blocks	The IPv6 CIDR block of the VPC in which the interface resides.	2016-06-30
placement/availability-zone	The Availability Zone in which the instance launched.	2008-02-01
placement/availability-zone-id	The static Availability Zone ID in which the instance is launched. The Availability Zone ID is consistent across accounts. However, it might be different from the Availability Zone, which can vary by account.	2020-08-24
placement/group-name	The name of the placement group in which the instance is launched.	2020-08-24
placement/host-id	The ID of the host on which the instance is launched. Applicable only to Dedicated Hosts.	2020-08-24
placement/partition-number	The number of the partition in which the instance is launched.	2020-08-24
placement/region	The AWS Region in which the instance is launched.	2020-08-24
product-codes	AWS Marketplace product codes associated with the instance, if any.	2007-03-01
public-hostname	The instance's public DNS. This category is only returned if the enableDnsHostnames attribute is set to true. For more information, see Using DNS with Your VPC in the <i>Amazon VPC User Guide</i> .	2007-01-19
public-ipv4	The public IPv4 address. If an Elastic IP address is associated with the instance, the value returned is the Elastic IP address.	2007-01-19
public-keys/0/openssh-key	Public key. Only available if supplied at instance launch time.	Version 1.0
ramdisk-id	The ID of the RAM disk specified at launch time, if applicable.	2007-10-10
reservation-id	The ID of the reservation.	Version 1.0

Data	Description	Release date
security-groups	The names of the security groups applied to the instance. After launch, you can change the security groups of the instances. Such changes are reflected here and in network/interfaces/macs/ <i>mac</i> /security-groups.	Version 1.0
services/domain	The domain for AWS resources for the Region.	2014-02-25
services/partition	The partition that the resource is in. For standard AWS Regions, the partition is aws. If you have resources in other partitions, the partition is aws- <i>partitionname</i> . For example, the partition for resources in the China (Beijing) Region is aws-cn.	2015-10-20
spot/instance-action	The action (hibernate, stop, or terminate) and the approximate time, in UTC, when the action will occur. This item is present only if the Spot Instance has been marked for hibernate, stop, or terminate. For more information, see instance-action (p. 431) .	2016-11-15
spot/termination-time	The approximate time, in UTC, that the operating system for your Spot Instance will receive the shutdown signal. This item is present and contains a time value (for example, 2015-01-05T18:02:00Z) only if the Spot Instance has been marked for termination by Amazon EC2. The termination-time item is not set to a time if you terminated the Spot Instance yourself. For more information, see termination-time (p. 432) .	2014-11-05

Dynamic data categories

The following table lists the categories of dynamic data.

Data	Description	Release date
fws/instance-monitoring	Value showing whether the customer has enabled detailed one-minute monitoring in CloudWatch. Valid values: enabled disabled	2009-04-04

Data	Description	Release date
instance-identity/document	JSON containing instance attributes, such as instance-id, private IP address, etc. See Instance identity documents (p. 684) .	2009-04-04
instance-identity/pkcs7	Used to verify the document's authenticity and content against the signature. See Instance identity documents (p. 684) .	2009-04-04
instance-identity/signature	Data that can be used by other parties to verify its origin and authenticity. See Instance identity documents (p. 684) .	2009-04-04

Instance identity documents

Each instance that you launch has an instance identity document that provides information about the instance itself. You can use the instance identity document to validate the attributes of the instance.

The instance identity document is generated when the instance is launched and it is exposed (in plaintext JSON format) through the Instance Metadata Service. The IP address 169.254.169.254 is a link-local address and is valid only from the instance. For more information, see [Link-local address](#) on Wikipedia.

You can retrieve the instance identity document from a running instance at any time. The instance identity document includes the following information:

Data	Description
devpayProductCodes	Deprecated.
marketplaceProductCode	The AWS Marketplace product code of the AMI used to launch the instance.
availabilityZone	The Availability Zone in which the instance is running.
privateIp	The private IPv4 address of the instance.
version	The version of the instance identity document format.
instanceId	The ID of the instance.
billingProducts	The billing product code of the AMI used to launch the instance.
instanceType	The instance type of the instance.
accountId	The ID of the AWS account that launched the instance.
imageId	The ID of the AMI used to launch the instance.
pendingTime	The date and time that the instance was launched.
architecture	The architecture of the AMI used to launch the instance (i386 x86_64 arm64).
kernelId	The ID of the kernel associated with the instance, if applicable.
ramdiskId	The ID of the RAM disk associated with the instance, if applicable.
region	The Region in which the instance is running.

Retrieve the plaintext instance identity document

To retrieve the plaintext instance identity document

Connect to the instance and run one of the following commands depending on the Instance Metadata Service (IMDS) version used by the instance.

IMDSv2

```
$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`\n&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/dynamic/instance-identity/document
```

IMDSv1

```
$ curl http://169.254.169.254/latest/dynamic/instance-identity/document
```

The following is example output.

```
{\n    "devpayProductCodes" : null,\n    "marketplaceProductCodes" : [ "1abc2defghijklm3nopqrs4tu" ],\n    "availabilityZone" : "us-west-2b",\n    "privateIp" : "10.158.112.84",\n    "version" : "2017-09-30",\n    "instanceId" : "i-1234567890abcdef0",\n    "billingProducts" : null,\n    "instanceType" : "t2.micro",\n    "accountId" : "123456789012",\n    "imageId" : "ami-5fb8c835",\n    "pendingTime" : "2016-11-19T16:32:11Z",\n    "architecture" : "x86_64",\n    "kernelId" : null,\n    "ramdiskId" : null,\n    "region" : "us-west-2"\n}
```

Verifying the instance identity document

If you intend to use the contents of the instance identity document for an important purpose, you should verify its contents and authenticity before using it.

The plaintext instance identity document is accompanied by three hashed and encrypted signatures. You can use these signatures to verify the origin and authenticity of the instance identity document and the information that it includes. The following signatures are provided:

- Base64-encoded signature—This is a base64-encoded SHA256 hash of the instance identity document that is encrypted using an RSA key pair.
- PKCS7 signature—This is a SHA1 hash of the instance identity document that is encrypted using a DSA key pair.
- RSA-2048 signature—This is a SHA256 hash of the instance identity document that is encrypted using an RSA-2048 key pair.

Each signature is available at a different endpoint in the instance metadata. You can use any one of these signatures depending on your hashing and encryption requirements. To verify the signatures, you must use the corresponding AWS public certificate.

Important

To validate the instance identity document using the base64-encoded signature or RSA2048 signature, you must request the corresponding AWS public certificate from [AWS Support](#).

The following topics provide detailed steps for validating the instance identity document using each signature.

- [Using the PKCS7 signature to verify the instance identity document \(p. 686\)](#)
- [Using the base64-encoded signature to verify the instance identity document \(p. 689\)](#)
- [Using the RSA-2048 signature to verify the instance identity document \(p. 690\)](#)

Using the PKCS7 signature to verify the instance identity document

This topic explains how to verify the instance identity document using the PKCS7 signature and the AWS DSA public certificate.

To verify the instance identity document using the PKCS7 signature and the AWS DSA public certificate

1. Connect to the instance.
2. Retrieve the PKCS7 signature from the instance metadata and add it to a file named `pkcs7`.
 - a. Add the `-----BEGIN PKCS7-----` header to the `pkcs7` file.

```
$ echo "-----BEGIN PKCS7-----" > pkcs7
```

- b. Retrieve the PKCS7 signature from the instance metadata and append it to the `pkcs7` file. Use one of the following commands depending on the IMDS version used by the instance.

IMDSv2

```
$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/dynamic/instance-identity/pkcs7 >> pkcs7
```

IMDSv1

```
$ curl -s http://169.254.169.254/latest/dynamic/instance-identity/pkcs7
>> pkcs7
```

- c. Append the `-----END PKCS7-----` footer to a new line in the `pkcs7` file.

```
$ echo "" >> pkcs7
$ echo "-----END PKCS7-----" >> pkcs7
```

3. Add the contents of the instance identity document from the instance metadata to a file named `document`. Use one of the following commands depending on the IMDS version used by the instance.

IMDSv2

```
$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/dynamic/instance-identity/document > document
```

IMDSv1

```
$ curl -s http://169.254.169.254/latest/dynamic/instance-identity/document  
> document
```

4. Add the AWS DSA public certificate to a file named `certificate`.

- a. Create the `certificate` file.

```
$ touch certificate
```

- b. Open the `certificate` file using your preferred text editor and add the contents of the AWS DSA public certificate. Choose the correct certificate for the AWS Region that your instance is in.

Important

If the AWS DSA public certificate for your Region is not listed below, contact [AWS Support](#).

Other AWS Regions

The following AWS public certificate is for all AWS Regions, except Hong Kong, Bahrain, China, and GovCloud.

```
-----BEGIN CERTIFICATE-----  
MIIC7TCCAq0CCQCWukjZ5V4azzAJBgcqhkjOOAODMFwxCzAJBgNVBAYTA1VTMRkw  
FwYDVQQIExBXYYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD  
VQOKExdBbWF6b24gV2ViIFNlcnPzY2VzIEzM0zAeFw0xMjAxMDUxMjU2MTJaFw0z  
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIExBXYYXNoaW5ndG9u  
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1  
cnZpY2VzIEzM0zCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e  
ih5006kK/n1Lz1lr7D8ZwtQP8fOEpp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3  
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P  
hviYt5JH/nY14hh3Pa1HJdskgQ1VALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j  
k+tKqMVHuAFcvAGKocTgsjJem6/5qomzJukDmbJNu9Qxw3rAotXau8Qe+MBCJ1/U  
hhy1KHVpcG19fueQ2s61L0CaO/buyU1CiYQk40KNHCCHfNiZbd1x1E9rpUp7bnF  
1Ra2v1ntMX3caRVDbtPEWmdxSCySYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf  
MNmP9CM5eovQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx411HW  
MXrs3Ig1b6+hUIB+S8dz8/mm0Obpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw  
vSeDCOUMYQR7R9LINYwouHiziqYQMAkGBqyGSM44BAMDLwAwLAIUWXBlk40xtTwSw  
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K  
-----END CERTIFICATE-----
```

Hong Kong Region

The AWS public certificate for the Hong Kong Region is as follows.

```
-----BEGIN CERTIFICATE-----  
MIIC7zCCAq4CCQC07MjE5Y3VLjAJBgcqhkjOOAODMFwxCzAJBgNVBAYTA1VTMRkw  
FwYDVQQIExBXYYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD  
VQOKExdBbWF6b24gV2ViIFNlcnPzY2VzIEzM0zAeFw0xOTAyMDMwMjIxMjFaFw00  
NTAyMDMwMjIxMjFaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIExBXYYXNoaW5ndG9u  
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1  
cnZpY2VzIEzM0zCCAbgwggEsBgcqhkjOOAQBMIIBHwKBgQDvQ9RzVvf4MAwGbqfX  
b1CvCoVb9957OkLGn/04CowHXJ+vTBR7eyIa6AoXltsQXB0mrJswToFKKxT4gbuw  
jK7s9QQX4CmTRWcEgO2RxtZSVjOhsUQMh+yf7Ht4OVL97LWnNfGsX2cwjcRWHYgI  
71vnubNBzLQHdSEwMNq0Bk76PwIVAMan6XIEEPnwr4e6u/RNnWBGKd9FAoGBAOOG  
eSNmxpW4QFu4pIlAykne6nTzKKHT87gdXkAkfoC5fafOxxhnE2HezZhP9Ap2tMV5  
8bWNvoPHvoKCQqwfM+OUB1AxC/3vqoVkKL2mG1KgUH9+httpMTkw03RREnKe7I50  
x9qDimJpOihrl4I0dYvy9xUOoz+DzFAW8+y1WVYpa4GFAAKBqQDbnBAKSxWr9QHY  
6Dt+EFdgz61AZLedeBKpaP53Z1DTO34J0C55YbJTwBTFGqPtOLxnUVD1GiD6GbmC  
80f3jvogPR1mSmGsydbNbZnbUEVWrRhe+y5zJ3g9qs/DwmDW0deEFvkhwVnLJkFJ
```

Amazon Elastic Compute Cloud
User Guide for Linux Instances
Instance metadata and user data

```
9pd0u/ibRPH1lE2nz6pK7Gb0QtLyHTAJBgcqhkjOOAQDAzAAMC0CFQCoJlwGtJQC
cLoM4p/jtVFOj26xbgIUUS4pDKyHaG/eaygLttFpFJqzWHc=
-----END CERTIFICATE-----
```

Bahrain Region

The AWS public certificate for the Bahrain Region is as follows.

```
-----BEGIN CERTIFICATE-----
MIIC7jCCAq4CCQCWVigSmP8RhTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnPzY2VzIExEQzAeFw0xOTAyMDUxMzA2MjFaFw00
NTAyMDUxMzA2MjFaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1
cnZpY2VzIExEQzAeFw0xOTAyMDUxMzA2MjFaFw00
8n6cLZ38VE7ZmrjZ90QV//Gst6S1h7euhC23YppKXi1zovefSDwFU54zi3/oJ++q
PH1P1WL8IZ34BUGRTG4TVolvpoSmjkMvyRu5h1dKtzjV93Cxx15gVgyk+o1IEG
fZ2Kbw/Dd8JfoPS7KaScmJkxQ1VA1Zb1adFRGa2qcMkW2HWASyND17bAoGBANtz
IdhfMq+12I5iofY2o13H121Kj3LtZrWEg3W+/4rVhL31Tm0Nne1r19yGujrjQwy5
Zp9V4A/w9w2010Lx4K6hj34Ef/y/aqnZwNdNhv/FQP7AzOfju+Y16L13OOHQzrLoZ
Q+9cF7zeosekEnBQx3v6psNknKgD3Shgx+GO/LpCA4GFAAKBqOCVs7m77nuNALz8
wvUqcooxXMPkxJFl54NxAsAu19KP9K9N4svm003Zrb7t2F0tXRM8zu3TqMpryq1o5
mpMPsZDg6RXo9BF7Hn0DoZ6PJTa6md+NyTJWJKvXC7iJ8fGDBJqTciUHuCKr
12AztQ8bFWsrTgTzPE3p6U5ckcgV1TAJBgcqhkjOOAQDAy8AMCwCFB2NZGwm5ED1
86ayV3c1PEDukgQIAhQow38rQkN/VwHVewS9DqEshXHjuQ==
-----END CERTIFICATE-----
```

GovCloud Regions

The AWS public certificate for the AWS GovCloud Regions is as follows.

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQQKExdBbWF6b24gV2ViIFNlcnPzY2VzIExEQzAeFw0x0MjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIExBXYXNoaW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFN1
cnZpY2VzIExEQzAeFw0x0MjAxMDUxMjU2MTJaFw0z
ih50O6kK/n1Lz1lr7d8ZwtQP8fOEpp5E2ng+D6UD1Z1gYipr58Kj3nssSNpI6bx3
VyIQzK7wLclnd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQ1VALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tKqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBCJ1/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buyu1CiYQk40KNHCHfNizbd1x1E9rpUp7bnF
lRa2v1ntMX3caRVDbt+PEWmdxSCy5FDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eovQOGx5ho8WqD+aTeb+s+k2tn92BBPqeZqpWra5P/+jrdKml1qx411HW
MXrs3IgIb6+hUIB+S8dz8/mm0Obpr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMQR7R9LINYwouHiziqQYMAkGBqyGSM44BAMDLwAwLAIUWXBlk40xtTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

China Regions

The AWS public certificate for the China (Beijing) and China (Ningxia) Regions is as follows.

```
-----BEGIN CERTIFICATE-----
MIIDNjCCAh4CCQD3yz1w1AVkTzANBgkqhkiG9w0BAQsFADBCMQswCQYDVQQGEwJV
UzEZMBcGA1UECBM0V2FzaGluZ3Rvb1BTdGF0ZTEQMA4GA1UEBxMHU2VhdHRsZTEg
MB4GA1UEChMXQW1hem9uIFd1YiBTZXJ2aN1cyBMTEMwIBcNMTUwNTEzMDk1OTE1
WhgPMjE5NDEwMTYwOTU5MTVaMFwxCzAJBgnVBAYTA1VTMRkwFwYDVQQIExBXYXNo
aW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24g
V2ViIFNlcnPzY2VzIExEQzCASAiwDQYJKoZIhvCNQEBBQADggEPADCCAQoCggEB
AMWk9vyppSmDU3AxZ2Cy2bvKeK3F1UqNpMuyeriizi+NTsZ8tQqtNloaQcqhto/l
-----END CERTIFICATE-----
```

```
gsw9+QSnEJyWnmivJWOBdn9CyDpN7cpHVmeGgNJI2fvImWyWe2f2Kq/BL9l7N7C
P2ZT52/sH9orlck1n2z08xPi7MItgPHQwu3OxsGQsAdWucdxjHGtdchulp0iuJ31
jsTAPKZ3p1/sxPXBXAgBMatPHhRBqhwHO/Twm4J3GmTLWN7oVDds4W3bPKQfnw3r
vtBj/SM4/IgQ3xJs1Fc190TzbQbgxi88R/gWTbs7GsyT2PzstU30yLdJhKfdZKz
/aIzraHvoDTWFaOdy0+OOaECAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAdSzN2+0E
V1BfR3DPWJHWRf1b7z1+1X/ZseW2hYE5r6YxrLv+1VPf/L5I6kB7GETqhZUqteY7
zAceoLrVu/7OynRyfQetJVGichaaxLNM3lcr6kcxOowb+WQQ84cwrB3keykH4gRX
KHB2rlWSxta+2panSE01JX2q5jhcfP90rD0tZj1pYv57N/Z9i0+dvQPJnChdq3BK
5pZlnIDnVVxqRike7BFy8tKyPj7HzoPEF5mh9Kfnn1YoSVu+611MVv/qRjnyKfs9
c96nE98sYFj0ZVBzxW8Sq4Gh8FivmFhbQp1peGC19idOUqxPxWsasWxQX00azysP
9RyWLHKxH1dMuA==
-----END CERTIFICATE-----
```

- c. Save and close the file.
5. Use the **OpenSSL smime** command to verify the signature. Include the **-verify** option to indicate that the signature needs to be verified, and the **-noverify** option to indicate that the certificate does not need to be verified.

```
$ openssl smime -verify -in pkcs7 -inform PEM -content document -certfile certificate -
noverify
```

If the signature is valid, the `Verification successful` message appears. If the signature cannot be verified, contact AWS Support.

Using the base64-encoded signature to verify the instance identity document

This topic explains how to verify the instance identity document using the base64-encoded signature and the AWS RSA public certificate.

Important

To validate the instance identity document using the base64-encoded signature, you must request the AWS RSA public certificate from [AWS Support](#).

To validate the instance identity document using the base64-encoded signature and the AWS RSA public certificate

1. Connect to the instance.
2. Retrieve the base64-encoded signature from the instance metadata, convert it to binary, and add it to a file named `signature`. Use one of the following commands depending on the IMDS version used by the instance.

IMDSv2

```
$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600" ` \ && curl -H "X-aws-ec2-metadata-token: $TOKEN"
-v http://169.254.169.254/latest/dynamic/instance-identity/signature | base64 -d
> signature
```

IMDSv1

```
$ curl -s http://169.254.169.254/latest/dynamic/instance-identity/signature |
base64 -d > signature
```

3. Retrieve the plaintext instance identity document from the instance metadata and add it to a file named `document`. Use one of the following commands depending on the IMDS version used by the instance.

IMDSv2

```
$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \ && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/dynamic/instance-identity/document > document
```

IMDSv1

```
$ curl -s http://169.254.169.254/latest/dynamic/instance-identity/document > document
```

4. Add the AWS RSA public certificate that you received from AWS Support to a file named *certificate*.
5. Extract the public key from the certificate that you received from AWS Support and save it to a file named *key*.

```
$ openssl x509 -pubkey -noout -in certificate > key
```

6. Use **OpenSSL dgst** command to verify the instance identity document.

```
$ openssl dgst -sha256 -verify key -signature signature document
```

If the signature is valid, the `Verified OK` message appears. If the signature cannot be verified, contact AWS Support.

Using the RSA-2048 signature to verify the instance identity document

This topic explains how to verify the instance identity document using the RSA-2048 signature and the AWS RSA-2048 public certificate.

Important

To validate the instance identity document using the RSA-2048 signature, you must request the AWS RSA-2048 public certificate from [AWS Support](#).

To verify the instance identity document using the RSA-2048 signature and the AWS RSA-2048 public certificate

1. Connect to the instance.
2. Retrieve the RSA-2048 signature from the instance metadata and add it to a file named *rsa2048*.
 - a. Add the `-----BEGIN PKCS7-----` header to the *rsa2048* file.

```
$ echo "-----BEGIN PKCS7-----" > rsa2048
```

- b. Retrieve the RSA-2048 signature from the instance metadata and append it to the *rsa2048* file. Use one of the following commands depending on the IMDS version used by the instance.

IMDSv2

```
$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \ && curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/dynamic/instance-identity/rsa2048 >> rsa2048
```

IMDSv1

```
$ curl -s http://169.254.169.254/latest/dynamic/instance-identity/rsa2048  
>> rsa2048
```

- c. Append the -----END PKCS7----- footer to a new line in the `rsa2048` file.

```
$ echo "" >> rsa2048  
$ echo "-----END PKCS7-----" >> rsa2048
```

3. Add the contents of the instance identity document from the instance metadata to a file named `document`. Use one of the following commands depending on the IMDS version used by the instance.

IMDSv2

```
$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \  
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/ dynamic/instance-identity/document > document
```

IMDSv1

```
$ curl -s http://169.254.169.254/latest/dynamic/instance-identity/document  
> document
```

4. Add the AWS RSA-2048 public certificate to a file named `certificate`.
 - a. Create the `certificate` file.
 - b. Open the `certificate` file using your preferred text editor and add the contents of the AWS RSA-2048 public certificate that you received from AWS Support.
 - c. Save and close the file.
5. Use the **OpenSSL smime** command to verify the signature. Include the `-verify` option to indicate that the signature needs to be verified, and the `-noverify` option to indicate that the certificate does not need to be verified.

```
$ openssl smime -verify -in rsa2048 -inform PEM -content document -certfile certificate  
-noverify
```

If the signature is valid, the `Verification successful` message appears. If the signature cannot be verified, contact AWS Support.

Amazon Elastic Inference

Amazon Elastic Inference (EI) is a resource you can attach to your Amazon EC2 CPU instances to accelerate your deep learning (DL) inference workloads. Amazon EI accelerators come in multiple sizes and are a cost-effective method to build intelligent capabilities into applications running on Amazon EC2 instances.

Amazon EI distributes model operations defined by TensorFlow, Apache MXNet, and the Open Neural Network Exchange (ONNX) format through MXNet between low-cost, DL inference accelerators and the CPU of the instance.

For more information about Amazon Elastic Inference, see the [Amazon EI Developer Guide](#).

Identify EC2 Linux instances

Your application might need to determine whether it is running on an EC2 instance.

For information about identifying Windows instances, see [Identify EC2 Windows Instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Inspecting the instance identity document

For a definitive and cryptographically verified method of identifying an EC2 instance, check the instance identity document, including its signature. These documents are available on every EC2 instance at the local, non-routable address `http://169.254.169.254/latest/dynamic/instance-identity/`. For more information, see [Instance identity documents \(p. 684\)](#).

Inspecting the system UUID

You can get the system UUID and look for the presence of the characters "ec2" or "EC2" in the beginning octet of the UUID. This method to determine whether a system is an EC2 instance is quick but potentially inaccurate because there is a small chance that a system that is not an EC2 instance could have a UUID that starts with these characters. Furthermore, for EC2 instances that are not using Amazon Linux, the distribution's implementation of SMBIOS might represent the UUID in little-endian format, therefore the "EC2" characters do not appear at the beginning of the UUID.

Example : Get the UUID from the hypervisor

If `/sys/hypervisor/uuid` exists, you can use the following command:

```
[ec2-user ~]$ cat /sys/hypervisor/uuid
```

In the following example output, the UUID starts with "ec2", which indicates that the system is probably an EC2 instance.

```
ec2e1916-9099-7caf-fd21-012345abcdef
```

Example : Get the UUID from DMI (HVM instances only)

On HVM instances only, you can use the Desktop Management Interface (DMI).

You can use the `dmidecode` tool to return the UUID. On Amazon Linux, use the following command to install the `dmidecode` tool if it's not already installed on your instance:

```
[ec2-user ~]$ sudo yum install dmidecode -y
```

Then run the following command:

```
[ec2-user ~]$ sudo dmidecode --string system-uuid
```

Alternatively, use the following command:

```
[ec2-user ~]$ sudo cat /sys/devices/virtual/dmi/id/product_uuid
```

In the following example output, the UUID starts with "EC2", which indicates that the system is probably an EC2 instance.

```
EC2E1916-9099-7CAF-FD21-01234ABCDEF
```

In the following example output, the UUID is represented in little-endian format.

```
45E12AEC-DCD1-B213-94ED-01234ABCDEF
```

On Nitro instances, the following command can be used:

```
[ec2-user ~]$ cat /sys/devices/virtual/dmi/id/board_asset_tag
```

This returns the instance ID, which is unique to EC2 instances:

```
i-0af01c0123456789a
```

Monitoring Amazon EC2

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Elastic Compute Cloud (Amazon EC2) instances and your AWS solutions. You should collect monitoring data from all of the parts in your AWS solutions so that you can more easily debug a multi-point failure if one occurs. Before you start monitoring Amazon EC2, however, you should create a monitoring plan that should include:

- What are your goals for monitoring?
- What resources will you monitor?
- How often will you monitor these resources?
- What monitoring tools will you use?
- Who will perform the monitoring tasks?
- Who should be notified when something goes wrong?

After you have defined your monitoring goals and have created your monitoring plan, the next step is to establish a baseline for normal Amazon EC2 performance in your environment. You should measure Amazon EC2 performance at various times and under different load conditions. As you monitor Amazon EC2, you should store a history of monitoring data that you've collected. You can compare current Amazon EC2 performance to this historical data to help you to identify normal performance patterns and performance anomalies, and devise methods to address them. For example, you can monitor CPU utilization, disk I/O, and network utilization for your EC2 instances. When performance falls outside your established baseline, you might need to reconfigure or optimize the instance to reduce CPU utilization, improve disk I/O, or reduce network traffic.

To establish a baseline you should, at a minimum, monitor the following items:

Item to monitor	Amazon EC2 metric	Monitoring agent/CloudWatch Logs
CPU utilization	CPUUtilization (p. 717)	
Network utilization	NetworkIn (p. 717) NetworkOut (p. 717)	
Disk performance	DiskReadOps (p. 717) DiskWriteOps (p. 717)	
Disk Reads/Writes	DiskReadBytes (p. 717) DiskWriteBytes (p. 717)	
Memory utilization, disk swap utilization, disk space utilization, page file utilization, log collection		[Linux and Windows Server instances] Collect Metrics and Logs from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent [Migration from previous CloudWatch Logs agent on

Item to monitor	Amazon EC2 metric	Monitoring agent/CloudWatch Logs
		Windows Server instances] Migrate Windows Server Instance Log Collection to the CloudWatch Agent

Automated and manual monitoring

AWS provides various tools that you can use to monitor Amazon EC2. You can configure some of these tools to do the monitoring for you, while some of the tools require manual intervention.

Monitoring tools

- [Automated monitoring tools \(p. 695\)](#)
- [Manual monitoring tools \(p. 696\)](#)

Automated monitoring tools

You can use the following automated monitoring tools to watch Amazon EC2 and report back to you when something is wrong:

- **System status checks** – monitor the AWS systems required to use your instance to ensure that they are working properly. These checks detect problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue or you can resolve it yourself (for example, by stopping and restarting or terminating and replacing an instance). Examples of problems that cause system status checks to fail include:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host that impact network reachability

For more information, see [Status checks for your instances \(p. 697\)](#).

- **Instance status checks** – monitor the software and network configuration of your individual instance. These checks detect problems that require your involvement to repair. When an instance status check fails, typically you will need to address the problem yourself (for example, by rebooting the instance or by making modifications in your operating system). Examples of problems that may cause instance status checks to fail include:

- Failed system status checks
- Misconfigured networking or startup configuration
- Exhausted memory
- Corrupted file system
- Incompatible kernel

For more information, see [Status checks for your instances \(p. 697\)](#).

- **Amazon CloudWatch alarms** – watch a single metric over a time period you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon Simple Notification Service (Amazon SNS) topic or Amazon EC2 Auto Scaling policy. Alarms invoke actions for sustained state changes only. CloudWatch alarms will not invoke actions simply because they are in a particular state; the state

must have changed and been maintained for a specified number of periods. For more information, see [Monitoring your instances using CloudWatch \(p. 715\)](#).

- **Amazon CloudWatch Events** – automate your AWS services and respond automatically to system events. Events from AWS services are delivered to CloudWatch Events in near real time, and you can specify automated actions to take when an event matches a rule you write. For more information, see [What is Amazon CloudWatch Events?](#).
- **Amazon CloudWatch Logs** – monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, or other sources. For more information, see the [Amazon CloudWatch Logs User Guide](#).
- **CloudWatch agent** – collect logs and system-level metrics from both hosts and guests on your EC2 instances and on-premises servers. For more information, see [Collecting Metrics and Logs from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent](#) in the *Amazon CloudWatch User Guide*.
- **AWS Management Pack for Microsoft System Center Operations Manager** – links Amazon EC2 instances and the Windows or Linux operating systems running inside them. The AWS Management Pack is an extension to Microsoft System Center Operations Manager. It uses a designated computer in your datacenter (called a watcher node) and the Amazon Web Services APIs to remotely discover and collect information about your AWS resources. For more information, see [AWS Management Pack for Microsoft System Center](#).

Manual monitoring tools

Another important part of monitoring Amazon EC2 involves manually monitoring those items that the monitoring scripts, status checks, and CloudWatch alarms don't cover. The Amazon EC2 and CloudWatch console dashboards provide an at-a-glance view of the state of your Amazon EC2 environment.

- Amazon EC2 Dashboard shows:
 - Service Health and Scheduled Events by Region
 - Instance state
 - Status checks
 - Alarm status
 - Instance metric details (In the navigation pane choose **Instances**, select an instance, and choose the **Monitoring** tab)
 - Volume metric details (In the navigation pane choose **Volumes**, select a volume, and choose the **Monitoring** tab)
- Amazon CloudWatch Dashboard shows:
 - Current alarms and status
 - Graphs of alarms and resources
 - Service health status

In addition, you can use CloudWatch to do the following:

- Graph Amazon EC2 monitoring data to troubleshoot issues and discover trends
- Search and browse all your AWS resource metrics
- Create and edit alarms to be notified of problems
- See at-a-glance overviews of your alarms and AWS resources

Best practices for monitoring

Use the following best practices for monitoring to help you with your Amazon EC2 monitoring tasks.

- Make monitoring a priority to head off small problems before they become big ones.

- Create and implement a monitoring plan that collects monitoring data from all of the parts in your AWS solution so that you can more easily debug a multi-point failure if one occurs. Your monitoring plan should address, at a minimum, the following questions:
 - What are your goals for monitoring?
 - What resources you will monitor?
 - How often you will monitor these resources?
 - What monitoring tools will you use?
 - Who will perform the monitoring tasks?
 - Who should be notified when something goes wrong?
- Automate monitoring tasks as much as possible.
- Check the log files on your EC2 instances.

Monitoring the status of your instances

You can monitor the status of your instances by viewing status checks and scheduled events for your instances.

A status check gives you the information that results from automated checks performed by Amazon EC2. These automated checks detect whether specific issues are affecting your instances. The status check information, together with the data provided by Amazon CloudWatch, gives you detailed operational visibility into each of your instances.

You can also see status of specific events that are scheduled for your instances. The status of events provides information about upcoming activities that are planned for your instances, such as rebooting or retirement. They also provide the scheduled start and end time of each event.

Contents

- [Status checks for your instances \(p. 697\)](#)
- [Scheduled events for your instances \(p. 704\)](#)

Status checks for your instances

With instance status monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues. You can view the results of these status checks to identify specific and detectable problems. The event status data augments the information that Amazon EC2 already provides about the state of each instance (such as pending, running, stopping) and the utilization metrics that Amazon CloudWatch monitors (CPU utilization, network traffic, and disk activity).

Status checks are performed every minute, returning a pass or a fail status. If all checks pass, the overall status of the instance is **OK**. If one or more checks fail, the overall status is **impaired**. Status checks are built into Amazon EC2, so they cannot be disabled or deleted.

When a status check fails, the corresponding CloudWatch metric for status checks is incremented. For more information, see [Status check metrics \(p. 723\)](#). You can use these metrics to create CloudWatch alarms that are triggered based on the result of the status checks. For example, you can create an alarm to warn you if status checks fail on a specific instance. For more information, see [Creating and editing status check alarms \(p. 701\)](#).

You can also create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying issue. For more information, see [Recover your instance \(p. 612\)](#).

Contents

- [Types of status checks \(p. 698\)](#)
- [Viewing status checks \(p. 698\)](#)
- [Reporting instance status \(p. 701\)](#)
- [Creating and editing status check alarms \(p. 701\)](#)

Types of status checks

There are two types of status checks: system status checks and instance status checks.

System status checks

System status checks monitor the AWS systems on which your instance runs. These checks detect underlying problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue, or you can resolve it yourself. For instances backed by Amazon EBS, you can stop and start the instance yourself, which in most cases results in the instance being migrated to a new host. For instances backed by instance store, you can terminate and replace the instance.

The following are examples of problems that can cause system status checks to fail:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host that impact network reachability

Instance status checks

Instance status checks monitor the software and network configuration of your individual instance. Amazon EC2 checks the health of the instance by sending an address resolution protocol (ARP) request to the network interface (NIC). These checks detect problems that require your involvement to repair. When an instance status check fails, you typically must address the problem yourself (for example, by rebooting the instance or by making instance configuration changes).

The following are examples of problems that can cause instance status checks to fail:

- Failed system status checks
- Incorrect networking or startup configuration
- Exhausted memory
- Corrupted file system
- Incompatible kernel

Viewing status checks

Amazon EC2 provides you with several ways to view and work with status checks.

Viewing status using the console

You can view status checks using the AWS Management Console.

New console

To view status checks (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. On the **Instances** page, the **Status check** column lists the operational status of each instance.
4. To view the status of a specific instance, select the instance, and then choose the **Status Checks** tab.

The screenshot shows the AWS EC2 Instances page with the 'Status Checks' tab selected. The tab bar includes 'Details', 'Security', 'Networking', 'Storage', 'Status Checks' (which is highlighted in orange), 'Monitoring', and 'Tags'. Below the tabs, there's a section titled 'Status Checks' with a 'Info' link. A note states: 'Status checks detect problems that may impair i-0c0186a12aab3741d (t2largeFromRHEL73asmallAMI) from running your application.' Under 'System status checks', it says 'System reachability check passed' with a green checkmark icon. At the bottom, there's a 'Need assistance?' section with a note about opening a support case if an instance is unreachable for 20 minutes, a 'Open support case' button, and links to 'AWS Support Center' and 'Discussion Forums'.

If you have an instance with a failed status check and the instance has been unreachable for over 20 minutes, choose **Open support case** to submit a request for assistance. To troubleshoot system or instance status check failures yourself, see [Troubleshooting instances with failed status checks \(p. 1256\)](#).

5. To review the CloudWatch metrics for status checks, select the instance, and then choose the **Monitoring** tab. Scroll until you see the graphs for the following metrics:
 - **Status check failed (any)**
 - **Status check failed (instance)**
 - **Status check failed (system)**

Old console

To view status checks (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. On the **Instances** page, the **Status Checks** column lists the operational status of each instance.
4. To view the status of a specific instance, select the instance, and then choose the **Status Checks** tab.

The screenshot shows the AWS Lambda console with the 'Status Checks' tab selected. The 'System Status Checks' section indicates that system reachability checks have passed. The 'Instance Status Checks' section shows a failed check for 'Instance reachability check' at October 20, 2018, with a link to learn more about the issue.

Status checks detect problems that may impair this instance from running your applications. [Learn more](#) about status checks.

[Create Status Check Alarm](#)

System Status Checks ⓘ

These checks monitor the AWS systems required to use this instance and ensure they are functioning properly.

System reachability check passed

Additional Resources

[Submit feedback](#) if our checks do not reflect your experience with this instance or if they do not detect the issues you are having.

Please note that we will not respond to customer support issues reported via this form. Please post your issue on the [Developer Forums](#) or contact us.

Instance Status Checks ⓘ

These checks monitor your software and services. **Instance reachability check failed at Oct 20, 2018**. [Learn more about this issue](#)

If you have an instance with a failed status check and the instance has been unreachable for over 20 minutes, choose **AWS Support** to submit a request for assistance. To troubleshoot system or instance status check failures yourself, see [Troubleshooting instances with failed status checks \(p. 1256\)](#).

5. To review the CloudWatch metrics for status checks, select the instance, and then choose the **Monitoring** tab. Scroll until you see the graphs for the following metrics:
 - **Status Check Failed (Any)**
 - **Status Check Failed (Instance)**
 - **Status Check Failed (System)**

Viewing status using the command line

You can view status checks for running instances using the `describe-instance-status` (AWS CLI) command.

To view the status of all instances, use the following command.

```
aws ec2 describe-instance-status
```

To get the status of all instances with an instance status of `impaired`, use the following command.

```
aws ec2 describe-instance-status \
--filters Name=instance-status.status,Values=impaired
```

To get the status of a single instance, use the following command.

```
aws ec2 describe-instance-status \
--instance-ids i-1234567890abcdef0
```

Alternatively, use the following commands:

- [Get-EC2InstanceState](#) (AWS Tools for Windows PowerShell)
- [DescribeInstanceState](#) (Amazon EC2 Query API)

If you have an instance with a failed status check, see [Troubleshooting instances with failed status checks \(p. 1256\)](#).

Reporting instance status

You can provide feedback if you are having problems with an instance whose status is not shown as impaired, or if you want to send AWS additional details about the problems you are experiencing with an impaired instance.

We use reported feedback to identify issues impacting multiple customers, but do not respond to individual account issues. Providing feedback does not change the status check results that you currently see for the instance.

Reporting status feedback using the console

New console

To report instance status (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose the **Status Checks** tab, choose **Actions** (the second **Actions** menu in the bottom half of the page), and then choose **Report instance status**.
4. Complete the **Report instance status** form, and then choose **Submit**.

Old console

To report instance status (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose the **Status Checks** tab, and choose **Submit feedback**.
4. Complete the **Report Instance Status** form, and then choose **Submit**.

Reporting status feedback using the command line

Use the `report-instance-status` (AWS CLI) command to send feedback about the status of an impaired instance.

```
aws ec2 report-instance-status \
  --instances i-1234567890abcdef0 \
  --status impaired \
  --reason-codes code
```

Alternatively, use the following commands:

- `Send-EC2InstanceState` (AWS Tools for Windows PowerShell)
- `ReportInstanceState` (Amazon EC2 Query API)

Creating and editing status check alarms

You can use the [status check metrics \(p. 723\)](#) to create CloudWatch alarms to notify you when an instance has a failed status check.

Creating a status check alarm using the console

Use the following procedure to configure an alarm that sends you a notification by email, or stops, terminates, or recovers an instance when it fails a status check.

New console

To create a status check alarm (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose the **Status Checks** tab, and choose **Actions**, **Create status check alarm**.
4. On the **Manage CloudWatch alarms** page, under **Add or edit alarm**, choose **Create a new alarm**.
5. For **Alarm notification**, turn the toggle on to configure Amazon Simple Notification Service (Amazon SNS) notifications. Select an existing Amazon SNS topic or enter a name to create a new topic.
6. For **Alarm action**, turn the toggle on to specify an action to take when the alarm is triggered. Select the action that you'd like to take from the dropdown.
7. For **Alarm thresholds**, select the metric and criteria for the alarm. In **Consecutive Period**, set the number of periods you want to evaluate and, in **Period**, enter the evaluation period duration before triggering the alarm and sending an email.

For example, you can leave the default settings for **Group samples by (Average)** and **Type of data to sample (CPU utilization)**. You can set **Alarm When** to **>=** and enter **0 .80** for **Percent**. For **Consecutive Period**, you can enter **1**. For **Period**, you can select **5 Minutes**.

8. (Optional) For **Sample metric data**, choose **Add to dashboard**.
9. Choose **Create**.

Important

If you added an email address to the list of recipients or created a new topic, Amazon SNS sends a subscription confirmation email message to each new address. Each recipient must confirm the subscription by choosing the link contained in that message. Alert notifications are sent only to confirmed addresses.

Old console

To create a status check alarm (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose the **Status Checks** tab, and choose **Create Status Check Alarm**.
4. Select **Send a notification to**. Choose an existing SNS topic, or choose **create topic** to create a new one. If creating a new topic, in **With these recipients**, enter your email address and the addresses of any additional recipients, separated by commas.
5. (Optional) Select **Take the action**, and then select the action that you'd like to take.
6. In **Whenever**, select the status check that you want to be notified about.

If you selected **Recover this instance** in the previous step, select **Status Check Failed (System)**.

7. In **For at least**, set the number of periods you want to evaluate and in **consecutive periods**, select the evaluation period duration before triggering the alarm and sending an email.
8. (Optional) In **Name of alarm**, replace the default name with another name for the alarm.

9. Choose **Create Alarm**.

Important

If you added an email address to the list of recipients or created a new topic, Amazon SNS sends a subscription confirmation email message to each new address. Each recipient must confirm the subscription by choosing the link contained in that message. Alert notifications are sent only to confirmed addresses.

If you need to make changes to an instance status alarm, you can edit it.

New console

To edit a status check alarm using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, Monitoring, Manage CloudWatch alarms**.
4. On the **Manage CloudWatch alarms** page, under **Add or edit alarm**, choose **Edit an existing alarm**.
5. For **Search for alarm**, choose the alarm to edit.
6. Make the desired changes, and then choose **Update**.

Old console

To edit a status check alarm using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, CloudWatch Monitoring, Add/Edit Alarms**.
4. In the **Alarm Details** dialog box, choose the name of the alarm.
5. In the **Edit Alarm** dialog box, make the desired changes, and then choose **Save**.

Creating a status check alarm using the AWS CLI

In the following example, the alarm publishes a notification to an SNS topic, `arn:aws:sns:us-west-2:111122223333:my-sns-topic`, when the instance fails either the instance check or system status check for at least two consecutive periods. The CloudWatch metric used is `StatusCheckFailed`.

To create a status check alarm using the AWS CLI

1. Select an existing SNS topic or create a new one. For more information, see [Using the AWS CLI with Amazon SNS](#) in the *AWS Command Line Interface User Guide*.
2. Use the following `list-metrics` command to view the available Amazon CloudWatch metrics for Amazon EC2.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

3. Use the following `put-metric-alarm` command to create the alarm.

```
aws cloudwatch put-metric-alarm --alarm-name StatusCheckFailed-Alarm-for-i-1234567890abcdef0 --metric-name StatusCheckFailed --namespace AWS/EC2 --statistic Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 --unit Count --period 300 --evaluation-periods 2 --threshold 1 --comparison-operator
```

```
GreaterThanOrEqualToThreshold --alarm-actions arn:aws:sns:us-west-2:111122223333:my-sns-topic
```

The period is the time frame, in seconds, in which Amazon CloudWatch metrics are collected. This example uses 300, which is 60 seconds multiplied by 5 minutes. The evaluation period is the number of consecutive periods for which the value of the metric must be compared to the threshold. This example uses 2. The alarm actions are the actions to perform when this alarm is triggered. This example configures the alarm to send an email using Amazon SNS.

Scheduled events for your instances

AWS can schedule events for your instances, such as a reboot, stop/start, or retirement. These events do not occur frequently. If one of your instances will be affected by a scheduled event, AWS sends an email to the email address that's associated with your AWS account prior to the scheduled event. The email provides details about the event, including the start and end date. Depending on the event, you might be able to take action to control the timing of the event.

Scheduled events are managed by AWS; you cannot schedule events for your instances. You can view the events scheduled by AWS, customize scheduled event notifications to include or remove tags from the email notification, perform actions when an instance is scheduled to reboot, retire, or stop.

To update the contact information for your account so that you can be sure to be notified about scheduled events, go to the [Account Settings](#) page.

Contents

- [Types of scheduled events \(p. 704\)](#)
- [Viewing scheduled events \(p. 704\)](#)
- [Customizing scheduled event notifications \(p. 708\)](#)
- [Working with instances scheduled to stop or retire \(p. 711\)](#)
- [Working with instances scheduled for reboot \(p. 711\)](#)
- [Working with instances scheduled for maintenance \(p. 713\)](#)
- [Rescheduling a scheduled event \(p. 713\)](#)

Types of scheduled events

Amazon EC2 can create the following types of events for your instances, where the event occurs at a scheduled time:

- **Instance stop:** At the scheduled time, the instance is stopped. When you start it again, it's migrated to a new host. Applies only to instances backed by Amazon EBS.
- **Instance retirement:** At the scheduled time, the instance is stopped if it is backed by Amazon EBS, or terminated if it is backed by instance store.
- **Instance reboot:** At the scheduled time, the instance is rebooted.
- **System reboot:** At the scheduled time, the host for the instance is rebooted.
- **System maintenance:** At the scheduled time, the instance might be temporarily affected by network maintenance or power maintenance.

Viewing scheduled events

In addition to receiving notification of scheduled events in email, you can check for scheduled events using one of the following methods.

New console

To view scheduled events for your instances using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. You can view scheduled events in the following screens:
 - In the navigation pane, choose **Events**. Any resources with an associated event are displayed. You can filter by **Resource ID**, **Resource type**, **Availability zone**, **Event status**, or **Event type**.

The screenshot shows the 'Events' page with a search bar and three active filters: 'Resource type: instance', 'Event status: Scheduled', and 'Event type: instance-stop'. Below the filters, there is a table header with columns: Resource ID, Event status, and Event type. A single row is visible, showing 'i-02c48ffba61cd16f' under Resource ID, 'Scheduled' under Event status, and 'instance-stop' under Event type.

- Alternatively, in the navigation pane, choose **EC2 Dashboard**. Any resources with an associated event are displayed under **Scheduled events**.

The screenshot shows the 'Scheduled events' section of the EC2 Dashboard for the 'US East (N. Virginia)' region. It indicates that there are 7 instance(s) have scheduled events and 1 volume(s) are impaired.

- Some events are also shown for affected resources. For example, in the navigation pane, choose **Instances** and select an instance. If the instance has an associated instance stop or instance retirement event, it is displayed in the lower pane.

The screenshot shows a message indicating that an instance is scheduled for retirement after February 12, 2020 at 12:00:00 AM UTC+2.

Old console

To view scheduled events for your instances using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. You can view scheduled events in the following screens:
 - In the navigation pane, choose **Events**. Any resources with an associated event are displayed. You can filter by resource type, or by specific event types. You can select the resource to view details.

Filter: All resource types ▾ All event types ▾ Ongoing and scheduled ▾				
	Resource Name ▾	Resource Type ▾	Resource Id ▾	Event Type ▾
	my-instance	instance	i-c3870335	instance-stop

Event: i-c3870335

Availability Zone us-west-2a
Event type instance-stop
Event status Scheduled
Description The instance is running on degraded hardware
Start time May 22, 2015 at 5:00:00 PM UTC-7
End time

- Alternatively, in the navigation pane, choose **EC2 Dashboard**. Any resources with an associated event are displayed under **Scheduled Events**.

Scheduled Events

US West (Oregon):

1 instances have scheduled events

- Some events are also shown for affected resources. For example, in the navigation pane, choose **Instances** and select an instance. If the instance has an associated instance stop or instance retirement event, it is displayed in the lower pane.



Retiring: This instance is scheduled for retirement after May 22, 2015 at 5:00:00 PM UTC-7. ⓘ

AWS CLI

To view scheduled events for your instances using the AWS CLI

Use the `describe-instance-status` command.

```
aws ec2 describe-instance-status \
--instance-id i-1234567890abcdef0 \
--query "InstanceStatuses[].[Events]"
```

The following example output shows a reboot event.

```
[{"Events": [
  {
    "InstanceEventId": "instance-event-0d59937288b749b32",
    "Code": "system-reboot",
    "Description": "The instance is scheduled for a reboot",
    "NotAfter": "2019-03-15T22:00:00.000Z",
    "NotBefore": "2019-03-14T20:00:00.000Z",
    "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"
  }
]}
```

```
    ]  
]
```

The following example output shows an instance retirement event.

```
[  
  "Events": [  
    {  
      "InstanceEventId": "instance-event-0e439355b779n26",  
  
      "Code": "instance-stop",  
      "Description": "The instance is running on degraded hardware",  
      "NotBefore": "2015-05-23T00:00:00.000Z"  
    }  
  ]  
]
```

PowerShell

To view scheduled events for your instances using the AWS Tools for Windows PowerShell

Use the following [Get-EC2InstanceState](#) command.

```
PS C:\> (Get-EC2InstanceState -InstanceId i-1234567890abcdef0).Events
```

The following example output shows an instance retirement event.

```
Code      : instance-stop  
Description : The instance is running on degraded hardware  
NotBefore : 5/23/2015 12:00:00 AM
```

Instance metadata

To view scheduled events for your instances using instance metadata

You can retrieve information about active maintenance events for your instances from the [instance metadata \(p. 658\)](#) using Instance Metadata Service Version 2 or Instance Metadata Service Version 1.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-  
ec2-metadata-token-ttl-seconds: 21600"` \  
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-  
data/events/maintenance/scheduled
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/events/maintenance/scheduled
```

The following is example output with information about a scheduled system reboot event, in JSON format.

```
[  
  {  
    "NotBefore" : "21 Jan 2019 09:00:43 GMT",  
    "Code" : "system-reboot",  
    "Description" : "scheduled reboot",  
    "EventId" : "instance-event-0d59937288b749b32",  
    "NotAfter" : "21 Jan 2019 09:17:23 GMT",  
    "State" : "active"
```

```
}
```

To view event history about completed or canceled events for your instances using instance metadata

You can retrieve information about completed or canceled events for your instances from [instance metadata \(p. 658\)](#) using Instance Metadata Service Version 2 or Instance Metadata Service Version 1.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/events/maintenance/history
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/events/maintenance/history
```

The following is example output with information about a system reboot event that was canceled, and a system reboot event that was completed, in JSON format.

```
[
  {
    "NotBefore" : "21 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "[Canceled] scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "21 Jan 2019 09:17:23 GMT",
    "State" : "canceled"
  },
  {
    "NotBefore" : "29 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "[Completed] scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "29 Jan 2019 09:17:23 GMT",
    "State" : "completed"
  }
]
```

Customizing scheduled event notifications

You can customize scheduled event notifications to include tags in the email notification. This makes it easier to identify the affected resource (instances or Dedicated Hosts) and to prioritize actions for the upcoming event.

When you customize event notifications to include tags, you can choose to include:

- All of the tags that are associated with the affected resource
- Only specific tags that are associated with the affected resource

For example, suppose that you assign `application`, `costcenter`, `project`, and `owner` tags to all of your instances. You can choose to include all of the tags in event notifications. Alternatively, if you'd like to see only the `owner` and `project` tags in event notifications, then you can choose to include only those tags.

After you select the tags to include, the event notifications will include the resource ID (instance ID or Dedicated Host ID) and the tag key and value pairs that are associated with the affected resource.

Topics

- [Including tags in event notifications \(p. 709\)](#)
- [Removing tags from event notifications \(p. 709\)](#)
- [Viewing the tags to be included in event notifications \(p. 710\)](#)

Including tags in event notifications

The tags that you choose to include apply to all resources (instances and Dedicated Hosts) in the selected Region. To customize event notifications in other Regions, first select the required Region and then perform the following steps.

You can include tags in event notifications using one of the following methods.

New console

To include tags in event notifications

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**.
3. Choose **Actions, Manage event notifications**.
4. Select **Include resource tags in event notifications**.
5. Do one of the following, depending on the tags that you want to include in event notifications:
 - To include all of the tags associated with the affected instance or Dedicated Host, select **Include all resource tags**.
 - To manually select the tags to include, select **Choose the tags to include**, and then for **Choose the tags to include**, enter the tag key and press **Enter**.
6. Choose **Save**.

AWS CLI

To include all tags in event notifications

Use the [register-instance-event-notification-attributes](#) AWS CLI command and set the `IncludeAllTagsOfInstance` parameter to `true`.

```
aws ec2 register-instance-event-notification-attributes --instance-tag-attribute "IncludeAllTagsOfInstance=true"
```

To include specific tags in event notifications

Use the [register-instance-event-notification-attributes](#) AWS CLI command and specify the tags to include using the `InstanceTagKeys` parameter.

```
aws ec2 register-instance-event-notification-attributes --instance-tag-attribute 'InstanceTagKeys=[ "tag_key_1", "tag_key_2", "tag_key_3"]'
```

Removing tags from event notifications

You can remove tags from event notifications using one of the following methods.

New console

To remove tags from event notifications

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**.
3. Choose **Actions, Manage event notifications**.
4. Do one of the following, depending on the tag that you want to remove from event notifications.
 - To remove all tags from event notifications, clear **Include resource tags in event notifications**.
 - To remove specific tags from event notifications, choose **Remove (X)** for the tags listed below the **Choose the tags to include** field.
5. Choose **Save**.

AWS CLI

To remove all tags from event notifications

Use the [deregister-instance-event-notification-attributes](#) AWS CLI command and set the `IncludeAllTagsOfInstance` parameter to `false`.

```
aws ec2 deregister-instance-event-notification-attributes --instance-tag-attribute "IncludeAllTagsOfInstance=false"
```

To remove specific tags from event notifications

Use the [deregister-instance-event-notification-attributes](#) AWS CLI command and specify the tags to remove using the `InstanceTagKeys` parameter.

```
aws ec2 deregister-instance-event-notification-attributes --instance-tag-attribute 'InstanceTagKeys=[ "tag_key_1", "tag_key_2", "tag_key_3" ]'
```

Viewing the tags to be included in event notifications

You can view the tags that are to be included in event notifications using one of the following methods.

New console

To view the tags that are to be included in event notifications

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**.
3. Choose **Actions, Manage event notifications**.

AWS CLI

To view the tags that are to be included in event notifications

Use the [describe-instance-event-notification-attributes](#) AWS CLI command.

```
aws ec2 describe-instance-event-notification-attributes
```

Working with instances scheduled to stop or retire

When AWS detects irreparable failure of the underlying host for your instance, it schedules the instance to stop or terminate, depending on the type of root device for the instance. If the root device is an EBS volume, the instance is scheduled to stop. If the root device is an instance store volume, the instance is scheduled to terminate. For more information, see [Instance retirement \(p. 604\)](#).

Important

Any data stored on instance store volumes is lost when an instance is stopped or terminated. This includes instance store volumes that are attached to an instance that has an EBS volume as the root device. Be sure to save data from your instance store volumes that you might need later before the instance is stopped or terminated.

Actions for Instances Backed by Amazon EBS

You can wait for the instance to stop as scheduled. Alternatively, you can stop and start the instance yourself, which migrates it to a new host. For more information about stopping your instance, in addition to information about the changes to your instance configuration when it's stopped, see [Stop and start your instance \(p. 587\)](#).

You can automate an immediate stop and start in response to a scheduled instance stop event. For more information, see [Automating Actions for EC2 Instances](#) in the *AWS Health User Guide*.

Actions for Instances Backed by Instance Store

We recommend that you launch a replacement instance from your most recent AMI and migrate all necessary data to the replacement instance before the instance is scheduled to terminate. Then, you can terminate the original instance, or wait for it to terminate as scheduled.

Working with instances scheduled for reboot

When AWS must perform tasks such as installing updates or maintaining the underlying host, it can schedule the instance or the underlying host for a reboot. You can [reschedule most reboot events \(p. 713\)](#) so that your instance is rebooted at a specific date and time that suits you.

If you stop your linked [EC2-Classic instance \(p. 895\)](#), it is automatically unlinked from the VPC and the VPC security groups are no longer associated with the instance. You can link your instance to the VPC again after you've restarted it.

Viewing the reboot event type

You can view whether a reboot event is an instance reboot or a system reboot using one of the following methods.

New console

To view the type of scheduled reboot event using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**.
3. Choose **Resource type: instance** from the filter list.
4. For each instance, view the value in the **Event type** column. The value is either **system-reboot** or **instance-reboot**.

Old console

To view the type of scheduled reboot event using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Events**.
3. Choose **Instance resources** from the filter list.
4. For each instance, view the value in the **Event Type** column. The value is either **system-reboot** or **instance-reboot**.

AWS CLI

To view the type of scheduled reboot event using the AWS CLI

Use the [describe-instance-status](#) command.

```
aws ec2 describe-instance-status --instance-id i-1234567890abcdef0
```

For scheduled reboot events, the value for `Code` is either `system-reboot` or `instance-reboot`. The following example output shows a `system-reboot` event.

```
[  
  "Events": [  
    {  
      "InstanceEventId": "instance-event-0d59937288b749b32",  
      "Code": "system-reboot",  
      "Description": "The instance is scheduled for a reboot",  
      "NotAfter": "2019-03-14T22:00:00.000Z",  
      "NotBefore": "2019-03-14T20:00:00.000Z",  
      "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"  
    }  
  ]  
]
```

Actions for instance reboot

You can wait for the instance reboot to occur within its scheduled maintenance window, [reschedule \(p. 713\)](#) the instance reboot to a date and time that suits you, or [reboot \(p. 603\)](#) the instance yourself at a time that is convenient for you.

After your instance is rebooted, the scheduled event is cleared and the event's description is updated. The pending maintenance to the underlying host is completed, and you can begin using your instance again after it has fully booted.

Actions for system reboot

It is not possible for you to reboot the system yourself. You can wait for the system reboot to occur during its scheduled maintenance window, or you can [reschedule \(p. 713\)](#) the system reboot to a date and time that suits you. A system reboot typically completes in a matter of minutes. After the system reboot has occurred, the instance retains its IP address and DNS name, and any data on local instance store volumes is preserved. After the system reboot is complete, the scheduled event for the instance is cleared, and you can verify that the software on your instance is operating as expected.

Alternatively, if it is necessary to maintain the instance at a different time and you can't reschedule the system reboot, then you can stop and start an Amazon EBS-backed instance, which migrates it to a new host. However, the data on the local instance store volumes is not preserved. You can also automate an immediate instance stop and start in response to a scheduled system reboot event. For more information, see [Automating Actions for EC2 Instances](#) in the *AWS Health User Guide*. For an instance store-backed instance, if you can't reschedule the system reboot, then you can launch a replacement instance from your most recent AMI, migrate all necessary data to the replacement instance before the scheduled maintenance window, and then terminate the original instance.

Working with instances scheduled for maintenance

When AWS must maintain the underlying host for an instance, it schedules the instance for maintenance. There are two types of maintenance events: network maintenance and power maintenance.

During network maintenance, scheduled instances lose network connectivity for a brief period of time. Normal network connectivity to your instance is restored after maintenance is complete.

During power maintenance, scheduled instances are taken offline for a brief period, and then rebooted. When a reboot is performed, all of your instance's configuration settings are retained.

After your instance has rebooted (this normally takes a few minutes), verify that your application is working as expected. At this point, your instance should no longer have a scheduled event associated with it, or if it does, the description of the scheduled event begins with **[Completed]**. It sometimes takes up to 1 hour for the instance status description to refresh. Completed maintenance events are displayed on the Amazon EC2 console dashboard for up to a week.

Actions for Instances Backed by Amazon EBS

You can wait for the maintenance to occur as scheduled. Alternatively, you can stop and start the instance, which migrates it to a new host. For more information about stopping your instance, in addition to information about the changes to your instance configuration when it's stopped, see [Stop and start your instance \(p. 587\)](#).

You can automate an immediate stop and start in response to a scheduled maintenance event. For more information, see [Automating Actions for EC2 Instances](#) in the *AWS Health User Guide*.

Actions for instances backed by instance store

You can wait for the maintenance to occur as scheduled. Alternatively, if you want to maintain normal operation during a scheduled maintenance window, you can launch a replacement instance from your most recent AMI, migrate all necessary data to the replacement instance before the scheduled maintenance window, and then terminate the original instance.

Rescheduling a scheduled event

You can reschedule an event so that it occurs at a specific date and time that suits you. Only events that have a deadline date can be rescheduled. There are other [limitations for rescheduling an event \(p. 715\)](#).

You can reschedule an event using one of the following methods.

New console

To reschedule an event using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**.
3. Choose **Resource type: instance** from the filter list.
4. Select one or more instances, and then choose **Actions, Schedule event**.

Only events that have an event deadline date, indicated by a value for **Deadline**, can be rescheduled. If one of the selected events does not have a deadline date, **Actions, Schedule event** is disabled.

5. For **New start time**, enter a new date and time for the event. The new date and time must occur before the **Event deadline**.
6. Choose **Save**.

It might take 1-2 minutes for the updated event start time to be reflected in the console.

Old console

To reschedule an event using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**.
3. Choose **Instance resources** from the filter list.
4. Select one or more instances, and then choose **Actions, Schedule Event**.

Only events that have an event deadline date, indicated by a value for **Event Deadline**, can be rescheduled.

5. For **Event start time**, enter a new date and time for the event. The new date and time must occur before the **Event Deadline**.
6. Choose **Schedule Event**.

It might take 1-2 minutes for the updated event start time to be reflected in the console.

AWS CLI

To reschedule an event using the AWS CLI

1. Only events that have an event deadline date, indicated by a value for `NotBeforeDeadline`, can be rescheduled. Use the `describe-instance-status` command to view the `NotBeforeDeadline` parameter value.

```
aws ec2 describe-instance-status --instance-id i-1234567890abcdef0
```

The following example output shows a system-reboot event that can be rescheduled because `NotBeforeDeadline` contains a value.

```
[  
    "Events": [  
        {  
            "InstanceEventId": "instance-event-0d59937288b749b32",  
            "Code": "system-reboot",  
            "Description": "The instance is scheduled for a reboot",  
            "NotAfter": "2019-03-14T22:00:00.000Z",  
            "NotBefore": "2019-03-14T20:00:00.000Z",  
            "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"  
        }  
    ]  
]
```

2. To reschedule the event, use the `modify-instance-event-start-time` command. Specify the new event start time using the `not-before` parameter. The new event start time must fall before the `NotBeforeDeadline`.

```
aws ec2 modify-instance-event-start-time --instance-id i-1234567890abcdef0  
--instance-event-id instance-event-0d59937288b749b32 --not-  
before 2019-03-25T10:00:00.000
```

It might take 1-2 minutes before the `describe-instance-status` command returns the updated `not-before` parameter value.

Limitations

- Only events with an event deadline date can be rescheduled. The event can be rescheduled up to the event deadline date. The **Deadline** column in the console and the `NotBeforeDeadline` field in the AWS CLI indicate if the event has a deadline date.
- Only events that have not yet started can be rescheduled. The **Start time** column in the console and the `NotBefore` field in the AWS CLI indicate the event start time. Events that are scheduled to start in the next 5 minutes cannot be rescheduled.
- The new event start time must be at least 60 minutes from the current time.
- If you reschedule multiple events using the console, the event deadline date is determined by the event with the earliest event deadline date.

Monitoring your instances using CloudWatch

You can monitor your instances using Amazon CloudWatch, which collects and processes raw data from Amazon EC2 into readable, near real-time metrics. These statistics are recorded for a period of 15 months, so that you can access historical information and gain a better perspective on how your web application or service is performing.

By default, Amazon EC2 sends metric data to CloudWatch in 5-minute periods. To send metric data for your instance to CloudWatch in 1-minute periods, you can enable detailed monitoring on the instance. For more information, see [Enable or turn off detailed monitoring for your instances \(p. 715\)](#).

The Amazon EC2 console displays a series of graphs based on the raw data from Amazon CloudWatch. Depending on your needs, you might prefer to get data for your instances from Amazon CloudWatch instead of the graphs in the console.

For more information about Amazon CloudWatch, see the [Amazon CloudWatch User Guide](#).

Contents

- [Enable or turn off detailed monitoring for your instances \(p. 715\)](#)
- [List the available CloudWatch metrics for your instances \(p. 717\)](#)
- [Get statistics for metrics for your instances \(p. 728\)](#)
- [Graph metrics for your instances \(p. 736\)](#)
- [Create a CloudWatch alarm for an instance \(p. 736\)](#)
- [Create alarms that stop, terminate, reboot, or recover an instance \(p. 738\)](#)

Enable or turn off detailed monitoring for your instances

By default, your instance is enabled for basic monitoring. You can optionally enable detailed monitoring. After you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period for the instance.

The following describes the data interval and charge for basic and detailed monitoring for instances.

Basic monitoring

Data is available automatically in 5-minute periods at no charge.

Detailed monitoring

Data is available in 1-minute periods for an additional charge.

To get this level of data, you must specifically enable it for the instance. For the instances where you've enabled detailed monitoring, you can also get aggregated data across groups of similar instances.

Charges for detailed monitoring

If you enable detailed monitoring, you are charged per metric that is sent to CloudWatch. You are not charged for data storage. For more information about pricing for detailed monitoring, see **Paid tier** on the [Amazon CloudWatch pricing page](#). For a pricing example, see **Example 1 - EC2 Detailed Monitoring** on the [Amazon CloudWatch pricing page](#).

Enabling detailed monitoring

You can enable detailed monitoring on an instance as you launch it or after the instance is running or stopped. Enabling detailed monitoring on an instance does not affect the monitoring of the EBS volumes attached to the instance. For more information, see [Amazon CloudWatch metrics for Amazon EBS \(p. 1172\)](#).

New console

To enable detailed monitoring for an existing instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, Monitoring, Manage detailed monitoring**.
4. On the **Detailed monitoring** detail page, for **Detailed monitoring**, select the **Enable** check box.
5. Choose **Save**.

Old console

To enable detailed monitoring for an existing instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, CloudWatch Monitoring, Enable Detailed Monitoring**.
4. In the **Enable Detailed Monitoring** dialog box, choose **Yes, Enable**.
5. Choose **Close**.

To enable detailed monitoring when launching an instance (console)

When launching an instance using the AWS Management Console, select the **Monitoring** check box on the **Configure Instance Details** page.

To enable detailed monitoring for an existing instance (AWS CLI)

Use the following [monitor-instances](#) command to enable detailed monitoring for the specified instances.

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

To enable detailed monitoring when launching an instance (AWS CLI)

Use the [run-instances](#) command with the **--monitoring** flag to enable detailed monitoring.

```
aws ec2 run-instances --image-id ami-09092360 --monitoring Enabled=true...
```

Turning off detailed monitoring

You can turn off detailed monitoring on an instance as you launch it or after the instance is running or stopped.

New console

To turn off detailed monitoring (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, Monitoring, Manage detailed monitoring**.
4. On the **Detailed monitoring** detail page, for **Detailed monitoring**, clear the **Enable** check box.
5. Choose **Save**.

Old console

To turn off detailed monitoring (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, CloudWatch Monitoring, Disable Detailed Monitoring**.
4. In the **Disable Detailed Monitoring** dialog box, choose **Yes, Disable**.
5. Choose **Close**.

To turn off detailed monitoring (AWS CLI)

Use the following `unmonitor-instances` command to turn off detailed monitoring for the specified instances.

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

List the available CloudWatch metrics for your instances

Amazon EC2 sends metrics to Amazon CloudWatch. You can use the AWS Management Console, the AWS CLI, or an API to list the metrics that Amazon EC2 sends to CloudWatch. By default, each data point covers the 5 minutes that follow the start time of activity for the instance. If you've enabled detailed monitoring, each data point covers the next minute of activity from the start time.

For information about getting the statistics for these metrics, see [Get statistics for metrics for your instances \(p. 728\)](#).

Contents

- [Instance metrics \(p. 718\)](#)
- [CPU credit metrics \(p. 720\)](#)
- [Amazon EBS metrics for Nitro-based instances \(p. 721\)](#)
- [Status check metrics \(p. 723\)](#)
- [Traffic mirroring metrics \(p. 723\)](#)
- [Amazon EC2 metric dimensions \(p. 723\)](#)
- [Amazon EC2 usage metrics \(p. 724\)](#)

- [Listing metrics using the console \(p. 725\)](#)
- [Listing metrics using the AWS CLI \(p. 727\)](#)

Instance metrics

The AWS/EC2 namespace includes the following instance metrics.

Metric	Description
CPUUtilization	<p>The percentage of allocated EC2 compute units that are currently in use on the instance. This metric identifies the processing power required to run an application on a selected instance.</p> <p>Depending on the instance type, tools in your operating system can show a lower percentage than CloudWatch when the instance is not allocated a full processor core.</p> <p>Units: Percent</p>
DiskReadOps	<p>Completed read operations from all instance store volumes available to the instance in a specified period of time.</p> <p>To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period.</p> <p>If there are no instance store volumes, either the value is 0 or the metric is not reported.</p> <p>Units: Count</p>
DiskWriteOps	<p>Completed write operations to all instance store volumes available to the instance in a specified period of time.</p> <p>To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period.</p> <p>If there are no instance store volumes, either the value is 0 or the metric is not reported.</p> <p>Units: Count</p>
DiskReadBytes	<p>Bytes read from all instance store volumes available to the instance.</p> <p>This metric is used to determine the volume of the data the application reads from the hard disk of the instance. This can be used to determine the speed of the application.</p> <p>The number reported is the number of bytes received during the period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to find Bytes/second. If you have detailed (one-minute) monitoring, divide it by 60.</p> <p>If there are no instance store volumes, either the value is 0 or the metric is not reported.</p> <p>Units: Bytes</p>

Metric	Description
DiskWriteBytes	<p>Bytes written to all instance store volumes available to the instance.</p> <p>This metric is used to determine the volume of the data the application writes onto the hard disk of the instance. This can be used to determine the speed of the application.</p> <p>The number reported is the number of bytes received during the period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to find Bytes/second. If you have detailed (one-minute) monitoring, divide it by 60.</p> <p>If there are no instance store volumes, either the value is 0 or the metric is not reported.</p> <p>Units: Bytes</p>
NetworkIn	<p>The number of bytes received on all network interfaces by the instance. This metric identifies the volume of incoming network traffic to a single instance.</p> <p>The number reported is the number of bytes received during the period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to find Bytes/second. If you have detailed (one-minute) monitoring, divide it by 60.</p> <p>Units: Bytes</p>
NetworkOut	<p>The number of bytes sent out on all network interfaces by the instance. This metric identifies the volume of outgoing network traffic from a single instance.</p> <p>The number reported is the number of bytes sent during the period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to find Bytes/second. If you have detailed (one-minute) monitoring, divide it by 60.</p> <p>Units: Bytes</p>
NetworkPacketsIn	<p>The number of packets received on all network interfaces by the instance. This metric identifies the volume of incoming traffic in terms of the number of packets on a single instance. This metric is available for basic monitoring only.</p> <p>Units: Count</p> <p>Statistics: Minimum, Maximum, Average</p>
NetworkPacketsOut	<p>The number of packets sent out on all network interfaces by the instance. This metric identifies the volume of outgoing traffic in terms of the number of packets on a single instance. This metric is available for basic monitoring only.</p> <p>Units: Count</p> <p>Statistics: Minimum, Maximum, Average</p>

Metric	Description
MetadataNoToken	<p>The number of times the instance metadata service was successfully accessed using a method that does not use a token.</p> <p>This metric is used to determine if there are any processes accessing instance metadata that are using Instance Metadata Service Version 1, which does not use a token. If all requests use token-backed sessions, i.e., Instance Metadata Service Version 2, the value is 0. For more information, see Transitioning to using Instance Metadata Service Version 2 (p. 660).</p> <p>Units: Count</p>

CPU credit metrics

The AWS/EC2 namespace includes the following CPU credit metrics for your [burstable performance instances \(p. 219\)](#).

Metric	Description
CPUCreditUsage	<p>The number of CPU credits spent by the instance for CPU utilization. One CPU credit equals one vCPU running at 100% utilization for one minute or an equivalent combination of vCPUs, utilization, and time (for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilization for two minutes).</p> <p>CPU credit metrics are available at a five-minute frequency only. If you specify a period greater than five minutes, use the <code>Sum</code> statistic instead of the <code>Average</code> statistic.</p> <p>Units: Credits (vCPU-minutes)</p>
CPUCreditBalance	<p>The number of earned CPU credits that an instance has accrued since it was launched or started. For T2 Standard, the CPUCreditBalance also includes the number of launch credits that have been accrued.</p> <p>Credits are accrued in the credit balance after they are earned, and removed from the credit balance when they are spent. The credit balance has a maximum limit, determined by the instance size. After the limit is reached, any new credits that are earned are discarded. For T2 Standard, launch credits do not count towards the limit.</p> <p>The credits in the CPUCreditBalance are available for the instance to spend to burst beyond its baseline CPU utilization.</p> <p>When an instance is running, credits in the CPUCreditBalance do not expire. When a T3 or T3a instance stops, the CPUCreditBalance value persists for seven days. Thereafter, all accrued credits are lost. When a T2 instance stops, the CPUCreditBalance value does not persist, and all accrued credits are lost.</p> <p>CPU credit metrics are available at a five-minute frequency only.</p>

Metric	Description
	Units: Credits (vCPU-minutes)
CPUSurplusCreditBalance	<p>The number of surplus credits that have been spent by an unlimited instance when its CPUCreditBalance value is zero.</p> <p>The CPUSurplusCreditBalance value is paid down by earned CPU credits. If the number of surplus credits exceeds the maximum number of credits that the instance can earn in a 24-hour period, the spent surplus credits above the maximum incur an additional charge.</p> <p>CPU credit metrics are available at a five-minute frequency only.</p> <p>Units: Credits (vCPU-minutes)</p>
CPUSurplusCreditsCharged	<p>The number of spent surplus credits that are not paid down by earned CPU credits, and which thus incur an additional charge.</p> <p>Spent surplus credits are charged when any of the following occurs:</p> <ul style="list-style-type: none"> • The spent surplus credits exceed the maximum number of credits that the instance can earn in a 24-hour period. Spent surplus credits above the maximum are charged at the end of the hour. • The instance is stopped or terminated. • The instance is switched from unlimited to standard. <p>CPU credit metrics are available at a five-minute frequency only.</p> <p>Units: Credits (vCPU-minutes)</p>

Amazon EBS metrics for Nitro-based instances

The AWS/EC2 namespace includes the following Amazon EBS metrics for the Nitro-based instances that are not bare metal instances. For the list of Nitro-based instance types, see [Instances built on the Nitro System \(p. 205\)](#).

Metric values for Nitro-based instances will always be integers (whole numbers), whereas values for Xen-based instances support decimals. Therefore, low instance CPU utilization on Nitro-based instances may appear to be rounded down to 0.

Metric	Description
EBSReadOps	<p>Completed read operations from all Amazon EBS volumes attached to the instance in a specified period of time.</p> <p>To calculate the average read I/O operations per second (Read IOPS) for the period, divide the total operations in the period by the number of seconds in that period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to calculate the Read IOPS. If you have detailed (one-minute) monitoring, divide it by 60.</p>

Metric	Description
	Unit: Count
EBSWriteOps	<p>Completed write operations to all EBS volumes attached to the instance in a specified period of time.</p> <p>To calculate the average write I/O operations per second (Write IOPS) for the period, divide the total operations in the period by the number of seconds in that period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to calculate the Write IOPS. If you have detailed (one-minute) monitoring, divide it by 60.</p> <p>Unit: Count</p>
EBSReadBytes	<p>Bytes read from all EBS volumes attached to the instance in a specified period of time.</p> <p>The number reported is the number of bytes read during the period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to find Read Bytes/second. If you have detailed (one-minute) monitoring, divide it by 60.</p> <p>Unit: Bytes</p>
EBSWriteBytes	<p>Bytes written to all EBS volumes attached to the instance in a specified period of time.</p> <p>The number reported is the number of bytes written during the period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to find Write Bytes/second. If you have detailed (one-minute) monitoring, divide it by 60.</p> <p>Unit: Bytes</p>
EBSIOBalance%	<p>Available only for the smaller instance sizes. Provides information about the percentage of I/O credits remaining in the burst bucket. This metric is available for basic monitoring only.</p> <p>The Sum statistic is not applicable to this metric.</p> <p>Unit: Percent</p>
EBSByteBalance%	<p>Available only for the smaller instance sizes. Provides information about the percentage of throughput credits remaining in the burst bucket. This metric is available for basic monitoring only.</p> <p>The Sum statistic is not applicable to this metric.</p> <p>Unit: Percent</p>

For information about the metrics provided for your EBS volumes, see [Amazon EBS metrics \(p. 1172\)](#).
For information about the metrics provided for your Spot fleets, see [CloudWatch metrics for Spot Fleet \(p. 412\)](#).

Status check metrics

The AWS/EC2 namespace includes the following status check metrics. By default, status check metrics are available at a 1-minute frequency at no charge. For a newly-launched instance, status check metric data is only available after the instance has completed the initialization state (within a few minutes of the instance entering the running state). For more information about EC2 status checks, see [Status checks for your instances \(p. 697\)](#).

Metric	Description
StatusCheckFailed	<p>Reports whether the instance has passed both the instance status check and the system status check in the last minute.</p> <p>This metric can be either 0 (passed) or 1 (failed).</p> <p>By default, this metric is available at a 1-minute frequency at no charge.</p> <p>Units: Count</p>
StatusCheckFailed_Instance	<p>Reports whether the instance has passed the instance status check in the last minute.</p> <p>This metric can be either 0 (passed) or 1 (failed).</p> <p>By default, this metric is available at a 1-minute frequency at no charge.</p> <p>Units: Count</p>
StatusCheckFailed_System	<p>Reports whether the instance has passed the system status check in the last minute.</p> <p>This metric can be either 0 (passed) or 1 (failed).</p> <p>By default, this metric is available at a 1-minute frequency at no charge.</p> <p>Units: Count</p>

Traffic mirroring metrics

The AWS/EC2 namespace includes metrics for mirrored traffic. For more information, see [Monitoring mirrored traffic using Amazon CloudWatch](#) in the *Amazon VPC Traffic Mirroring Guide*.

Amazon EC2 metric dimensions

You can use the following dimensions to refine the metrics listed in the previous tables.

Dimension	Description
AutoScalingGroupName	This dimension filters the data you request for all instances in a specified capacity group. An <i>Auto Scaling group</i> is a collection of

Dimension	Description
	instances you define if you're using Auto Scaling. This dimension is available only for Amazon EC2 metrics when the instances are in such an Auto Scaling group. Available for instances with Detailed or Basic Monitoring enabled.
<code>ImageId</code>	This dimension filters the data you request for all instances running this Amazon EC2 Amazon Machine Image (AMI). Available for instances with Detailed Monitoring enabled.
<code>InstanceId</code>	This dimension filters the data you request for the identified instance only. This helps you pinpoint an exact instance from which to monitor data.
<code>InstanceType</code>	This dimension filters the data you request for all instances running with this specified instance type. This helps you categorize your data by the type of instance running. For example, you might compare data from an m1.small instance and an m1.large instance to determine which has the better business value for your application. Available for instances with Detailed Monitoring enabled.

Amazon EC2 usage metrics

You can use CloudWatch usage metrics to provide visibility into your account's usage of resources. Use these metrics to visualize your current service usage on CloudWatch graphs and dashboards.

Amazon EC2 usage metrics correspond to AWS service quotas. You can configure alarms that alert you when your usage approaches a service quota. For more information about CloudWatch integration with service quotas, see [Service Quotas Integration and Usage Metrics](#).

Amazon EC2 publishes the following metrics in the `AWS/Usage` namespace.

Metric	Description
<code>ResourceCount</code>	The number of the specified resources running in your account. The resources are defined by the dimensions associated with the metric. The most useful statistic for this metric is <code>MAXIMUM</code> , which represents the maximum number of resources used during the 1-minute period.

The following dimensions are used to refine the usage metrics that are published by Amazon EC2.

Dimension	Description
<code>Service</code>	The name of the AWS service containing the resource. For Amazon EC2 usage metrics, the value for this dimension is <code>EC2</code> .
<code>Type</code>	The type of entity that is being reported. Currently, the only valid value for Amazon EC2 usage metrics is <code>Resource</code> .
<code>Resource</code>	The type of resource that is running. Currently, the only valid value for Amazon EC2 usage metrics is <code>vCPU</code> , which returns information on instances that are running.

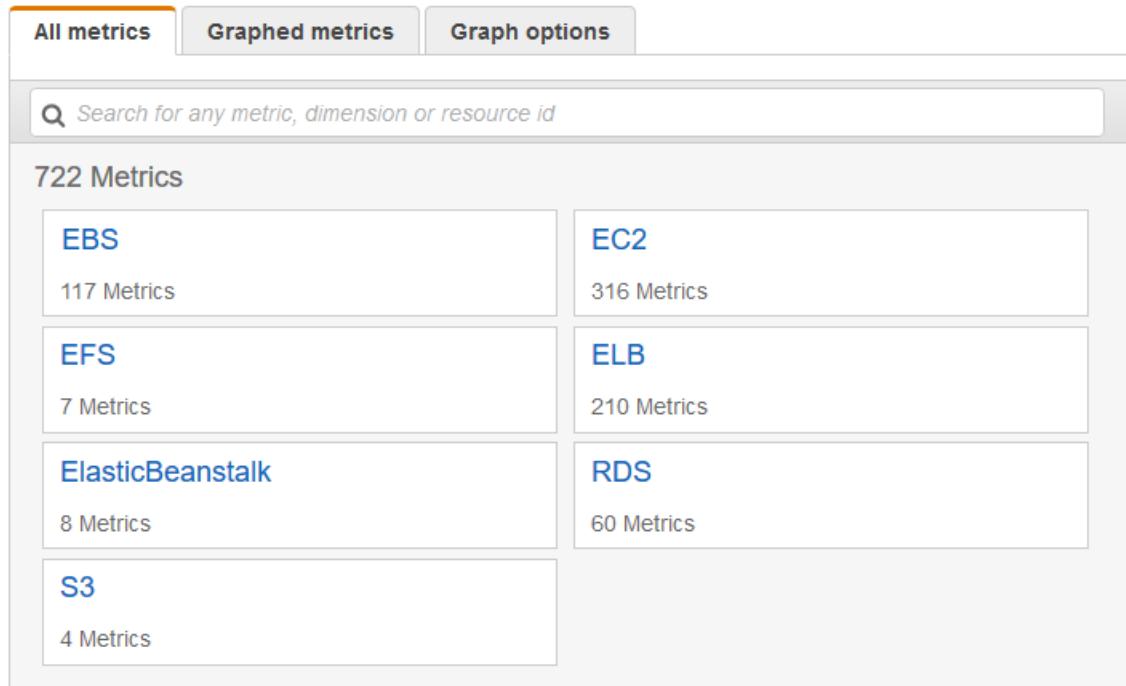
Dimension	Description
Class	The class of resource being tracked. For Amazon EC2 usage metrics with vCPU as the value of the Resource dimension, the valid values are Standard/OnDemand, F/OnDemand, G/OnDemand, Inf/OnDemand, P/OnDemand, and X/OnDemand. The values for this dimension define the first letter of the instance types that are reported by the metric. For example, Standard/OnDemand returns information about all running instances with types that start with A, C, D, H, I, M, R, T, and Z, and G/OnDemand returns information about all running instances with types that start with G.

List available metrics using the console

Metrics are grouped first by namespace, and then by the various dimension combinations within each namespace. For example, you can view all metrics provided by Amazon EC2, or metrics grouped by instance ID, instance type, image (AMI) ID, or Auto Scaling group.

To view available metrics by category (console)

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** metric namespace.



4. Select a metric dimension (for example, **Per-Instance Metrics**).

The screenshot shows the Amazon CloudWatch Metrics console interface. At the top, there are three tabs: "All metrics" (which is selected), "Graphed metrics", and "Graph options". Below the tabs, the navigation bar shows "All > EC2" and a search bar with the placeholder "Search for any metric, dimension or resource id". The main content area displays "103 Metrics" and lists them in five categories:

- By Auto Scaling Group**: 28 Metrics
- By Image (AMI) Id**: 7 Metrics
- Per-Instance Metrics**: 54 Metrics
- Aggregated by Instance Type**: 7 Metrics
- Across All Instances**: 7 Metrics

5. To sort the metrics, use the column heading. To graph a metric, select the check box next to the metric. To filter by resource, choose the resource ID and then choose **Add to search**. To filter by metric, choose the metric name and then choose **Add to search**.

The screenshot shows the AWS CloudWatch Metrics console. At the top, there are three tabs: "All metrics", "Graphed metrics", and "Graph options". Below the tabs, the navigation path is "All > EC2 > Per-Instance Metrics". A search bar is present with the placeholder "Search for any metric, dimension or resource id". The main area displays a table of metrics for instances. One row is selected, showing the Instance Name as "my-instance" and the InstanceId as "i-abbc12a7". A context menu is open over the InstanceId column, with the cursor hovering over the "Jump to resource" option. The menu items are: "Add to search", "Search for this only", "Add to graph", "Graph this metric only", "Graph all search results", and "Jump to resource".

	Instance Name (192)	InstanceId	Metric Name
<input type="checkbox"/>	my-instance	i-abbc12a7	CPUUtilization
<input type="checkbox"/>	my-instance		DiskReadBytes
<input type="checkbox"/>	my-instance		DiskReadOps
<input type="checkbox"/>	my-instance		DiskWriteBytes
<input type="checkbox"/>	my-instance		DiskWriteOps
<input type="checkbox"/>	my-instance		NetworkIn
<input type="checkbox"/>	my-instance		NetworkOut
<input type="checkbox"/>	my-instance	i-abbc12a7	NetworkPacketsIn
<input type="checkbox"/>	my-instance	i-abbc12a7	NetworkPacketsOut

Listing metrics using the AWS CLI

Use the [list-metrics](#) command to list the CloudWatch metrics for your instances.

To list all the available metrics for Amazon EC2 (AWS CLI)

The following example specifies the AWS/EC2 namespace to view all the metrics for Amazon EC2.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

The following is example output:

```
{  
    "Metrics": [  
        {  
            "Namespace": "AWS/EC2",  
            "Dimensions": [  
                {  
                    "Name": "InstanceId",  
                    "Value": "i-1234567890abcdef0"  
                }  
            ],  
            "MetricName": "NetworkOut"  
        },  
        {  
            "Namespace": "AWS/EC2",  
            "Dimensions": [  
                {  
                    "Name": "InstanceId",  
                    "Value": "i-1234567890abcdef0"  
                }  
            ],  
            "MetricName": "NetworkIn"  
        }  
    ]  
}
```

```
        "MetricName": "CPUUtilization"
    },
{
    "Namespace": "AWS/EC2",
    "Dimensions": [
        {
            "Name": "InstanceId",
            "Value": "i-1234567890abcdef0"
        }
    ],
    "MetricName": "NetworkIn"
},
...
]
```

To list all the available metrics for an instance (AWS CLI)

The following example specifies the AWS/EC2 namespace and the `InstanceId` dimension to view the results for the specified instance only.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --dimensions
    Name=InstanceId,Value=i-1234567890abcdef0
```

To list a metric across all instances (AWS CLI)

The following example specifies the AWS/EC2 namespace and a metric name to view the results for the specified metric only.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --metric-name CPUUtilization
```

Get statistics for metrics for your instances

You can get statistics for the CloudWatch metrics for your instances.

Contents

- [Statistics overview \(p. 728\)](#)
- [Get statistics for a specific instance \(p. 729\)](#)
- [Aggregate statistics across instances \(p. 732\)](#)
- [Aggregate statistics by Auto Scaling group \(p. 734\)](#)
- [Aggregate statistics by AMI \(p. 735\)](#)

Statistics overview

Statistics are metric data aggregations over specified periods of time. CloudWatch provides statistics based on the metric data points provided by your custom data or provided by other services in AWS to CloudWatch. Aggregations are made using the namespace, metric name, dimensions, and the data point unit of measure, within the time period you specify. The following table describes the available statistics.

Statistic	Description
Minimum	The lowest value observed during the specified period. You can use this value to determine low volumes of activity for your application.

Statistic	Description
Maximum	The highest value observed during the specified period. You can use this value to determine high volumes of activity for your application.
Sum	All values submitted for the matching metric added together. This statistic can be useful for determining the total volume of a metric.
Average	The value of Sum / SampleCount during the specified period. By comparing this statistic with the Minimum and Maximum, you can determine the full scope of a metric and how close the average use is to the Minimum and Maximum. This comparison helps you to know when to increase or decrease your resources as needed.
SampleCount	The count (number) of data points used for the statistical calculation.
pNN.NN	The value of the specified percentile. You can specify any percentile, using up to two decimal places (for example, p95.45).

Get statistics for a specific instance

The following examples show you how to use the AWS Management Console or the AWS CLI to determine the maximum CPU utilization of a specific EC2 instance.

Requirements

- You must have the ID of the instance. You can get the instance ID using the AWS Management Console or the [describe-instances](#) command.
- By default, basic monitoring is enabled, but you can enable detailed monitoring. For more information, see [Enable or turn off detailed monitoring for your instances \(p. 715\)](#).

To display the CPU utilization for a specific instance (console)

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** metric namespace.

The screenshot shows the AWS CloudWatch Metrics console with the following interface elements:

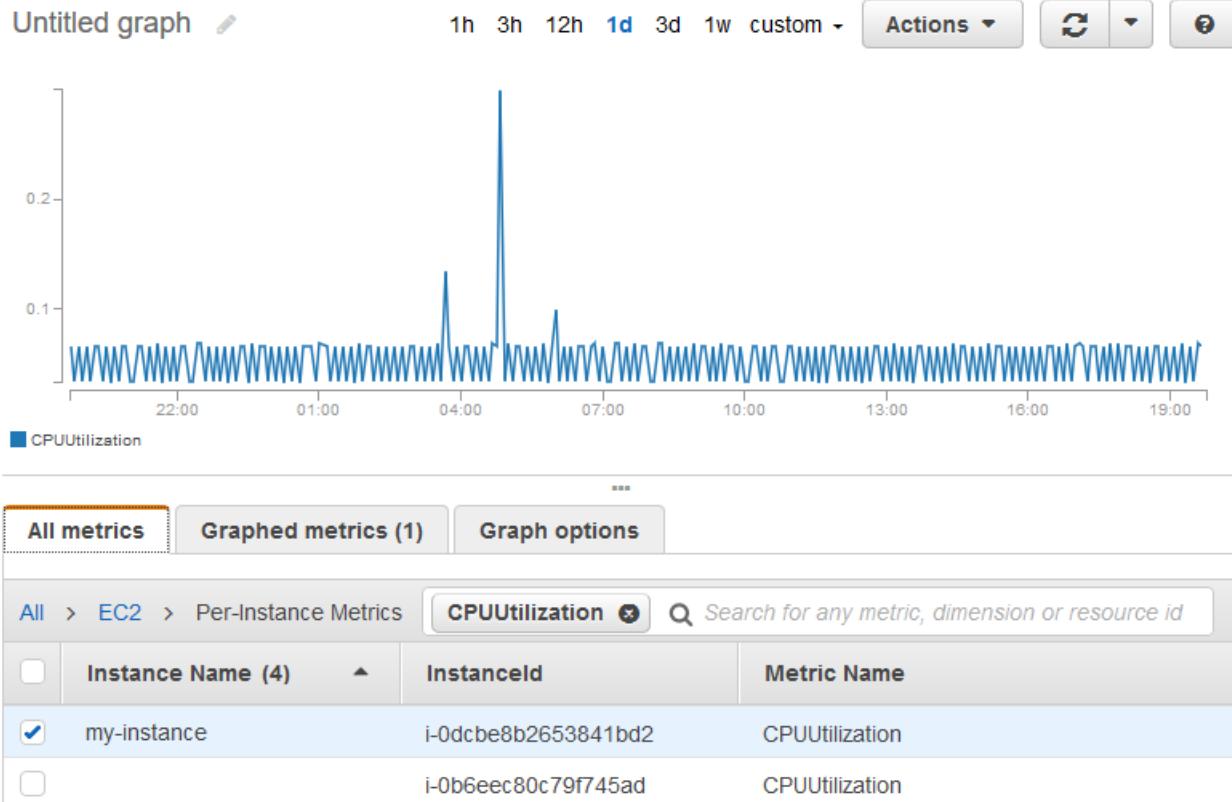
- Top Navigation:** Buttons for "All metrics", "Graphed metrics", and "Graph options".
- Search Bar:** A search input field with placeholder text "Search for any metric, dimension or resource id".
- Section Header:** "722 Metrics"
- Service Metrics Summary:** A grid of boxes showing metrics for different services:
 - EBS: 117 Metrics
 - EC2: 316 Metrics
 - EFS: 7 Metrics
 - ELB: 210 Metrics
 - ElasticBeanstalk: 8 Metrics
 - RDS: 60 Metrics
 - S3: 4 Metrics

4. Choose the **Per-Instance Metrics** dimension.

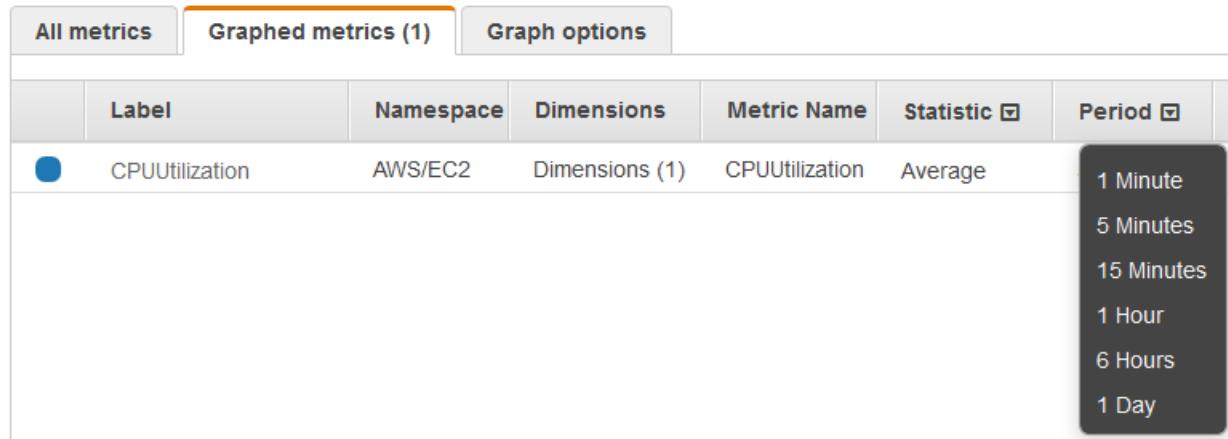
The screenshot shows the AWS CloudWatch Metrics console for the EC2 service with the following interface elements:

- Top Navigation:** Buttons for "All metrics", "Graphed metrics", and "Graph options".
- Breadcrumb:** "All > EC2"
- Search Bar:** A search input field with placeholder text "Search for any metric, dimension or resource id".
- Section Header:** "103 Metrics"
- Dimension Options:** A list of dimension categories:
 - By Auto Scaling Group
 - By Image (AMI) Id
 - Per-Instance Metrics** (highlighted in blue)
 - Aggregated by Instance Type
 - Across All Instances

- In the search field, enter **CPUutilization** and press Enter. Choose the row for the specific instance, which displays a graph for the **CPUUtilization** metric for the instance. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.



- To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.



To get the CPU utilization for a specific instance (AWS CLI)

Use the following [get-metric-statistics](#) command to get the **CPUUtilization** metric for the specified instance, using the specified period and time interval:

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization --  
period 3600 \  
--statistics Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 \  
--start-time 2016-10-18T23:18:00 --end-time 2016-10-19T23:18:00
```

The following is example output. Each value represents the maximum CPU utilization percentage for a single EC2 instance.

```
{  
    "Datapoints": [  
        {  
            "Timestamp": "2016-10-19T00:18:00Z",  
            "Maximum": 0.3300000000000002,  
            "Unit": "Percent"  
        },  
        {  
            "Timestamp": "2016-10-19T03:18:00Z",  
            "Maximum": 99.67000000000002,  
            "Unit": "Percent"  
        },  
        {  
            "Timestamp": "2016-10-19T07:18:00Z",  
            "Maximum": 0.3400000000000002,  
            "Unit": "Percent"  
        },  
        {  
            "Timestamp": "2016-10-19T12:18:00Z",  
            "Maximum": 0.3400000000000002,  
            "Unit": "Percent"  
        },  
        ...  
    ],  
    "Label": "CPUUtilization"  
}
```

Aggregate statistics across instances

Aggregate statistics are available for the instances that have detailed monitoring enabled. Instances that use basic monitoring are not included in the aggregates. In addition, Amazon CloudWatch does not aggregate data across regions. Therefore, metrics are completely separate between regions. Before you can get statistics aggregated across instances, you must enable detailed monitoring (at an additional charge), which provides data in 1-minute periods.

This example shows you how to use detailed monitoring to get the average CPU usage for your EC2 instances. Because no dimension is specified, CloudWatch returns statistics for all dimensions in the AWS/EC2 namespace.

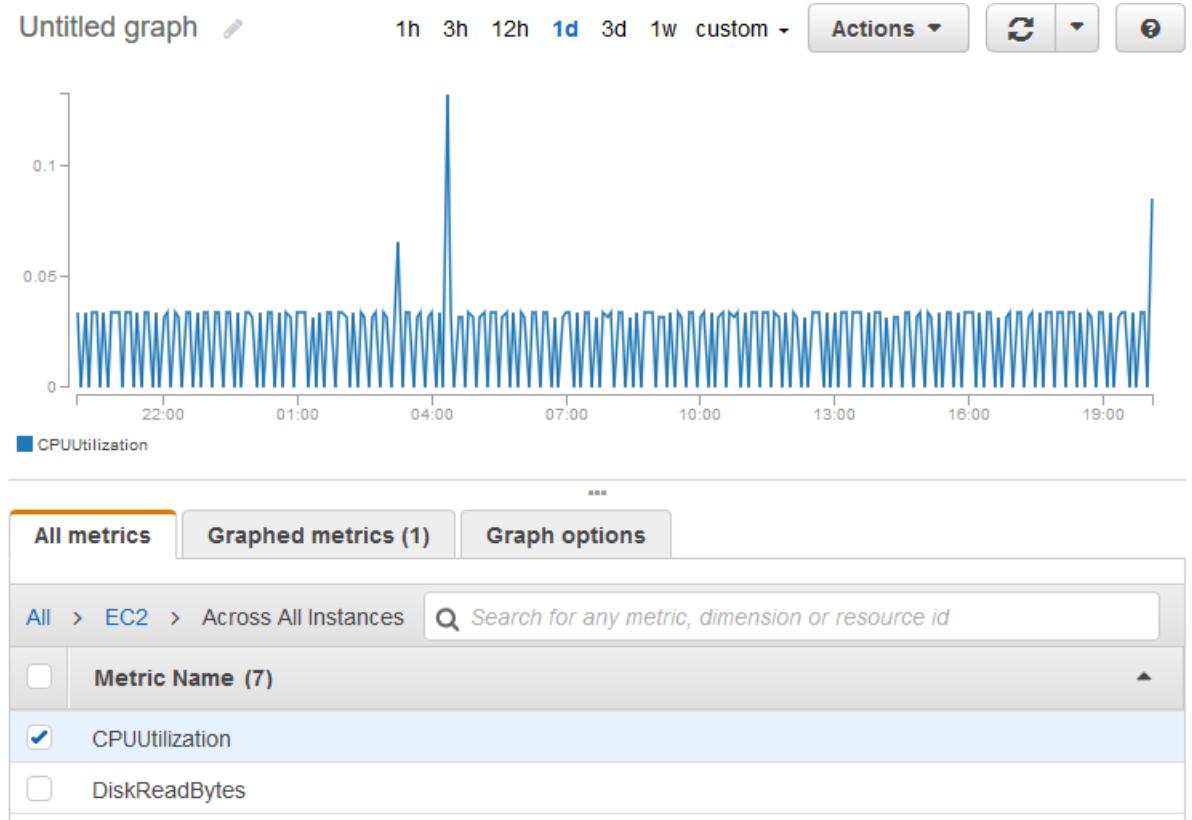
Important

This technique for retrieving all dimensions across an AWS namespace does not work for custom namespaces that you publish to Amazon CloudWatch. With custom namespaces, you must specify the complete set of dimensions that are associated with any given data point to retrieve statistics that include the data point.

To display average CPU utilization across your instances (console)

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** namespace and then choose **Across All Instances**.

4. Choose the row that contains **CPUUtilization**, which displays a graph for the metric for all your EC2 instances. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.



5. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

To get average CPU utilization across your instances (AWS CLI)

Use the [get-metric-statistics](#) command as follows to get the average of the **CPUUtilization** metric across your instances.

```
aws cloudwatch get-metric-statistics \
--namespace AWS/EC2 \
--metric-name CPUUtilization \
--period 3600 --statistics "Average" "SampleCount" \
--start-time 2016-10-11T23:18:00 \
--end-time 2016-10-12T23:18:00
```

The following is example output:

```
{
  "Datapoints": [
    {
      "SampleCount": 238.0,
      "Timestamp": "2016-10-12T07:18:00Z",
      "Average": 0.038235294117647062,
      "Unit": "Percent"
    },
    {
      "SampleCount": 238.0,
      "Timestamp": "2016-10-12T07:18:00Z",
      "Average": 0.038235294117647062,
      "Unit": "Percent"
    }
  ]
}
```

```
"SampleCount": 240.0,  
"Timestamp": "2016-10-12T09:18:00Z",  
"Average": 0.1667083333333332,  
"Unit": "Percent"  
,  
{  
    "SampleCount": 238.0,  
    "Timestamp": "2016-10-11T23:18:00Z",  
    "Average": 0.041596638655462197,  
    "Unit": "Percent"  
,  
    ...  
],  
"Label": "CPUUtilization"  
}
```

Aggregate statistics by Auto Scaling group

You can aggregate statistics for the EC2 instances in an Auto Scaling group. Note that Amazon CloudWatch cannot aggregate data across regions. Metrics are completely separate between regions.

This example shows you how to retrieve the total bytes written to disk for one Auto Scaling group. The total is computed for one-minute periods for a 24-hour interval across all EC2 instances in the specified Auto Scaling group.

To display DiskWriteBytes for the instances in an Auto Scaling group (console)

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** namespace and then choose **By Auto Scaling Group**.
4. Choose the row for the **DiskWriteBytes** metric and the specific Auto Scaling group, which displays a graph for the metric for the instances in the Auto Scaling group. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.
5. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

To display DiskWriteBytes for the instances in an Auto Scaling group (AWS CLI)

Use the [get-metric-statistics](#) command as follows.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name DiskWriteBytes --  
period 360 \  
--statistics "Sum" "SampleCount" --dimensions Name=AutoScalingGroupName,Value=my-asg --  
start-time 2016-10-16T23:18:00 --end-time 2016-10-18T23:18:00
```

The following is example output:

```
{  
    "Datapoints": [  
        {  
            "SampleCount": 18.0,  
            "Timestamp": "2016-10-19T21:36:00Z",  
            "Sum": 0.0,  
            "Unit": "Bytes"  
        },  
        {  
            "SampleCount": 5.0,
```

```
        "Timestamp": "2016-10-19T21:42:00Z",
        "Sum": 0.0,
        "Unit": "Bytes"
    },
    "Label": "DiskWriteBytes"
}
```

Aggregate statistics by AMI

You can aggregate statistics for your instances that have detailed monitoring enabled. Instances that use basic monitoring are not included. Note that Amazon CloudWatch cannot aggregate data across regions. Metrics are completely separate between regions.

Before you can get statistics aggregated across instances, you must enable detailed monitoring (at an additional charge), which provides data in 1-minute periods. For more information, see [Enable or turn off detailed monitoring for your instances \(p. 715\)](#).

This example shows you how to determine average CPU utilization for all instances that use a specific Amazon Machine Image (AMI). The average is over 60-second time intervals for a one-day period.

To display the average CPU utilization by AMI (console)

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** namespace and then choose **By Image (AMI) Id**.
4. Choose the row for the **CPUUtilization** metric and the specific AMI, which displays a graph for the metric for the specified AMI. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.
5. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

To get the average CPU utilization for an image ID (AWS CLI)

Use the `get-metric-statistics` command as follows.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization --
period 3600 \
--statistics Average --dimensions Name=ImageId,Value=ami-3c47a355 --start-
time 2016-10-10T00:00:00 --end-time 2016-10-11T00:00:00
```

The following is example output. Each value represents an average CPU utilization percentage for the EC2 instances running the specified AMI.

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-10T07:00:00Z",
            "Average": 0.04100000000000009,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T14:00:00Z",
            "Average": 0.079579831932773085,
            "Unit": "Percent"
        }
    ]
}
```

```
{  
    "Timestamp": "2016-10-10T06:00:00Z",  
    "Average": 0.03600000000000011,  
    "Unit": "Percent"  
,  
    ...  
],  
"Label": "CPUUtilization"  
}
```

Graph metrics for your instances

After you launch an instance, you can open the Amazon EC2 console and view the monitoring graphs for an instance on the **Monitoring** tab. Each graph is based on one of the available Amazon EC2 metrics.

The following graphs are available:

- Average CPU Utilization (Percent)
- Average Disk Reads (Bytes)
- Average Disk Writes (Bytes)
- Maximum Network In (Bytes)
- Maximum Network Out (Bytes)
- Summary Disk Read Operations (Count)
- Summary Disk Write Operations (Count)
- Summary Status (Any)
- Summary Status Instance (Count)
- Summary Status System (Count)

For more information about the metrics and the data they provide to the graphs, see [List the available CloudWatch metrics for your instances \(p. 717\)](#).

Graph Metrics Using the CloudWatch Console

You can also use the CloudWatch console to graph metric data generated by Amazon EC2 and other AWS services. For more information, see [Graph Metrics](#) in the *Amazon CloudWatch User Guide*.

Create a CloudWatch alarm for an instance

You can create a CloudWatch alarm that monitors CloudWatch metrics for one of your instances. CloudWatch will automatically send you a notification when the metric reaches a threshold you specify. You can create a CloudWatch alarm using the Amazon EC2 console, or using the more advanced options provided by the CloudWatch console.

To create an alarm using the CloudWatch console

For examples, see [Creating Amazon CloudWatch Alarms](#) in the *Amazon CloudWatch User Guide*.

New console

To create an alarm using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, Monitoring, Manage CloudWatch alarms**.
4. On the **Manage CloudWatch alarms** detail page, under **Add or edit alarm**, select **Create a new alarm**.
5. For **Alarm notification**, choose whether to turn the toggle on or off to configure Amazon Simple Notification Service (Amazon SNS) notifications. Enter an existing Amazon SNS topic or enter a name to create a new topic.
6. For **Alarm action**, choose whether to turn the toggle on or off to specify an action to take when the alarm is triggered. Select an action from the dropdown.
7. For **Alarm thresholds**, select the metric and criteria for the alarm. For example, you can leave the default settings for **Group samples by (Average)** and **Type of data to sample (CPU utilization)**. For **Alarm when**, choose **>=** and enter **0 . 80**. For **Consecutive period**, enter **1**. For **Period**, select **5 minutes**.
8. (Optional) For **Sample metric data**, choose **Add to dashboard**.
9. Choose **Create**.

Old console

To create an alarm using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. On the **Monitoring** tab located at the bottom of the page, choose **Create Alarm**. Or, from the **Actions** dropdown, choose **CloudWatch Monitoring, Add/Edit Alarm**.
5. In the **Create Alarm** dialog box, do the following:
 - a. Choose **create topic**. For **Send a notification to**, enter a name for the SNS topic. For **With these recipients**, enter one or more email addresses to receive notification.
 - b. Specify the metric and the criteria for the policy. For example, you can leave the default settings for **Whenever** (Average of CPU Utilization). For **Is**, choose **>=** and enter 80 percent. For **For at least**, enter 1 consecutive period of 5 Minutes.
 - c. Choose **Create Alarm**.

Create Alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define. To edit an alarm, first choose whom to notify and then define when the notification should be sent.

Send a notification to: my-topic [cancel](#)

With these recipients: me@mycompany.com

Take the action: Recover this instance [i](#)
 Stop this instance [i](#)
 Terminate this instance [i](#)
 Reboot this instance [i](#)

Whenever: Average [▼](#) of CPU Utilization [▼](#)

Is: [>=](#) Percent

For at least: consecutive period(s) of [5 Minutes](#) [▼](#)

Name of alarm: CPU-Utilization

[Cancel](#) [Create Alarm](#)

Create alarms that stop, terminate, reboot, or recover an instance

Using Amazon CloudWatch alarm actions, you can create alarms that automatically stop, terminate, reboot, or recover your instances. You can use the stop or terminate actions to help you save money when you no longer need an instance to be running. You can use the reboot and recover actions to automatically reboot those instances or recover them onto new hardware if a system impairment occurs.

The `AWSLambdaRoleForCloudWatchEvents` service-linked role enables AWS to perform alarm actions on your behalf. The first time you create an alarm in the AWS Management Console, the IAM CLI, or the IAM API, CloudWatch creates the service-linked role for you.

There are a number of scenarios in which you might want to automatically stop or terminate your instance. For example, you might have instances dedicated to batch payroll processing jobs or scientific computing tasks that run for a period of time and then complete their work. Rather than letting those instances sit idle (and accrue charges), you can stop or terminate them, which can help you to save money. The main difference between using the stop and the terminate alarm actions is that you can easily restart a stopped instance if you need to run it again later, and you can keep the same instance ID and root volume. However, you cannot restart a terminated instance. Instead, you must launch a new instance.

You can add the stop, terminate, reboot, or recover actions to any alarm that is set on an Amazon EC2 per-instance metric, including basic and detailed monitoring metrics provided by Amazon CloudWatch

(in the AWS/EC2 namespace), as well as any custom metrics that include the `InstanceId` dimension, as long as its value refers to a valid running Amazon EC2 instance.

Console Support

You can create alarms using the Amazon EC2 console or the CloudWatch console. The procedures in this documentation use the Amazon EC2 console. For procedures that use the CloudWatch console, see [Create Alarms That Stop, Terminate, Reboot, or Recover an Instance](#) in the *Amazon CloudWatch User Guide*.

Permissions

If you are an AWS Identity and Access Management (IAM) user, you must have the following permissions to create or modify an alarm:

- `iam:CreateServiceLinkedRole`, `iam:GetPolicy`, `iam:GetPolicyVersion`, and `iam:GetRole`
 - For all alarms with Amazon EC2 actions
- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` – For all alarms on Amazon EC2 instance status metrics
- `ec2:StopInstances` – For alarms with stop actions
- `ec2:TerminateInstances` – For alarms with terminate actions
- No specific permissions are needed for alarms with recover actions.

If you have read/write permissions for Amazon CloudWatch but not for Amazon EC2, you can still create an alarm but the stop or terminate actions won't be performed on the Amazon EC2 instance. However, if you are later granted permission to use the associated Amazon EC2 APIs, the alarm actions you created earlier are performed. For more information about IAM permissions, see [Policies and Permissions](#) in the *IAM User Guide*.

Contents

- [Adding stop actions to Amazon CloudWatch alarms \(p. 739\)](#)
- [Adding terminate actions to Amazon CloudWatch alarms \(p. 740\)](#)
- [Adding reboot actions to Amazon CloudWatch alarms \(p. 741\)](#)
- [Adding recover actions to Amazon CloudWatch alarms \(p. 742\)](#)
- [Using the Amazon CloudWatch console to view alarm and action history \(p. 743\)](#)
- [Amazon CloudWatch alarm action scenarios \(p. 743\)](#)

Adding stop actions to Amazon CloudWatch alarms

You can create an alarm that stops an Amazon EC2 instance when a certain threshold has been met. For example, you may run development or test instances and occasionally forget to shut them off. You can create an alarm that is triggered when the average CPU utilization percentage has been lower than 10 percent for 24 hours, signaling that it is idle and no longer in use. You can adjust the threshold, duration, and period to suit your needs, plus you can add an Amazon Simple Notification Service (Amazon SNS) notification so that you receive an email when the alarm is triggered.

Instances that use an Amazon EBS volume as the root device can be stopped or terminated, whereas instances that use the instance store as the root device can only be terminated.

To create an alarm to stop an idle instance (Amazon EC2 console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance. On the **Monitoring** tab, choose **Create Alarm**.

4. In the **Create Alarm** dialog box, do the following:

- a. To receive an email when the alarm is triggered, for **Send a notification to**, choose an existing Amazon SNS topic, or choose **create topic** to create a new one.

To create a new topic, for **Send a notification to**, enter a name for the topic, and then for **With these recipients**, enter the email addresses of the recipients (separated by commas). After you create the alarm, you will receive a subscription confirmation email that you must accept before you can get notifications for this topic.

- b. Choose **Take the action, Stop this instance**.
 - c. For **Whenever**, choose the statistic you want to use and then choose the metric. In this example, choose **Average** and **CPU Utilization**.
 - d. For **Is**, specify the metric threshold. In this example, enter **10** percent.
 - e. For **For at least**, specify the evaluation period for the alarm. In this example, enter **24** consecutive period(s) of **1 Hour**.
 - f. To change the name of the alarm, for **Name of alarm**, enter a new name. Alarm names must contain only ASCII characters.

If you don't enter a name for the alarm, Amazon CloudWatch automatically creates one for you.

Note

You can adjust the alarm configuration based on your own requirements before creating the alarm, or you can edit them later. This includes the metric, threshold, duration, action, and notification settings. However, after you create an alarm, you cannot edit its name later.

- g. Choose **Create Alarm**.

Adding terminate actions to Amazon CloudWatch alarms

You can create an alarm that terminates an EC2 instance automatically when a certain threshold has been met (as long as termination protection is not enabled for the instance). For example, you might want to terminate an instance when it has completed its work, and you don't need the instance again. If you might want to use the instance later, you should stop the instance instead of terminating it. For information on enabling and disabling termination protection for an instance, see [Enabling termination protection \(p. 609\)](#).

To create an alarm to terminate an idle instance (Amazon EC2 console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 2. In the navigation pane, choose **Instances**.
 3. Select the instance. On the **Monitoring** tab, choose **Create Alarm**.
 4. In the **Create Alarm** dialog box, do the following:
 - a. To receive an email when the alarm is triggered, for **Send a notification to**, choose an existing Amazon SNS topic, or choose **create topic** to create a new one.
- To create a new topic, for **Send a notification to**, enter a name for the topic, and then for **With these recipients**, enter the email addresses of the recipients (separated by commas). After you create the alarm, you will receive a subscription confirmation email that you must accept before you can get notifications for this topic.
- b. Choose **Take the action, Terminate this instance**.
 - c. For **Whenever**, choose a statistic and then choose the metric. In this example, choose **Average** and **CPU Utilization**.
 - d. For **Is**, specify the metric threshold. In this example, enter **10** percent.

- e. For **For at least**, specify the evaluation period for the alarm. In this example, enter **24** consecutive period(s) of **1 Hour**.
- f. To change the name of the alarm, for **Name of alarm**, enter a new name. Alarm names must contain only ASCII characters.

If you don't enter a name for the alarm, Amazon CloudWatch automatically creates one for you.

Note

You can adjust the alarm configuration based on your own requirements before creating the alarm, or you can edit them later. This includes the metric, threshold, duration, action, and notification settings. However, after you create an alarm, you cannot edit its name later.

- g. Choose **Create Alarm**.

Adding reboot actions to Amazon CloudWatch alarms

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically reboots the instance. The reboot alarm action is recommended for Instance Health Check failures (as opposed to the recover alarm action, which is suited for System Health Check failures). An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance. When you reboot an instance, it remains on the same physical host, so your instance keeps its public DNS name, private IP address, and any data on its instance store volumes.

Rebooting an instance doesn't start a new instance billing period (with a minimum one-minute charge), unlike stopping and restarting your instance. For more information, see [Reboot Your Instance](#) in the *Amazon EC2 User Guide for Linux Instances*.

Important

To avoid a race condition between the reboot and recover actions, avoid setting the same number of evaluation periods for a reboot alarm and a recover alarm. We recommend that you set reboot alarms to three evaluation periods of one minute each. For more information, see [Evaluating an Alarm](#) in the *Amazon CloudWatch User Guide*.

To create an alarm to reboot an instance (Amazon EC2 console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance. On the **Monitoring** tab, choose **Create Alarm**.
4. In the **Create Alarm** dialog box, do the following:
 - a. To receive an email when the alarm is triggered, for **Send a notification to**, choose an existing Amazon SNS topic, or choose **create topic** to create a new one.

To create a new topic, for **Send a notification to**, enter a name for the topic, and for **With these recipients**, enter the email addresses of the recipients (separated by commas). After you create the alarm, you will receive a subscription confirmation email that you must accept before you can get notifications for this topic.
 - b. Select **Take the action, Reboot this instance**.
 - c. For **Whenever**, choose **Status Check Failed (Instance)**.
 - d. For **For at least**, specify the evaluation period for the alarm. In this example, enter **3** consecutive period(s) of **1 Minute**.
 - e. To change the name of the alarm, for **Name of alarm**, enter a new name. Alarm names must contain only ASCII characters.

If you don't enter a name for the alarm, Amazon CloudWatch automatically creates one for you.
- f. Choose **Create Alarm**.

Adding recover actions to Amazon CloudWatch alarms

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance. If the instance becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair, you can automatically recover the instance. Terminated instances cannot be recovered. A recovered instance is identical to the original instance, including the instance ID, private IP addresses, Elastic IP addresses, and all instance metadata.

CloudWatch prevents you from adding a recovery action to an alarm that is on an instance which does not support recovery actions.

When the `StatusCheckFailed_System` alarm is triggered, and the recover action is initiated, you are notified by the Amazon SNS topic that you chose when you created the alarm and associated the recover action. During instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is lost. When the process is complete, information is published to the SNS topic you've configured for the alarm. Anyone who is subscribed to this SNS topic receives an email notification that includes the status of the recovery attempt and any further instructions. You notice an instance reboot on the recovered instance.

The recover action can be used only with `StatusCheckFailed_System`, not with `StatusCheckFailed_Instance`.

The following problems can cause system status checks to fail:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host that impact network reachability

The recover action is supported only on instances with the following characteristics:

- Use one of the following instance types: A1, C3, C4, C5, C5a, C5n, C6g, Inf1, M3, M4, M5, M5a, M5n, M6g, P3, R3, R4, R5, R5a, R5n, R6g, T2, T3, T3a, T4g, X1, or X1e
- Use default or dedicated instance tenancy
- Use EBS volumes only (do not configure instance store volumes). For more information, see ['Recover this instance' is disabled](#).

If your instance has a public IP address, it retains the public IP address after recovery.

Important

To avoid a race condition between the reboot and recover actions, avoid setting the same number of evaluation periods for a reboot alarm and a recover alarm. We recommend that you set recover alarms to two evaluation periods of one minute each. For more information, see [Evaluating an Alarm](#) in the *Amazon CloudWatch User Guide*.

To create an alarm to recover an instance (Amazon EC2 console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance. On the **Monitoring** tab, choose **Create Alarm**.
4. In the **Create Alarm** dialog box, do the following:
 - a. To receive an email when the alarm is triggered, for **Send a notification to**, choose an existing Amazon SNS topic, or choose **create topic** to create a new one.

To create a new topic, for **Send a notification to**, enter a name for the topic, and for **With these recipients**, enter the email addresses of the recipients (separated by commas). After you create the alarm, you will receive a subscription confirmation email that you must accept before you can get email for this topic.

Note

- Users must subscribe to the specified SNS topic to receive email notifications when the alarm is triggered.
 - The AWS account root user always receives email notifications when automatic instance recovery actions occur, even if an SNS topic is not specified.
 - The AWS account root user always receives email notifications when automatic instance recovery actions occur, even if it is not subscribed to the specified SNS topic.
- b. Select **Take the action, Recover this instance**.
 - c. For **Whenever**, choose **Status Check Failed (System)**.
 - d. For **For at least**, specify the evaluation period for the alarm. In this example, enter **2 consecutive period(s) of 1 Minute**.
 - e. To change the name of the alarm, for **Name of alarm**, enter a new name. Alarm names must contain only ASCII characters.

If you don't enter a name for the alarm, Amazon CloudWatch automatically creates one for you.
 - f. Choose **Create Alarm**.

Using the Amazon CloudWatch console to view alarm and action history

You can view alarm and action history in the Amazon CloudWatch console. Amazon CloudWatch keeps the last two weeks' worth of alarm and action history.

To view the history of triggered alarms and actions (CloudWatch console)

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Alarms**.
3. Select an alarm.
4. The **Details** tab shows the most recent state transition along with the time and metric values.
5. Choose the **History** tab to view the most recent history entries.

Amazon CloudWatch alarm action scenarios

You can use the Amazon EC2 console to create alarm actions that stop or terminate an Amazon EC2 instance when certain conditions are met. In the following screen capture of the console page where you set the alarm actions, we've numbered the settings. We've also numbered the settings in the scenarios that follow, to help you create the appropriate actions.

Create Alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you specify. To edit an alarm, first choose whom to notify and then define when the notification should be sent.

① Send a notification to: [create topic](#)

② Take the action: Recover this instance [i](#)
 Stop this instance [i](#)
 Terminate this instance [i](#)
 Reboot this instance [i](#)

Whenever: of
Is: Percent

For at least: consecutive period(s) of

Name of alarm:

Scenario 1: Stop idle development and test instances

Create an alarm that stops an instance used for software development or testing when it has been idle for at least an hour.

Setting	Value
1	Stop
2	Maximum
3	CPUUtilization
4	<=
5	10%
6	60 minutes
7	1

Scenario 2: Stop idle instances

Create an alarm that stops an instance and sends an email when the instance has been idle for 24 hours.

Setting	Value
1	Stop and email
2	Average
3	CPUUtilization
4	<=
5	5%
6	60 minutes
7	24

Scenario 3: Send email about web servers with unusually high traffic

Create an alarm that sends email when an instance exceeds 10 GB of outbound network traffic per day.

Setting	Value
1	Email
2	Sum
3	NetworkOut
4	>
5	10 GB
6	1 day
7	1

Scenario 4: Stop web servers with unusually high traffic

Create an alarm that stops an instance and send a text message (SMS) if outbound traffic exceeds 1 GB per hour.

Setting	Value
1	Stop and send SMS
2	Sum
3	NetworkOut
4	>
5	1 GB

Setting	Value
6	1 hour
7	1

Scenario 5: Stop an instance experiencing a memory leak

Create an alarm that stops an instance when memory utilization reaches or exceeds 90%, so that application logs can be retrieved for troubleshooting.

Note

The MemoryUtilization metric is a custom metric. In order to use the MemoryUtilization metric, you must install the Perl scripts for Linux instances. For more information, see [Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances](#).

Setting	Value
1	Stop
2	Maximum
3	MemoryUtilization
4	>=
5	90%
6	1 minute
7	1

Scenario 6: Stop an impaired instance

Create an alarm that stops an instance that fails three consecutive status checks (performed at 5-minute intervals).

Setting	Value
1	Stop
2	Average
3	StatusCheckFailed_System
4	>=
5	1
6	15 minutes
7	1

Scenario 7: Terminate instances when batch processing jobs are complete

Create an alarm that terminates an instance that runs batch jobs when it is no longer sending results data.

Setting	Value
1	Terminate
2	Maximum
3	NetworkOut
4	<=
5	100,000 bytes
6	5 minutes
7	1

Automating Amazon EC2 with CloudWatch Events

Amazon CloudWatch Events enables you to automate your AWS services and respond automatically to system events such as application availability issues or resource changes. Events from AWS services are delivered to CloudWatch Events in near real time. You can write simple rules to indicate which events are of interest to you, and the automated actions to take when an event matches a rule. The actions that can be automatically triggered include the following:

- Invoking an AWS Lambda function
- Invoking Amazon EC2 Run Command
- Relaying the event to Amazon Kinesis Data Streams
- Activating an AWS Step Functions state machine
- Notifying an Amazon SNS topic or an Amazon SQS queue

Some examples of using CloudWatch Events with Amazon EC2 include:

- Activating a Lambda function whenever a new Amazon EC2 instance starts.
- Notifying an Amazon SNS topic when an Amazon EBS volume is created or modified.
- Sending a command to one or more Amazon EC2 instances using Amazon EC2 Run Command whenever a certain event in another AWS service occurs.

For more information, see the [Amazon CloudWatch Events User Guide](#).

Monitoring memory and disk metrics for Amazon EC2 Linux instances

You can use Amazon CloudWatch to collect metrics and logs from the operating systems for your EC2 instances.

Important

The monitoring scripts are deprecated. We recommend that you use the CloudWatch agent to collect metrics and logs. We provide this information about the monitoring scripts for customers who are still migrating from the deprecated monitoring scripts to the CloudWatch agent.

Collecting metrics using the CloudWatch agent

You can use the CloudWatch agent to collect both system metrics and log files from Amazon EC2 instances and on-premises servers. The agent supports both Windows Server and Linux, and enables you to select the metrics to be collected, including sub-resource metrics such as per-CPU core. We recommend that you use the agent to collect metrics and logs instead of using the monitoring scripts. For more information, see [Collect Metrics from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent](#) in the *Amazon CloudWatch User Guide*.

Deprecated: Collecting metrics using the CloudWatch monitoring scripts

Important

We provide information about the monitoring scripts for customers who have not yet migrated from the deprecated monitoring scripts to the CloudWatch agent.

The monitoring scripts demonstrate how to produce and consume custom metrics for Amazon CloudWatch. These sample Perl scripts comprise a fully functional example that reports memory, swap, and disk space utilization metrics for a Linux instance.

Standard Amazon CloudWatch usage charges for custom metrics apply to your use of these scripts. For more information, see the [Amazon CloudWatch](#) pricing page.

Contents

- [Supported systems \(p. 748\)](#)
- [Required permissions \(p. 748\)](#)
- [Install required packages \(p. 749\)](#)
- [Install monitoring scripts \(p. 750\)](#)
- [mon-put-instance-data.pl \(p. 751\)](#)
- [mon-get-instance-stats.pl \(p. 754\)](#)
- [Viewing your custom metrics in the console \(p. 755\)](#)
- [Troubleshooting \(p. 755\)](#)

Supported systems

The monitoring scripts were tested on instances using the following systems. Using the monitoring scripts on any other operating system is unsupported.

- Amazon Linux 2
- Amazon Linux AMI 2014.09.2 and later
- Red Hat Enterprise Linux 6.9 and 7.4
- SUSE Linux Enterprise Server 12
- Ubuntu Server 14.04 and 16.04

Required permissions

Ensure that the scripts have permission to call the following actions by associating an IAM role with your instance:

- `cloudwatch:PutMetricData`

-
- cloudwatch:GetMetricStatistics
 - cloudwatch>ListMetrics
 - ec2:DescribeTags

For more information, see [Working with IAM roles \(p. 976\)](#).

Install required packages

With some versions of Linux, you must install additional Perl modules before you can use the monitoring scripts.

To install the required packages on Amazon Linux 2 and Amazon Linux AMI

1. Log on to your instance. For more information, see [Connect to your Linux instance \(p. 563\)](#).
2. At a command prompt, install packages as follows:

```
sudo yum install -y perl-Switch perl-Datetime perl-Sys-Syslog perl-LWP-Protocol-https  
perl-Digest-SHA.x86_64
```

To install the required packages on Ubuntu

1. Log on to your instance. For more information, see [Connect to your Linux instance \(p. 563\)](#).
2. At a command prompt, install packages as follows:

```
sudo apt-get update  
sudo apt-get install unzip  
sudo apt-get install libwww-perl libdatetime-perl
```

To install the required packages on Red Hat Enterprise Linux 7

1. Log on to your instance. For more information, see [Connect to your Linux instance \(p. 563\)](#).
2. At a command prompt, install packages as follows:

```
sudo yum install perl-Switch perl-Datetime perl-Sys-Syslog perl-LWP-Protocol-https  
perl-Digest-SHA --enablerepo="rhui-REGION-rhel-server-optional" -y  
sudo yum install zip unzip
```

To install the required packages on Red Hat Enterprise Linux 6.9

1. Log on to your instance. For more information, see [Connect to your Linux instance \(p. 563\)](#).
2. At a command prompt, install packages as follows:

```
sudo yum install perl-Datetime perl-CPAN perl-Net-SSLeay perl-IO-Socket-SSL perl-  
Digest-SHA gcc -y  
sudo yum install zip unzip
```

3. Run CPAN as an elevated user:

```
sudo cpan
```

Press ENTER through the prompts until you see the following prompt:

```
cpan[1]>
```

4. At the CPAN prompt, run each of the below commands: run one command and it installs, and when you return to the CPAN prompt, run the next command. Press ENTER like before when prompted to continue through the process:

```
cpan[1]> install YAML
cpan[2]> install LWP::Protocol::https
cpan[3]> install Sys::Syslog
cpan[4]> install Switch
```

To install the required packages on SUSE

1. Log on to your instance. For more information, see [Connect to your Linux instance \(p. 563\)](#).
2. On servers running SUSE Linux Enterprise Server 12, you might need to download the perl-Switch package. You can download and install this package using the following commands:

```
wget http://download.opensuse.org/repositories/devel:/languages:/perl/SLE_12_SP3/
noarch/perl-Switch-2.17-32.1.noarch.rpm
sudo rpm -i perl-Switch-2.17-32.1.noarch.rpm
```

3. Install the required packages as follows:

```
sudo zypper install perl-Switch perl-DateTime
sudo zypper install -y "perl(LWP::Protocol::https)"
```

Install monitoring scripts

The following steps show you how to download, uncompress, and configure the CloudWatch Monitoring Scripts on an EC2 Linux instance.

To download, install, and configure the monitoring scripts

1. At a command prompt, move to a folder where you want to store the monitoring scripts and run the following command to download them:

```
curl https://aws-cloudwatch.s3.amazonaws.com/downloads/
CloudWatchMonitoringScripts-1.2.2.zip -O
```

2. Run the following commands to install the monitoring scripts you downloaded:

```
unzip CloudWatchMonitoringScripts-1.2.2.zip && \
rm CloudWatchMonitoringScripts-1.2.2.zip && \
cd aws-scripts-mon
```

The package for the monitoring scripts contains the following files:

- **CloudWatchClient.pm** – Shared Perl module that simplifies calling Amazon CloudWatch from other scripts.
- **mon-put-instance-data.pl** – Collects system metrics on an Amazon EC2 instance (memory, swap, disk space utilization) and sends them to Amazon CloudWatch.

-
- **mon-get-instance-stats.pl** – Queries Amazon CloudWatch and displays the most recent utilization statistics for the EC2 instance on which this script is executed.
 - **awscreds.template** – File template for AWS credentials that stores your access key ID and secret access key.
 - **LICENSE.txt** – Text file containing the Apache 2.0 license.
 - **NOTICE.txt** – Copyright notice.

mon-put-instance-data.pl

This script collects memory, swap, and disk space utilization data on the current system. It then makes a remote call to Amazon CloudWatch to report the collected data as custom metrics.

Options

Name	Description
--mem-util	Collects and sends the MemoryUtilization metrics in percentages. This metric counts memory allocated by applications and the operating system as used, and also includes cache and buffer memory as used if you specify the --mem-used-incl-cache-buff option.
--mem-used	Collects and sends the MemoryUsed metrics, reported in megabytes. This metric counts memory allocated by applications and the operating system as used, and also includes cache and buffer memory as used if you specify the --mem-used-incl-cache-buff option.
--mem-used-incl-cache-buff	If you include this option, memory currently used for cache and buffers is counted as "used" when the metrics are reported for --mem-util, --mem-used, and --mem-avail.
--mem-avail	Collects and sends the MemoryAvailable metrics, reported in megabytes. This metric counts memory allocated by applications and the operating system as used, and also includes cache and buffer memory as used if you specify the --mem-used-incl-cache-buff option.
--swap-util	Collects and sends SwapUtilization metrics, reported in percentages.
--swap-used	Collects and sends SwapUsed metrics, reported in megabytes.
--disk-path=PATH	Selects the disk on which to report. PATH can specify a mount point or any file located on a mount point for the filesystem that needs to be reported. For selecting multiple disks, specify a --disk-path=PATH for each one of them. To select a disk for the filesystems mounted on / and /home, use the following parameters: --disk-path=/ --disk-path=/home
--disk-space-util	Collects and sends the DiskSpaceUtilization metric for the selected disks. The metric is reported in percentages.

Amazon Elastic Compute Cloud
 User Guide for Linux Instances
 Deprecated: Collecting metrics using
 the CloudWatch monitoring scripts

Name	Description
	Note that the disk utilization metrics calculated by this script differ from the values calculated by the <code>df -k -l</code> command. If you find the values from <code>df -k -l</code> more useful, you can change the calculations in the script.
--disk-space-used	<p>Collects and sends the <code>DiskSpaceUsed</code> metric for the selected disks. The metric is reported by default in gigabytes.</p> <p>Due to reserved disk space in Linux operating systems, disk space used and disk space available might not accurately add up to the amount of total disk space.</p>
--disk-space-avail	<p>Collects and sends the <code>DiskSpaceAvailable</code> metric for the selected disks. The metric is reported in gigabytes.</p> <p>Due to reserved disk space in the Linux operating systems, disk space used and disk space available might not accurately add up to the amount of total disk space.</p>
--memory-units=UNITS	Specifies units in which to report memory usage. If not specified, memory is reported in megabytes. UNITS may be one of the following: bytes, kilobytes, megabytes, gigabytes.
--disk-space-units=UNITS	Specifies units in which to report disk space usage. If not specified, disk space is reported in gigabytes. UNITS may be one of the following: bytes, kilobytes, megabytes, gigabytes.
--aws-credential-file=PATH	<p>Provides the location of the file containing AWS credentials.</p> <p>This parameter cannot be used with the <code>--aws-access-key-id</code> and <code>--aws-secret-key</code> parameters.</p>
--aws-access-key-id=VALUE	Specifies the AWS access key ID to use to identify the caller. Must be used together with the <code>--aws-secret-key</code> option. Do not use this option with the <code>--aws-credential-file</code> parameter.
--aws-secret-key=VALUE	Specifies the AWS secret access key to use to sign the request to CloudWatch. Must be used together with the <code>--aws-access-key-id</code> option. Do not use this option with <code>--aws-credential-file</code> parameter.
--aws-iam-role=VALUE	<p>Specifies the IAM role used to provide AWS credentials. The value <code>=VALUE</code> is required. If no credentials are specified, the default IAM role associated with the EC2 instance is applied. Only one IAM role can be used. If no IAM roles are found, or if more than one IAM role is found, the script will return an error.</p> <p>Do not use this option with the <code>--aws-credential-file</code>, <code>--aws-access-key-id</code>, or <code>--aws-secret-key</code> parameters.</p>
--aggregated[=only]	Adds aggregated metrics for instance type, AMI ID, and overall for the Region. The value <code>=only</code> is optional; if specified, the script reports only aggregated metrics.

Name	Description
--auto-scaling[=only]	Adds aggregated metrics for the Auto Scaling group. The value <code>=only</code> is optional; if specified, the script reports only Auto Scaling metrics. The IAM policy associated with the IAM account or role using the scripts need to have permissions to call the EC2 action DescribeTags .
--verify	Performs a test run of the script that collects the metrics, prepares a complete HTTP request, but does not actually call CloudWatch to report the data. This option also checks that credentials are provided. When run in verbose mode, this option outputs the metrics that will be sent to CloudWatch.
--from-cron	Use this option when calling the script from cron. When this option is used, all diagnostic output is suppressed, but error messages are sent to the local system log of the user account.
--verbose	Displays detailed information about what the script is doing.
--help	Displays usage information.
--version	Displays the version number of the script.

Examples

The following examples assume that you provided an IAM role or `awscreds.conf` file. Otherwise, you must provide credentials using the `--aws-access-key-id` and `--aws-secret-key` parameters for these commands.

The following example performs a simple test run without posting data to CloudWatch.

```
./mon-put-instance-data.pl --mem-util --verify --verbose
```

The following example collects all available memory metrics and sends them to CloudWatch, counting cache and buffer memory as used

```
./mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-util --mem-used --mem-avail
```

The following example collects aggregated metrics for an Auto Scaling group and sends them to Amazon CloudWatch without reporting individual instance metrics.

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail --auto-scaling=only
```

The following example collects aggregated metrics for instance type, AMI ID and region, and sends them to Amazon CloudWatch without reporting individual instance metrics

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail --aggregated=only
```

To set a cron schedule for metrics reported to CloudWatch, start editing the crontab using the `crontab -e` command. Add the following command to report memory and disk space utilization to CloudWatch every five minutes:

```
*/5 * * * * ~/aws-scripts-mon/mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-util --disk-space-util --disk-path=/ --from-cron
```

If the script encounters an error, it writes the error message in the system log.

mon-get-instance-stats.pl

This script queries CloudWatch for statistics on memory, swap, and disk space metrics within the time interval provided using the number of most recent hours. This data is provided for the Amazon EC2 instance on which this script is executed.

Options

Name	Description
--recent-hours=N	Specifies the number of recent hours to report on, as represented by N where N is an integer.
--aws-credential-file=PATH	Provides the location of the file containing AWS credentials.
--aws-access-key-id=VALUE	Specifies the AWS access key ID to use to identify the caller. Must be used together with the --aws-secret-key option. Do not use this option with the --aws-credential-file option.
--aws-secret-key=VALUE	Specifies the AWS secret access key to use to sign the request to CloudWatch. Must be used together with the --aws-access-key-id option. Do not use this option with --aws-credential-file option.
--aws-iam-role=VALUE	Specifies the IAM role used to provide AWS credentials. The value =VALUE is required. If no credentials are specified, the default IAM role associated with the EC2 instance is applied. Only one IAM role can be used. If no IAM roles are found, or if more than one IAM role is found, the script will return an error. Do not use this option with the --aws-credential-file, --aws-access-key-id, or --aws-secret-key parameters.
--verify	Performs a test run of the script. This option also checks that credentials are provided.
--verbose	Displays detailed information about what the script is doing.
--help	Displays usage information.
--version	Displays the version number of the script.

Example

To get utilization statistics for the last 12 hours, run the following command:

```
./mon-get-instance-stats.pl --recent-hours=12
```

The following is an example response:

```
Instance metric statistics for the last 12 hours.

CPU Utilization
    Average: 1.06%, Minimum: 0.00%, Maximum: 15.22%

Memory Utilization
```

```
Average: 6.84%, Minimum: 6.82%, Maximum: 6.89%
```

Swap Utilization

```
Average: N/A, Minimum: N/A, Maximum: N/A
```

Disk Space Utilization on /dev/xvda1 mounted as /

```
Average: 9.69%, Minimum: 9.69%, Maximum: 9.69%
```

Viewing your custom metrics in the console

After you successfully run the `mon-put-instance-data.pl` script, you can view your custom metrics in the Amazon CloudWatch console.

To view custom metrics

1. Run `mon-put-instance-data.pl` as described previously.
2. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
3. Choose **View Metrics**.
4. For **Viewing**, your custom metrics posted by the script are displayed with the prefix `System/Linux`.

Troubleshooting

The **CloudWatchClient.pm** module caches instance metadata locally. If you create an AMI from an instance where you have run the monitoring scripts, any instances launched from the AMI within the cache TTL (default: six hours, 24 hours for Auto Scaling groups) emit metrics using the instance ID of the original instance. After the cache TTL time period passes, the script retrieves fresh data and the monitoring scripts use the instance ID of the current instance. To immediately correct this, remove the cached data using the following command:

```
rm /var/tmp/aws-mon/instance-id
```

Logging Amazon EC2 and Amazon EBS API calls with AWS CloudTrail

Amazon EC2 and Amazon EBS are integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon EC2 and Amazon EBS. CloudTrail captures all API calls for Amazon EC2 and Amazon EBS as events, including calls from the console and from code calls to the APIs. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon EC2 and Amazon EBS. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Amazon EC2 and Amazon EBS, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the [AWS CloudTrail User Guide](#).

Amazon EC2 and Amazon EBS information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon EC2 and Amazon EBS, that activity is recorded in a CloudTrail event along with other AWS service events

in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing Events with CloudTrail Event History](#).

For an ongoing record of events in your AWS account, including events for Amazon EC2 and Amazon EBS, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- [Overview for Creating a Trail](#)
- [CloudTrail Supported Services and Integrations](#)
- [Configuring Amazon SNS Notifications for CloudTrail](#)
- [Receiving CloudTrail Log Files from Multiple Regions](#) and [Receiving CloudTrail Log Files from Multiple Accounts](#)

All Amazon EC2 actions, and Amazon EBS management actions, are logged by CloudTrail and are documented in the [Amazon EC2 API Reference](#). For example, calls to the [RunInstances](#), [DescribeInstances](#), or [CreateImage](#) actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the [CloudTrail userIdentity Element](#).

Understanding Amazon EC2 and Amazon EBS log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

The following log file record shows that a user terminated an instance.

```
{  
  "Records": [  
    {  
      "eventVersion": "1.03",  
      "userIdentity": {  
        "type": "Root",  
        "principalId": "123456789012",  
        "arn": "arn:aws:iam::123456789012:root",  
        "accountId": "123456789012",  
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",  
        "userName": "user"  
      },  
      "eventTime": "2016-05-20T08:27:45Z",  
      "eventSource": "ec2.amazonaws.com",  
      "eventName": "TerminateInstances",  
      "awsRegion": "us-west-2",  
      "version": "1"  
    }  
  ]  
}
```

```
"sourceIPAddress":"198.51.100.1",
"userAgent":"aws-cli/1.10.10 Python/2.7.9 Windows/7botocore/1.4.1",
"requestParameters":{
    "instancesSet":{
        "items":[
            {
                "instanceId":"i-1a2b3c4d"
            }
        ]
    },
    "responseElements":{
        "instancesSet":{
            "items":[
                {
                    "instanceId":"i-1a2b3c4d",
                    "currentState":{
                        "code":32,
                        "name":"shutting-down"
                    },
                    "previousState":{
                        "code":16,
                        "name":"running"
                    }
                }
            ]
        },
        "requestID":"be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
        "eventID":"6e12345-2a4e-417c-aa78-7594fEXAMPLE",
        "eventType":"AwsApiCall",
        "recipientAccountId":"123456789012"
    }
}
]
```

Using AWS CloudTrail to audit users that connect via EC2 Instance Connect

Use AWS CloudTrail to audit the users that connect to your instances via EC2 Instance Connect.

To audit SSH activity via EC2 Instance Connect using the AWS CloudTrail console

1. Open the AWS CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Verify that you are in the correct Region.
3. In the navigation pane, choose **Event history**.
4. For **Filter**, choose **Event source, ec2-instance-connect.amazonaws.com**.
5. (Optional) For **Time range**, select a time range.
6. Choose the **Refresh events** icon.
7. The page displays the events that correspond to the **SendSSHPublicKey** API calls. Expand an event using the arrow to view additional details, such as the user name and AWS access key that was used to make the SSH connection, and the source IP address.
8. To display the full event information in JSON format, choose **View event**. The **requestParameters** field contains the destination instance ID, OS user name, and public key that were used to make the SSH connection.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDEFGGNOMOOCB6XYTQEXAMPLE",
        "arn": "arn:aws:iam::1234567890120:user/IAM-friendly-name",
```

```
    "accountId": "123456789012",
    "accessKeyId": "ABCDEFGHIJKLMNO01234567890EXAMPLE",
    "userName": "IAM-friendly-name",
    "sessionContext": {
        "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2018-09-21T21:37:58Z"
        }
    },
    "eventTime": "2018-09-21T21:38:00Z",
    "eventSource": "ec2-instance-connect.amazonaws.com",
    "eventName": "SendSSHPublicKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "123.456.789.012",
    "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
    "requestParameters": {
        "instanceId": "i-0123456789EXAMPLE",
        "osUser": "ec2-user",
        "SSHKey": {
            "publicKey": "ssh-rsa ABCDEFGHIJKLMNOP01234567890EXAMPLE"
        }
    },
    "responseElements": null,
    "requestID": "1a2s3d4f-bde6-11e8-a892-f7ec64543add",
    "eventID": "1a2w3d4r5-a88f-4e28-b3bf-30161f75be34",
    "eventType": "AwsApiCall",
    "recipientAccountId": "0987654321"
}
```

If you have configured your AWS account to collect CloudTrail events in an S3 bucket, you can download and audit the information programmatically. For more information, see [Getting and Viewing Your CloudTrail Log Files](#) in the *AWS CloudTrail User Guide*.

Networking in Amazon EC2

Amazon EC2 provides the following networking features.

Features

- [Amazon EC2 instance IP addressing \(p. 759\)](#)
- [Bring your own IP addresses \(BYOIP\) in Amazon EC2 \(p. 775\)](#)
- [Elastic IP addresses \(p. 781\)](#)
- [Elastic network interfaces \(p. 789\)](#)
- [Enhanced networking on Linux \(p. 813\)](#)
- [Elastic Fabric Adapter \(p. 838\)](#)
- [Placement groups \(p. 869\)](#)
- [Network maximum transmission unit \(MTU\) for your EC2 instance \(p. 881\)](#)
- [Virtual private clouds \(p. 884\)](#)
- [EC2-Classic \(p. 884\)](#)

Amazon EC2 instance IP addressing

Amazon EC2 and Amazon VPC support both the IPv4 and IPv6 addressing protocols. By default, Amazon EC2 and Amazon VPC use the IPv4 addressing protocol; you can't disable this behavior. When you create a VPC, you must specify an IPv4 CIDR block (a range of private IPv4 addresses). You can optionally assign an IPv6 CIDR block to your VPC and subnets, and assign IPv6 addresses from that block to instances in your subnet. IPv6 addresses are reachable over the Internet. For more information about IPv6, see [IP Addressing in Your VPC](#) in the *Amazon VPC User Guide*.

Contents

- [Private IPv4 addresses and internal DNS hostnames \(p. 759\)](#)
- [Public IPv4 addresses and external DNS hostnames \(p. 760\)](#)
- [Elastic IP addresses \(IPv4\) \(p. 761\)](#)
- [Amazon DNS server \(p. 761\)](#)
- [IPv6 addresses \(p. 761\)](#)
- [Working with the IPv4 addresses for your instances \(p. 762\)](#)
- [Working with the IPv6 addresses for your instances \(p. 765\)](#)
- [Multiple IP addresses \(p. 767\)](#)

Private IPv4 addresses and internal DNS hostnames

A private IPv4 address is an IP address that's not reachable over the Internet. You can use private IPv4 addresses for communication between instances in the same VPC. For more information about the standards and specifications of private IPv4 addresses, see [RFC 1918](#). We allocate private IPv4 addresses to instances using DHCP.

Note

You can create a VPC with a publicly routable CIDR block that falls outside of the private IPv4 address ranges specified in RFC 1918. However, for the purposes of this documentation, we refer

to private IPv4 addresses (or 'private IP addresses') as the IP addresses that are within the IPv4 CIDR range of your VPC.

When you launch an instance, we allocate a primary private IPv4 address for the instance. Each instance is also given an internal DNS hostname that resolves to the primary private IPv4 address; for example, `ip-10-251-50-12.ec2.internal`. You can use the internal DNS hostname for communication between instances in the same VPC, but we can't resolve the internal DNS hostname outside of the VPC.

An instance receives a primary private IP address from the IPv4 address range of the subnet. For more information, see [VPC and subnet sizing](#) in the *Amazon VPC User Guide*. If you don't specify a primary private IP address when you launch the instance, we select an available IP address in the subnet's IPv4 range for you. Each instance has a default network interface (`eth0`) that is assigned the primary private IPv4 address. You can also specify additional private IPv4 addresses, known as *secondary private IPv4 addresses*. Unlike primary private IP addresses, secondary private IP addresses can be reassigned from one instance to another. For more information, see [Multiple IP addresses \(p. 767\)](#).

A private IPv4 address, regardless of whether it is a primary or secondary address, remains associated with the network interface when the instance is stopped and started, or hibernated and started, and is released when the instance is terminated.

Public IPv4 addresses and external DNS hostnames

A public IP address is an IPv4 address that's reachable from the Internet. You can use public addresses for communication between your instances and the Internet.

Each instance that receives a public IP address is also given an external DNS hostname; for example, `ec2-203-0-113-25.compute-1.amazonaws.com`. We resolve an external DNS hostname to the public IP address of the instance from outside its VPC, and to the private IPv4 address of the instance from inside its VPC. The public IP address is mapped to the primary private IP address through network address translation (NAT). For more information, see [RFC 1631: The IP Network Address Translator \(NAT\)](#).

When you launch an instance in a default VPC, we assign it a public IP address by default. When you launch an instance into a nondefault VPC, the subnet has an attribute that determines whether instances launched into that subnet receive a public IP address from the public IPv4 address pool. By default, we don't assign a public IP address to instances launched in a nondefault subnet.

You can control whether your instance receives a public IP address as follows:

- Modifying the public IP addressing attribute of your subnet. For more information, see [Modifying the public IPv4 addressing attribute for your subnet](#) in the *Amazon VPC User Guide*.
- Enabling or disabling the public IP addressing feature during launch, which overrides the subnet's public IP addressing attribute. For more information, see [Assigning a public IPv4 address during instance launch \(p. 764\)](#).

A public IP address is assigned to your instance from Amazon's pool of public IPv4 addresses, and is not associated with your AWS account. When a public IP address is disassociated from your instance, it is released back into the public IPv4 address pool, and you cannot reuse it.

You cannot manually associate or disassociate a public IP address from your instance. Instead, in certain cases, we release the public IP address from your instance, or assign it a new one:

- We release your instance's public IP address when it is stopped, hibernated, or terminated. Your stopped or hibernated instance receives a new public IP address when it is started.
- We release your instance's public IP address when you associate an Elastic IP address with it. When you disassociate the Elastic IP address from your instance, it receives a new public IP address.
- If the public IP address of your instance in a VPC has been released, it will not receive a new one if there is more than one network interface attached to your instance.

- If your instance's public IP address is released while it has a secondary private IP address that is associated with an Elastic IP address, the instance does not receive a new public IP address.

If you require a persistent public IP address that can be associated to and from instances as you require, use an Elastic IP address instead.

If you use dynamic DNS to map an existing DNS name to a new instance's public IP address, it might take up to 24 hours for the IP address to propagate through the Internet. As a result, new instances might not receive traffic while terminated instances continue to receive requests. To solve this problem, use an Elastic IP address. You can allocate your own Elastic IP address, and associate it with your instance. For more information, see [Elastic IP addresses \(p. 781\)](#).

If you assign an Elastic IP address to an instance, it receives an IPv4 DNS hostname if DNS hostnames are enabled. For more information, see [Using DNS with your VPC](#) in the *Amazon VPC User Guide*.

Note

Instances that access other instances through their public NAT IP address are charged for regional or Internet data transfer, depending on whether the instances are in the same Region.

Elastic IP addresses (IPv4)

An Elastic IP address is a public IPv4 address that you can allocate to your account. You can associate it to and disassociate it from instances as you require. It's allocated to your account until you choose to release it. For more information about Elastic IP addresses and how to use them, see [Elastic IP addresses \(p. 781\)](#).

We do not support Elastic IP addresses for IPv6.

Amazon DNS server

Amazon provides a DNS server that resolves Amazon-provided IPv4 DNS hostnames to IPv4 addresses. The Amazon DNS server is located at the base of your VPC network range plus two. For more information, see [Amazon DNS server](#) in the *Amazon VPC User Guide*.

IPv6 addresses

You can optionally associate an IPv6 CIDR block with your VPC, and associate IPv6 CIDR blocks with your subnets. The IPv6 CIDR block for your VPC is automatically assigned from Amazon's pool of IPv6 addresses; you cannot choose the range yourself. For more information, see the following topics in the *Amazon VPC User Guide*:

- [VPC and subnet sizing for IPv6](#)
- [Associating an IPv6 CIDR block with your VPC](#)
- [Associating an IPv6 CIDR block with your subnet](#)

IPv6 addresses are globally unique, and therefore reachable over the Internet. Your instance receives an IPv6 address if an IPv6 CIDR block is associated with your VPC and subnet, and if one of the following is true:

- Your subnet is configured to automatically assign an IPv6 address to an instance during launch. For more information, see [Modifying the IPv6 addressing attribute for your subnet](#).
- You assign an IPv6 address to your instance during launch.
- You assign an IPv6 address to the primary network interface of your instance after launch.
- You assign an IPv6 address to a network interface in the same subnet, and attach the network interface to your instance after launch.

When your instance receives an IPv6 address during launch, the address is associated with the primary network interface (eth0) of the instance. You can disassociate the IPv6 address from the network interface. We do not support IPv6 DNS hostnames for your instance.

An IPv6 address persists when you stop and start, or hibernate and start, your instance, and is released when you terminate your instance. You cannot reassign an IPv6 address while it's assigned to another network interface—you must first unassign it.

You can assign additional IPv6 addresses to your instance by assigning them to a network interface attached to your instance. The number of IPv6 addresses you can assign to a network interface and the number of network interfaces you can attach to an instance varies per instance type. For more information, see [IP addresses per network interface per instance type \(p. 791\)](#).

Working with the IPv4 addresses for your instances

You can assign a public IPv4 address to your instance when you launch it. You can view the IPv4 addresses for your in the console through either the **Instances** page or the **Network Interfaces** page.

Contents

- [Viewing the IPv4 addresses \(p. 762\)](#)
- [Assigning a public IPv4 address during instance launch \(p. 764\)](#)

Viewing the IPv4 addresses

You can use the Amazon EC2 console to view the private IPv4 addresses, public IPv4 addresses, and Elastic IP addresses of your instances. You can also determine the public IPv4 and private IPv4 addresses of your instance from within your instance by using instance metadata. For more information, see [Instance metadata and user data \(p. 658\)](#).

The public IPv4 address is displayed as a property of the network interface in the console, but it's mapped to the primary private IPv4 address through NAT. Therefore, if you inspect the properties of your network interface on your instance, for example, through `ifconfig` (Linux) or `ipconfig` (Windows), the public IPv4 address is not displayed. To determine your instance's public IPv4 address from an instance, use instance metadata.

New console

To view the IPv4 addresses for an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select your instance.
3. The following information is available on the **Networking** tab:
 - **Public IPv4 address** — The public IPv4 address. If you associated an Elastic IP address with the instance or the primary network interface, this is the Elastic IP address.
 - **Public IPv4 DNS** — The external DNS hostname.
 - **Private IPv4 addresses** — The private IPv4 address.
 - **Private IPv4 DNS** — The internal DNS hostname.
 - **Secondary private IPv4 addresses** — Any secondary private IPv4 addresses.
 - **Elastic IP addresses** — Any associated Elastic IP addresses.
4. Alternatively, under **Network interfaces** on the **Networking** tab, choose the interface ID for the primary network interface (for example, eni-123abc456def78901). The following information is available:
 - **Private DNS (IPv4)** — The internal DNS hostname.

- **Primary private IPv4 IP** — The primary private IPv4 address.
- **Secondary private IPv4 IPs** — Any secondary private IPv4 addresses.
- **Public DNS** — The external DNS hostname.
- **IPv4 Public IP** — The public IPv4 address. If you associated an Elastic IP address with the instance or the primary network interface, this is the Elastic IP address.
- **Elastic IPs** — Any associated Elastic IP addresses.

Old console

To view the IPv4 addresses for an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select your instance.
3. The following information is available on the **Description** tab:
 - **Private DNS** — The internal DNS hostname.
 - **Private IPs** — The private IPv4 address.
 - **Secondary private IPs** — Any secondary private IPv4 addresses.
 - **Public DNS** — The external DNS hostname.
 - **IPv4 Public IP** — The public IPv4 address. If you associated an Elastic IP address with the instance or the primary network interface, this is the Elastic IP address.
 - **Elastic IPs** — Any associated Elastic IP addresses.
4. Alternatively, you can view the IPv4 addresses for the instance using the primary network interface. Under **Network interfaces** on the **Description** tab, choose **eth0**, and then choose the interface ID (for example, eni-123abc456def78901). The following information is available:
 - **Private DNS (IPv4)** — The internal DNS hostname.
 - **Primary private IPv4 IP** — The primary private IPv4 address.
 - **Secondary private IPv4 IPs** — Any secondary private IPv4 addresses.
 - **Public DNS** — The external DNS hostname.
 - **IPv4 Public IP** — The public IPv4 address. If you associated an Elastic IP address with the instance or the primary network interface, this is the Elastic IP address.
 - **Elastic IPs** — Any associated Elastic IP addresses.

To view the IPv4 addresses for an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-instances \(AWS CLI\)](#)
- [Get-EC2Instance \(AWS Tools for Windows PowerShell\)](#).

To determine your instance's IPv4 addresses using instance metadata

1. Connect to your instance. For more information, see [Connect to your Linux instance \(p. 563\)](#).
2. Use the following command to access the private IP address:

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
```

```
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/local-ipv4
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4
```

3. Use the following command to access the public IP address:

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/public-ipv4
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-ipv4
```

If an Elastic IP address is associated with the instance, the value returned is that of the Elastic IP address.

Assigning a public IPv4 address during instance launch

Each subnet has an attribute that determines whether instances launched into that subnet are assigned a public IP address. By default, nondefault subnets have this attribute set to false, and default subnets have this attribute set to true. When you launch an instance, a public IPv4 addressing feature is also available for you to control whether your instance is assigned a public IPv4 address; you can override the default behavior of the subnet's IP addressing attribute. The public IPv4 address is assigned from Amazon's pool of public IPv4 addresses, and is assigned to the network interface with the device index of eth0. This feature depends on certain conditions at the time you launch your instance.

Considerations

- You can't manually disassociate the public IP address from your instance after launch. Instead, it's automatically released in certain cases, after which you cannot reuse it. For more information, see [Public IPv4 addresses and external DNS hostnames \(p. 760\)](#). If you require a persistent public IP address that you can associate or disassociate at will, assign an Elastic IP address to the instance after launch instead. For more information, see [Elastic IP addresses \(p. 781\)](#).
- You cannot auto-assign a public IP address if you specify more than one network interface. Additionally, you cannot override the subnet setting using the auto-assign public IP feature if you specify an existing network interface for eth0.
- The public IP addressing feature is only available during launch. However, whether you assign a public IP address to your instance during launch or not, you can associate an Elastic IP address with your instance after it's launched. For more information, see [Elastic IP addresses \(p. 781\)](#). You can also modify your subnet's public IPv4 addressing behavior. For more information, see [Modifying the public IPv4 addressing attribute for your subnet](#).

To enable or disable the public IP addressing feature using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.

3. Select an AMI and an instance type, and then choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, for **Network**, select a VPC. The **Auto-assign Public IP** list is displayed. Choose **Enable** or **Disable** to override the default setting for the subnet.
5. Follow the steps on the next pages of the wizard to complete your instance's setup. For more information about the wizard configuration options, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#). On the final **Review Instance Launch** page, review your settings, and then choose **Launch** to choose a key pair and launch your instance.
6. On the **Instances** page, select your new instance and view its public IP address in **IPv4 Public IP** field in the details pane.

To enable or disable the public IP addressing feature using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- Use the `--associate-public-ip-address` or the `--no-associate-public-ip-address` option with the `run-instances` command (AWS CLI)
- Use the `-AssociatePublicIp` parameter with the `New-EC2Instance` command (AWS Tools for Windows PowerShell)

Working with the IPv6 addresses for your instances

You can view the IPv6 addresses assigned to your instance, assign a public IPv6 address to your instance, or unassign an IPv6 address from your instance. You can view these addresses in the console through either the **Instances** page or the **Network Interfaces** page.

Contents

- [Viewing the IPv6 addresses \(p. 765\)](#)
- [Assigning an IPv6 address to an instance \(p. 766\)](#)
- [Unassigning an IPv6 address from an instance \(p. 767\)](#)

Viewing the IPv6 addresses

You can use the Amazon EC2 console, AWS CLI, and instance metadata to view the IPv6 addresses for your instances.

New console

To view the IPv4 addresses for an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. On the **Networking** tab, locate **IPv6 addresses**.
5. Alternatively, under **Network interfaces** on the **Networking** tab, choose the interface ID for the network interface (for example, eni-123abc456def78901). Locate **IPv6 IPs**.

Old console

To view the IPv4 addresses for an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. On the **Networking** tab, locate **IPv6 IPs**.
5. Alternatively, under **Network interfaces** on the **Description** tab, choose **eth0**, and then choose the interface ID (for example, eni-123abc456def78901). Locate **IPv6 IPs**.

To view the IPv6 addresses for an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-instances](#) (AWS CLI)
- [Get-EC2Instance](#) (AWS Tools for Windows PowerShell).

To view the IPv6 addresses for an instance using instance metadata

1. Connect to your instance. For more information, see [Connect to your Linux instance \(p. 563\)](#).
2. Use the following command to view the IPv6 address (you can get the MAC address from <http://169.254.169.254/latest/meta-data/network/interfaces/macs/>).

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

Assigning an IPv6 address to an instance

If your VPC and subnet have IPv6 CIDR blocks associated with them, you can assign an IPv6 address to your instance during or after launch. The IPv6 address is assigned from the IPv6 address range of the subnet, and is assigned to the network interface with the device index of eth0.

To assign an IPv6 address to an instance during launch

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select an AMI and an instance type that supports IPv6, and choose **Next: Configure Instance Details**.
3. On the **Configure Instance Details** page, for **Network**, select a VPC and for **Subnet**, select a subnet. For **Auto-assign IPv6 IP**, choose **Enable**.
4. Follow the remaining steps in the wizard to launch your instance.

To assign an IPv6 address to an instance after launch

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.

3. Select your instance, and choose **Actions, Networking, Manage IP addresses**.
4. Expand the network interface. Under **IPv6 addresses**, choose **Assign new IP address**. Enter an IPv6 address from the range of the subnet or leave the field blank to let Amazon choose an IPv6 address for you.
5. Choose **Save**.

To assign an IPv6 address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- Use the `--ipv6-addresses` option with the `run-instances` command (AWS CLI)
- Use the `Ipv6Addresses` property for `-NetworkInterface` in the `New-EC2Instance` command (AWS Tools for Windows PowerShell)
- `assign-ipv6-addresses` (AWS CLI)
- `Register-EC2Ipv6AddressList` (AWS Tools for Windows PowerShell)

Unassigning an IPv6 address from an instance

You can unassign an IPv6 address from an instance at any time.

To unassign an IPv6 address from an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance, and choose **Actions, Networking, Manage IP addresses**.
4. Expand the network interface. Under **IPv6 addresses**, choose **Unassign** next to the IPv6 address.
5. Choose **Save**.

To unassign an IPv6 address from an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `unassign-ipv6-addresses` (AWS CLI)
- `Unregister-EC2Ipv6AddressList` (AWS Tools for Windows PowerShell).

Multiple IP addresses

You can specify multiple private IPv4 and IPv6 addresses for your instances. The number of network interfaces and private IPv4 and IPv6 addresses that you can specify for an instance depends on the instance type. For more information, see [IP addresses per network interface per instance type \(p. 791\)](#).

It can be useful to assign multiple IP addresses to an instance in your VPC to do the following:

- Host multiple websites on a single server by using multiple SSL certificates on a single server and associating each certificate with a specific IP address.
- Operate network appliances, such as firewalls or load balancers, that have multiple IP addresses for each network interface.
- Redirect internal traffic to a standby instance in case your instance fails, by reassigning the secondary IP address to the standby instance.

Contents

- [How multiple IP addresses work \(p. 768\)](#)
- [Working with multiple IPv4 addresses \(p. 768\)](#)
- [Working with multiple IPv6 addresses \(p. 772\)](#)

How multiple IP addresses work

The following list explains how multiple IP addresses work with network interfaces:

- You can assign a secondary private IPv4 address to any network interface. The network interface need not be attached to the instance.
- You can assign multiple IPv6 addresses to a network interface that's in a subnet that has an associated IPv6 CIDR block.
- You must choose a secondary IPv4 address from the IPv4 CIDR block range of the subnet for the network interface.
- You must choose IPv6 addresses from the IPv6 CIDR block range of the subnet for the network interface.
- You associate security groups with network interfaces, not individual IP addresses. Therefore, each IP address you specify in a network interface is subject to the security group of its network interface.
- Multiple IP addresses can be assigned and unassigned to network interfaces attached to running or stopped instances.
- Secondary private IPv4 addresses that are assigned to a network interface can be reassigned to another one if you explicitly allow it.
- An IPv6 address cannot be reassigned to another network interface; you must first unassign the IPv6 address from the existing network interface.
- When assigning multiple IP addresses to a network interface using the command line tools or API, the entire operation fails if one of the IP addresses can't be assigned.
- Primary private IPv4 addresses, secondary private IPv4 addresses, Elastic IP addresses, and IPv6 addresses remain with a secondary network interface when it is detached from an instance or attached to an instance.
- Although you can't detach the primary network interface from an instance, you can reassign the secondary private IPv4 address of the primary network interface to another network interface.

The following list explains how multiple IP addresses work with Elastic IP addresses (IPv4 only):

- Each private IPv4 address can be associated with a single Elastic IP address, and vice versa.
- When a secondary private IPv4 address is reassigned to another interface, the secondary private IPv4 address retains its association with an Elastic IP address.
- When a secondary private IPv4 address is unassigned from an interface, an associated Elastic IP address is automatically disassociated from the secondary private IPv4 address.

Working with multiple IPv4 addresses

You can assign a secondary private IPv4 address to an instance, associate an Elastic IPv4 address with a secondary private IPv4 address, and unassign a secondary private IPv4 address.

Contents

- [Assigning a secondary private IPv4 address \(p. 769\)](#)

- [Configuring the operating system on your instance to recognize secondary private IPv4 addresses \(p. 770\)](#)
- [Associating an Elastic IP address with the secondary private IPv4 address \(p. 771\)](#)
- [Viewing your secondary private IPv4 addresses \(p. 771\)](#)
- [Unassigning a secondary private IPv4 address \(p. 771\)](#)

Assigning a secondary private IPv4 address

You can assign the secondary private IPv4 address to the network interface for an instance as you launch the instance, or after the instance is running. This section includes the following procedures.

- [To assign a secondary private IPv4 address when launching an instance \(p. 769\)](#)
- [To assign a secondary IPv4 address during launch using the command line \(p. 770\)](#)
- [To assign a secondary private IPv4 address to a network interface \(p. 770\)](#)
- [To assign a secondary private IPv4 to an existing instance using the command line \(p. 770\)](#)

To assign a secondary private IPv4 address when launching an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. Select an AMI, then choose an instance type and choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, for **Network**, select a VPC and for **Subnet**, select a subnet.
5. In the **Network Interfaces** section, do the following, and then choose **Next: Add Storage**:
 - To add another network interface, choose **Add Device**. The console enables you to specify up to two network interfaces when you launch an instance. After you launch the instance, choose **Network Interfaces** in the navigation pane to add additional network interfaces. The total number of network interfaces that you can attach varies by instance type. For more information, see [IP addresses per network interface per instance type \(p. 791\)](#).

Important

When you add a second network interface, the system can no longer auto-assign a public IPv4 address. You will not be able to connect to the instance over IPv4 unless you assign an Elastic IP address to the primary network interface (eth0). You can assign the Elastic IP address after you complete the Launch wizard. For more information, see [Working with Elastic IP addresses \(p. 782\)](#).

6. For each network interface, under **Secondary IP addresses**, choose **Add IP**, and then enter a private IP address from the subnet range, or accept the default **Auto-assign** value to let Amazon select an address.
7. On the next **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume), and then choose **Next: Add Tags**.
8. On the **Add Tags** page, specify tags for the instance, such as a user-friendly name, and then choose **Next: Configure Security Group**.
9. On the **Configure Security Group** page, select an existing security group or create a new one. Choose **Review and Launch**.
10. On the **Review Instance Launch** page, review your settings, and then choose **Launch** to choose a key pair and launch your instance. If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

Important

After you have added a secondary private IP address to a network interface, you must connect to the instance and configure the secondary private IP address on the instance itself. For more

information, see [Configuring the operating system on your instance to recognize secondary private IPv4 addresses \(p. 770\)](#).

To assign a secondary IPv4 address during launch using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).
 - The `--secondary-private-ip-addresses` option with the `run-instances` command (AWS CLI)
 - Define `-NetworkInterface` and specify the `PrivateIpAddresses` parameter with the `New-EC2Instance` command (AWS Tools for Windows PowerShell).

To assign a secondary private IPv4 address to a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**, and then select the network interface attached to the instance.
3. Choose **Actions, Manage IP Addresses**.
4. Under **IPv4 Addresses**, choose **Assign new IP**.
5. Enter a specific IPv4 address that's within the subnet range for the instance, or leave the field blank to let Amazon select an IP address for you.
6. (Optional) Choose **Allow reassignment** to allow the secondary private IP address to be reassigned if it is already assigned to another network interface.
7. Choose **Yes, Update**.

Alternatively, you can assign a secondary private IPv4 address to an instance. Choose **Instances** in the navigation pane, select the instance, and then choose **Actions, Networking, Manage IP Addresses**. You can configure the same information as you did in the steps above. The IP address is assigned to the primary network interface (`eth0`) for the instance.

To assign a secondary private IPv4 to an existing instance using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).
 - `assign-private-ip-addresses` (AWS CLI)
 - `Register-EC2PrivateIpAddress` (AWS Tools for Windows PowerShell)

Configuring the operating system on your instance to recognize secondary private IPv4 addresses

After you assign a secondary private IPv4 address to your instance, you need to configure the operating system on your instance to recognize the secondary private IP address.

- If you are using Amazon Linux, the `ec2-net-utils` package can take care of this step for you. It configures additional network interfaces that you attach while the instance is running, refreshes secondary IPv4 addresses during DHCP lease renewal, and updates the related routing rules. You can immediately refresh the list of interfaces by using the command `sudo service network restart` and then view the up-to-date list using `ip addr 1i`. If you require manual control over your network configuration, you can remove the `ec2-net-utils` package. For more information, see [Configuring your network interface using ec2-net-utils \(p. 811\)](#).
- If you are using another Linux distribution, see the documentation for your Linux distribution. Search for information about configuring additional network interfaces and secondary IPv4 addresses. If the

instance has two or more interfaces on the same subnet, search for information about using routing rules to work around asymmetric routing.

For information about configuring a Windows instance, see [Configuring a secondary private IP address for your Windows instance in a VPC](#) in the *Amazon EC2 User Guide for Windows Instances*.

Associating an Elastic IP address with the secondary private IPv4 address

To associate an Elastic IP address with a secondary private IPv4 address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Actions**, and then select **Associate address**.
4. For **Network interface**, select the network interface, and then select the secondary IP address from the **Private IP** list.
5. Choose **Associate**.

To associate an Elastic IP address with a secondary private IPv4 address using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).
 - `associate-address` (AWS CLI)
 - `Register-EC2Address` (AWS Tools for Windows PowerShell)

Viewing your secondary private IPv4 addresses

To view the private IPv4 addresses assigned to a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface with private IP addresses to view.
4. On the **Details** tab in the details pane, check the **Primary private IPv4 IP** and **Secondary private IPv4 IPs** fields for the primary private IPv4 address and any secondary private IPv4 addresses assigned to the network interface.

To view the private IPv4 addresses assigned to an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance with private IPv4 addresses to view.
4. On the **Description** tab in the details pane, check the **Private IPs** and **Secondary private IPs** fields for the primary private IPv4 address and any secondary private IPv4 addresses assigned to the instance through its network interface.

Unassigning a secondary private IPv4 address

If you no longer require a secondary private IPv4 address, you can unassign it from the instance or the network interface. When a secondary private IPv4 address is unassigned from a network interface, the Elastic IP address (if it exists) is also disassociated.

To unassign a secondary private IPv4 address from an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select an instance, choose **Actions, Networking, Manage IP Addresses**.
4. Under **IPv4 Addresses**, choose **Unassign** for the IPv4 address to unassign.
5. Choose **Yes, Update**.

To unassign a secondary private IPv4 address from a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface, choose **Actions, Manage IP Addresses**.
4. Under **IPv4 Addresses**, choose **Unassign** for the IPv4 address to unassign.
5. Choose **Yes, Update**.

To unassign a secondary private IPv4 address using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).
 - [unassign-private-ip-addresses](#) (AWS CLI)
 - [Unregister-EC2PrivateIpAddress](#) (AWS Tools for Windows PowerShell)

Working with multiple IPv6 addresses

You can assign multiple IPv6 addresses to your instance, view the IPv6 addresses assigned to your instance, and unassign IPv6 addresses from your instance.

Contents

- [Assigning multiple IPv6 addresses \(p. 772\)](#)
- [Viewing your IPv6 addresses \(p. 774\)](#)
- [Unassigning an IPv6 address \(p. 774\)](#)

Assigning multiple IPv6 addresses

You can assign one or more IPv6 addresses to your instance during launch or after launch. To assign an IPv6 address to an instance, the VPC and subnet in which you launch the instance must have an associated IPv6 CIDR block. For more information, see [VPCs and Subnets](#) in the *Amazon VPC User Guide*.

To assign multiple IPv6 addresses during launch

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the dashboard, choose **Launch Instance**.
3. Select an AMI, choose an instance type, and choose **Next: Configure Instance Details**. Ensure that you choose an instance type that supports IPv6. For more information, see [Instance types \(p. 200\)](#).
4. On the **Configure Instance Details** page, select a VPC from the **Network** list, and a subnet from the **Subnet** list.
5. In the **Network Interfaces** section, do the following, and then choose **Next: Add Storage**:

- To assign a single IPv6 address to the primary network interface (eth0), under **IPv6 IPs**, choose **Add IP**. To add a secondary IPv6 address, choose **Add IP** again. You can enter an IPv6 address from the range of the subnet, or leave the default **Auto-assign** value to let Amazon choose an IPv6 address from the subnet for you.
 - Choose **Add Device** to add another network interface and repeat the steps above to add one or more IPv6 addresses to the network interface. The console enables you to specify up to two network interfaces when you launch an instance. After you launch the instance, choose **Network Interfaces** in the navigation pane to add additional network interfaces. The total number of network interfaces that you can attach varies by instance type. For more information, see [IP addresses per network interface per instance type \(p. 791\)](#).
6. Follow the next steps in the wizard to attach volumes and tag your instance.
 7. On the **Configure Security Group** page, select an existing security group or create a new one. If you want your instance to be reachable over IPv6, ensure that your security group has rules that allow access from IPv6 addresses. For more information, see [Security group rules reference \(p. 1011\)](#). Choose **Review and Launch**.
 8. On the **Review Instance Launch** page, review your settings, and then choose **Launch** to choose a key pair and launch your instance. If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

You can use the **Instances** screen Amazon EC2 console to assign multiple IPv6 addresses to an existing instance. This assigns the IPv6 addresses to the primary network interface (eth0) for the instance. To assign a specific IPv6 address to the instance, ensure that the IPv6 address is not already assigned to another instance or network interface.

To assign multiple IPv6 addresses to an existing instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance, choose **Actions, Networking, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Assign new IP** for each IPv6 address you want to add. You can specify an IPv6 address from the range of the subnet, or leave the **Auto-assign** value to let Amazon choose an IPv6 address for you.
5. Choose **Yes, Update**.

Alternatively, you can assign multiple IPv6 addresses to an existing network interface. The network interface must have been created in a subnet that has an associated IPv6 CIDR block. To assign a specific IPv6 address to the network interface, ensure that the IPv6 address is not already assigned to another network interface.

To assign multiple IPv6 addresses to a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select your network interface, choose **Actions, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Assign new IP** for each IPv6 address you want to add. You can specify an IPv6 address from the range of the subnet, or leave the **Auto-assign** value to let Amazon choose an IPv6 address for you.
5. Choose **Yes, Update**.

CLI overview

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- **Assign an IPv6 address during launch:**

- Use the `--ipv6-addresses` or `--ipv6-address-count` options with the [run-instances](#) command (AWS CLI)
- Define `-NetworkInterface` and specify the `Ipv6Addresses` or `Ipv6AddressCount` parameters with the [New-EC2Instance](#) command (AWS Tools for Windows PowerShell).

- **Assign an IPv6 address to a network interface:**

- [assign-ipv6-addresses](#) (AWS CLI)
- [Register-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell)

Viewing your IPv6 addresses

You can view the IPv6 addresses for an instance or for a network interface.

To view the IPv6 addresses assigned to an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance. In the details pane, review the **IPv6 IPs** field.

To view the IPv6 addresses assigned to a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select your network interface. In the details pane, review the **IPv6 IPs** field.

CLI overview

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- **View the IPv6 addresses for an instance:**

- [describe-instances](#) (AWS CLI)
- [Get-EC2Instance](#) (AWS Tools for Windows PowerShell).

- **View the IPv6 addresses for a network interface:**

- [describe-network-interfaces](#) (AWS CLI)
- [Get-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Unassigning an IPv6 address

You can unassign an IPv6 address from the primary network interface of an instance, or you can unassign an IPv6 address from a network interface.

To unassign an IPv6 address from an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.

3. Select your instance, choose **Actions, Networking, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Unassign** for the IPv6 address to unassign.
5. Choose **Yes, Update**.

To unassign an IPv6 address from a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select your network interface, choose **Actions, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Unassign** for the IPv6 address to unassign.
5. Choose **Save**.

CLI overview

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [unassign-ipv6-addresses](#) (AWS CLI)
- [Unregister-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell).

Bring your own IP addresses (BYOIP) in Amazon EC2

You can bring part or all of your public IPv4 address range or IPv6 address range from your on-premises network to your AWS account. You continue to own the address range, but AWS advertises it on the internet by default. After you bring the address range to AWS, it appears in your account as an address pool.

BYOIP is not available in all Regions and for all resources. For a list of supported Regions and resources, see the [FAQ for Bring Your Own IP](#).

Note

The following steps describe how to bring your own IP address range for use in Amazon EC2 only. For steps to bring your own IP address range for use in AWS Global Accelerator, see [Bring your own IP addresses \(BYOIP\) in the AWS Global Accelerator Developer Guide](#).

Topics

- [Requirements \(p. 775\)](#)
- [Prepare to bring your address range to your AWS account \(p. 776\)](#)
- [Provision the address range for use with AWS \(p. 778\)](#)
- [Advertise the address range through AWS \(p. 779\)](#)
- [Work with your address range \(p. 779\)](#)
- [Deprovision the address range \(p. 780\)](#)

Requirements

- The address range must be registered with your Regional internet registry (RIR), such as the American Registry for Internet Numbers (ARIN), Réseaux IP Européens Network Coordination Centre (RIPE), or

Asia-Pacific Network Information Centre (APNIC). It must be registered to a business or institutional entity and cannot be registered to an individual person.

- The most specific IPv4 address range that you can bring is /24.
- The most specific IPv6 address range that you can bring is /48 for CIDRs that are publicly advertised, and /56 for CIDRs that are [not publicly advertised \(p. 778\)](#).
- You can bring each address range to one Region at a time.
- You can bring a total of five IPv4 and IPv6 address ranges per Region to your AWS account.
- The addresses in the IP address range must have a clean history. We might investigate the reputation of the IP address range and reserve the right to reject an IP address range if it contains an IP address that has a poor reputation or is associated with malicious behavior.
- You must own the IP address that you use. This means that only the following are supported:
 - ARIN - "Direct Allocation" and "Direct Assignment" network types
 - RIPE - "ALLOCATED PA", "LEGACY", "ASSIGNED PI", and "ALLOCATED-BY-RIR" allocation statuses
 - APNIC – "ALLOCATED PORTABLE" and "ASSIGNED PORTABLE" allocation statuses

Prepare to bring your address range to your AWS account

To ensure that only you can bring your address range to your AWS account, you must authorize Amazon to advertise the address range. You must also provide proof that you own the address range through a signed authorization message.

A Route Origin Authorization (ROA) is a cryptographic statement about your route announcements that you can create through your RIR. It contains the address range, the Autonomous System numbers (ASN) that are allowed to advertise the address range, and an expiration date. An ROA authorizes Amazon to advertise an address range under a specific AS number. However, it does not authorize your AWS account to bring the address range to AWS. To authorize your AWS account to bring an address range to AWS, you must publish a self-signed X509 certificate in the Registry Data Access Protocol (RDAP) remarks for the address range. The certificate contains a public key, which AWS uses to verify the authorization-context signature that you provide. Keep your private key secure and use it to sign the authorization-context message.

The commands in these tasks are supported on Linux. On Windows, you can use the [Windows Subsystem for Linux](#) to run Linux commands.

Tasks

- [Create a ROA object \(p. 776\)](#)
- [Create a self-signed X509 certificate \(p. 777\)](#)
- [Create a signed authorization message \(p. 777\)](#)

Create a ROA object

Create a ROA object to authorize Amazon ASNs 16509 and 14618 to advertise your address range, plus the ASNs that are currently authorized to advertise the address range. You must set the maximum length to the size of the smallest prefix that you want to bring (for example, /24). It might take up to 24 hours for the ROA to become available to Amazon. For more information, see the following:

- ARIN — [ROA Requests](#)
- RIPE — [Managing ROAs](#)

- APNIC — Route Management

Create a self-signed X509 certificate

Use the following procedure to create a self-signed X509 certificate and add it to the RDAP record for your RIR. The **openssl** commands require OpenSSL version 1.0.2 or later.

Copy the commands below and replace only the placeholder values (in colored italic text).

To create a self-signed X509 certificate and add it to the RDAP record

1. Generate an RSA 2048-bit key pair as shown in the following.

```
openssl genrsa -out private.key 2048
```

2. Create a public X509 certificate from the key pair using the following command. In this example, the certificate expires in 365 days, after which time it cannot be trusted. Be sure to set the expiration appropriately. When prompted for information, you can accept the default values.

```
openssl req -new -x509 -key private.key -days 365 | tr -d "\n" > publickey.cer
```

3. Update the RDAP record for your RIR with the X509 certificate. Be sure to copy the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- from the certificate. Be sure that you have removed newline characters, if you haven't already done so using the **tr -d "\n"** commands in the previous steps. To view your certificate, run the following command.

```
cat publickey.cer
```

For ARIN, add the certificate in the "Public Comments" section for your address range. Do not add it to the comments section for your organization.

For RIPE, add the certificate as a new "descr" field for your address range. Do not add it to the comments section for your organization.

For APNIC, email the public key to helpdesk@apnic.net to manually add it to the "remarks" field for your address range. Send the email using the APNIC authorized contact for the IP addresses.

Create a signed authorization message

The format of the signed authorization message is as follows, where the date is the expiry date of the message.

```
1|aws|account|cidr|YYYYMMDD|SHA256|RSAPSS
```

To create a signed authorization message

1. Create a plaintext authorization message and store it in a variable named `text_message` as shown in the following example. Copy the following example and replace only the example account number, address range, and expiry date with your own values.

```
text_message="1|aws|123456789012|198.51.100.0/24|20191201|SHA256|RSAPSS"
```

2. Sign the authorization message in `text_message` using the key pair that you created, and store it in a variable named `signed_message`.

```
signed_message=$(echo $text_message | tr -d "\n" | openssl dgst -sha256 -sigopt rsa_padding_mode:pss -sigopt rsa_pss_saltlen:-1 -sign private.key -keyform PEM | openssl base64 | tr -- '+=' '-'_-' | tr -d "\n")
```

Important

We recommend that you copy and paste this command. Do not modify or replace any of the values.

Provision the address range for use with AWS

When you provision an address range for use with AWS, you are confirming that you own the address range and are authorizing Amazon to advertise it. We also verify that you own the address range through a signed authorization message. This message is signed with the self-signed X509 key pair that you used when updating the RDAP record with the X509 certificate.

To provision the address range, use the following [provision-byoip-cidr](#) command. Replace the example address range with your own address range. The `--cidr-authorization-context` option uses the variables that you created previously, not the ROA message.

```
aws ec2 provision-byoip-cidr --cidr address-range --cidr-authorization-context  
Message="$text_message",Signature="$signed_message"
```

Provisioning an address range is an asynchronous operation, so the call returns immediately, but the address range is not ready to use until its status changes from `pending-provision` to `provisioned`. It can take up to three weeks to complete the provisioning process. To monitor the status of the address ranges that you've provisioned, use the following [describe-byoip-cidrs](#) command.

```
aws ec2 describe-byoip-cidrs --max-results 5
```

If there are issues during provisioning and the status goes to `failed-provision`, you must run the `provision-byoip-cidr` command again after the issues have been resolved.

Provision an IPv6 address range that's not publicly advertised

By default, an address range is provisioned to be publicly advertised to the internet. You can provision an IPv6 address range that will not be publicly advertised. When you associate an IPv6 CIDR block from a non-public address range with a VPC, the IPv6 CIDR can only be accessed through an AWS Direct Connect connection.

An ROA is not required to provision a non-public address range.

To provision an IPv6 address range that will not be publicly advertised, use the following [provision-byoip-cidr](#) command.

```
aws ec2 provision-byoip-cidr --cidr address-range --cidr-authorization-context  
Message="$text_message",Signature="$signed_message" --no-publicly-advertisable
```

Important

You can only set the `publicly-advertisable` or `no-publicly-advertisable` flag during provisioning. You cannot change the advertisable status of an address range later.

Advertise the address range through AWS

After the address range is provisioned, it is ready to be advertised. You must advertise the exact address range that you provisioned. You can't advertise only a portion of the provisioned address range.

If you provisioned an IPv6 address range that will not be publicly advertised, you do not need to complete this step.

We recommend that you stop advertising the address range from other locations before you advertise it through AWS. If you keep advertising your IP address range from other locations, we can't reliably support it or troubleshoot issues. Specifically, we can't guarantee that traffic to the address range will enter our network.

To minimize down time, you can configure your AWS resources to use an address from your address pool before it is advertised, and then simultaneously stop advertising it from the current location and start advertising it through AWS. For more information about allocating an Elastic IP address from your address pool, see [Allocating an Elastic IP address \(p. 782\)](#).

To advertise the address range, use the following [advertise-byoip-cidr](#) command.

```
aws ec2 advertise-byoip-cidr --cidr address-range
```

Important

You can run the [advertise-byoip-cidr](#) command at most once every 10 seconds, even if you specify different address ranges each time.

To stop advertising the address range, use the following [withdraw-byoip-cidr](#) command.

```
aws ec2 withdraw-byoip-cidr --cidr address-range
```

Important

You can run the [withdraw-byoip-cidr](#) command at most once every 10 seconds, even if you specify different address ranges each time.

Work with your address range

You can view and work with the IPv4 and IPv6 address ranges that you've provisioned in your account.

IPv4 address ranges

You can create an Elastic IP address from your IPv4 address pool and use it with your AWS resources, such as EC2 instances, NAT gateways, and Network Load Balancers.

To view information about the IPv4 address pools that you've provisioned in your account, use the following [describe-public-ipv4-pools](#) command.

```
aws ec2 describe-public-ipv4-pools
```

To create an Elastic IP address from your IPv4 address pool, use the [allocate-address](#) command. You can use the `--public-ipv4-pool` option to specify the ID of the address pool returned by [describe-byoip-cidrs](#). Or you can use the `--address` option to specify an address from the address range that you provisioned.

IPv6 address ranges

To view information about the IPv6 address pools that you've provisioned in your account, use the following [describe-ipv6-pools](#) command.

```
aws ec2 describe-ipv6-pools
```

To create a VPC and specify an IPv6 CIDR from your IPv6 address pool, use the following [create-vpc](#) command. To let Amazon choose the IPv6 CIDR from your IPv6 address pool, omit the `--ipv6-cidr-block` option.

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --ipv6-cidr-block ipv6-cidr --ipv6-pool pool-id
```

To associate an IPv6 CIDR block from your IPv6 address pool with a VPC, use the following [associate-vpc-cidr-block](#) command. To let Amazon choose the IPv6 CIDR from your IPv6 address pool, omit the `--ipv6-cidr-block` option.

```
aws ec2 associate-vpc-cidr-block --vpc-id vpc-123456789abc123ab --ipv6-cidr-block ipv6-cidr --ipv6-pool pool-id
```

To view your VPCs and the associated IPv6 address pool information, use the [describe-vpcs](#) command. To view information about associated IPv6 CIDR blocks from a specific IPv6 address pool, use the following [get-associated-ipv6-pool-cidrs](#) command.

```
aws ec2 get-associated-ipv6-pool-cidrs --pool-id pool-id
```

If you disassociate the IPv6 CIDR block from your VPC, it's released back into your IPv6 address pool.

For more information about working with IPv6 CIDR blocks in the VPC console, see [Working with VPCs and Subnets](#) in the *Amazon VPC User Guide*.

Deprovision the address range

To stop using your address range with AWS, first release any Elastic IP addresses and disassociate any IPv6 CIDR blocks that are still allocated from the address pool. Then stop advertising the address range, and finally, deprovision the address range.

You cannot deprovision a portion of the address range. If you want to use a more specific address range with AWS, deprovision the entire address range and provision a more specific address range.

(IPv4) To release each Elastic IP address, use the following [release-address](#) command.

```
aws ec2 release-address --allocation-id eipalloc-123456789abcabcabc
```

(IPv6) To disassociate an IPv6 CIDR block, use the following [disassociate-vpc-cidr-block](#) command.

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-12345abcd1234abc1
```

To stop advertising the address range, use the following [withdraw-byoip-cidr](#) command.

```
aws ec2 withdraw-byoip-cidr --cidr address-range
```

To deprovision the address range, use the following [deprovision-byoip-cidr](#) command.

```
aws ec2 deprovision-byoip-cidr --cidr address-range
```

It can take up to a day to deprovision an address range.

Elastic IP addresses

An *Elastic IP address* is a static IPv4 address designed for dynamic cloud computing. By using an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account. An Elastic IP address is allocated to your AWS account, and is yours until you release it.

An Elastic IP address is a public IPv4 address, which is reachable from the internet. If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet. For example, this allows you to connect to your instance from your local computer.

We currently do not support Elastic IP addresses for IPv6.

Contents

- [Elastic IP address basics \(p. 781\)](#)
- [Working with Elastic IP addresses \(p. 782\)](#)
- [Using reverse DNS for email applications \(p. 788\)](#)
- [Elastic IP address limit \(p. 788\)](#)

Elastic IP address basics

The following are the basic characteristics of an Elastic IP address:

- An Elastic IP address is static; it does not change over time.
- To use an Elastic IP address, you first allocate one to your account, and then associate it with your instance or a network interface.
- When you associate an Elastic IP address with an instance, it is also associated with the instance's primary network interface. When you associate an Elastic IP address with a network interface that is attached to an instance, it is also associated with the instance.
- When you associate an Elastic IP address with an instance or its primary network interface, the instance's public IPv4 address (if it had one) is released back into Amazon's pool of public IPv4 addresses. You cannot reuse a public IPv4 address, and you cannot convert a public IPv4 address to an Elastic IP address. For more information, see [Public IPv4 addresses and external DNS hostnames \(p. 760\)](#).
- You can disassociate an Elastic IP address from a resource, and then associate it with a different resource. To avoid unexpected behavior, ensure that all active connections to the resource named in the existing association are closed before you make the change. After you have associated your Elastic IP address to a different resource, you can reopen your connections to the newly associated resource.
- A disassociated Elastic IP address remains allocated to your account until you explicitly release it.
- To ensure efficient use of Elastic IP addresses, we impose a small hourly charge if an Elastic IP address is not associated with a running instance, or if it is associated with a stopped instance or an unattached network interface. While your instance is running, you are not charged for one Elastic IP address associated with the instance, but you are charged for any additional Elastic IP addresses associated with the instance. For more information, see the section for Elastic IP Addresses on the [Amazon EC2 Pricing, On-Demand Pricing page](#).
- When you associate an Elastic IP address with an instance that previously had a public IPv4 address, the public DNS host name of the instance changes to match the Elastic IP address.

- We resolve a public DNS host name to the public IPv4 address or the Elastic IP address of the instance outside the network of the instance, and to the private IPv4 address of the instance from within the network of the instance.
- An Elastic IP address comes from Amazon's pool of IPv4 addresses, or from a custom IP address pool that you have brought to your AWS account.
- When you allocate an Elastic IP address from an IP address pool that you have brought to your AWS account, it does not count toward your Elastic IP address limits. For more information, see [Elastic IP address limit \(p. 788\)](#).
- When you allocate the Elastic IP addresses, you can associate the Elastic IP addresses with a network border group. This is the location from which we advertise the CIDR block. Setting the network border group limits the CIDR block to this group. If you do not specify the network border group, we set the border group containing all of the Availability Zones in the Region (for example, us-west-2).
- An Elastic IP address is for use in a specific network border group only.
- An Elastic IP address is for use in a specific Region only, and cannot be moved to a different Region.

Working with Elastic IP addresses

The following sections describe how you can work with Elastic IP addresses.

Tasks

- [Allocating an Elastic IP address \(p. 782\)](#)
- [Describing your Elastic IP addresses \(p. 783\)](#)
- [Tagging an Elastic IP address \(p. 784\)](#)
- [Associating an Elastic IP address with a running instance or network interface \(p. 785\)](#)
- [Disassociating an Elastic IP address \(p. 786\)](#)
- [Releasing an Elastic IP address \(p. 787\)](#)
- [Recovering an Elastic IP address \(p. 787\)](#)

Allocating an Elastic IP address

You can allocate an Elastic IP address from Amazon's pool of public IPv4 addresses, or from a custom IP address pool that you have brought to your AWS account. For more information about bringing your own IP address range to your AWS account, see [Bring your own IP addresses \(BYOIP\) in Amazon EC2 \(p. 775\)](#).

You can allocate an Elastic IP address using one of the following methods.

New console

To allocate an Elastic IP address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate Elastic IP address**.
4. For **Scope**, choose either **VPC** or **EC2-Classic** depending on the scope in which it will be used.
5. (VPC scope only) For **Public IPv4 address pool** choose one of the following:
 - **Amazon's pool of IP addresses**—If you want an IPv4 address to be allocated from Amazon's pool of IP addresses.
 - **My pool of public IPv4 addresses**—If you want to allocate an IPv4 address from an IP address pool that you have brought to your AWS account. This option is disabled if you do not have any IP address pools.

- **Customer owned pool of IPv4 addresses**—If you want to allocate an IPv4 address from a pool created from your on-premises network for use with an AWS Outpost. This option is disabled if you do not have an AWS Outpost.
6. Choose **Allocate**.

Old console

To allocate an Elastic IP address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate new address**.
4. For **IPv4 address pool**, choose **Amazon pool**.
5. Choose **Allocate**, and close the confirmation screen.

AWS CLI

To allocate an Elastic IP address

Use the `allocate-address` AWS CLI command.

PowerShell

To allocate an Elastic IP address

Use the `New-EC2Address` AWS Tools for Windows PowerShell command.

Describing your Elastic IP addresses

You can describe an Elastic IP address using one of the following methods.

New console

To describe your Elastic IP addresses

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to view and choose **Actions, View details**.

Old console

To describe your Elastic IP addresses

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select a filter from the Resource Attribute list to begin searching. You can use multiple filters in a single search.

AWS CLI

To describe your Elastic IP addresses

Use the `describe-addresses` AWS CLI command.

PowerShell

To describe your Elastic IP addresses

Use the [Get-EC2Address](#) AWS Tools for Windows PowerShell command.

Tagging an Elastic IP address

You can assign custom tags to your Elastic IP addresses to categorize them in different ways, for example, by purpose, owner, or environment. This helps you to quickly find a specific Elastic IP address based on the custom tags that you assigned to it.

You can only tag Elastic IP addresses that are in the VPC scope.

Note

Cost allocation tracking using Elastic IP address tags is not supported.

You can tag an Elastic IP address using one of the following methods.

New console

To tag an Elastic IP address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to tag and choose **Actions, View details**.
4. In the **Tags** section, choose **Manage tags**.
5. Specify a tag key and value pair.
6. (Optional) Choose **Add tag** to add additional tags.
7. Choose **Save**.

Old console

To tag an Elastic IP address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to tag and choose **Tags**.
4. Choose **Add/Edit Tags**.
5. In the **Add/Edit Tags** dialog box, choose **Create Tag**, and then specify the key and value for the tag.
6. (Optional) Choose **Create Tag** to add additional tags to the Elastic IP address.
7. Choose **Save**.

AWS CLI

To tag an Elastic IP address

Use the [create-tags](#) AWS CLI command.

```
aws ec2 create-tags --resources eipalloc-12345678 --tags Key=Owner,Value=TeamA
```

PowerShell

To tag an Elastic IP address

Use the [New-EC2Tag](#) AWS Tools for Windows PowerShell command.

The New-EC2Tag command needs a Tag parameter, which specifies the key and value pair to be used for the Elastic IP address tag. The following commands create the Tag parameter.

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag
PS C:\> $tag.Key = "Owner"
PS C:\> $tag.Value = "TeamA"
```

```
PS C:\> New-EC2Tag -Resource eipalloc-12345678 -Tag $tag
```

Associating an Elastic IP address with a running instance or network interface

If you're associating an Elastic IP address with your instance to enable communication with the internet, you must also ensure that your instance is in a public subnet. For more information, see [Internet Gateways](#) in the *Amazon VPC User Guide*.

You can associate an Elastic IP address with an instance or network interface using one of the following methods.

New console

To associate an Elastic IP address with an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to associate and choose **Actions, Associate Elastic IP address**.
4. For **Resource type**, choose **Instance**.
5. For instance, choose the instance with which to associate the Elastic IP address. You can also enter text to search for a specific instance.
6. (Optional) For **Private IP address**, specify a private IP address with which to associate the Elastic IP address.
7. Choose **Associate**.

To associate an Elastic IP address with a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to associate and choose **Actions, Associate Elastic IP address**.
4. For **Resource type**, choose **Network interface**.
5. For **Network interface**, choose the network interface with which to associate the Elastic IP address. You can also enter text to search for a specific network interface.
6. (Optional) For **Private IP address**, specify a private IP address with which to associate the Elastic IP address.
7. Choose **Associate**.

Old console

To associate an Elastic IP address with an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select an Elastic IP address and choose **Actions, Associate address**.
4. Select the instance from **Instance** and then choose **Associate**.

AWS CLI

To associate an Elastic IP address

Use the [associate-address](#) AWS CLI command.

PowerShell

To associate an Elastic IP address

Use the [Register-EC2Address](#) AWS Tools for Windows PowerShell command.

Disassociating an Elastic IP address

You can disassociate an Elastic IP address from an instance or network interface at any time. After you disassociate the Elastic IP address, you can reassociate it with another resource.

You can disassociate an Elastic IP address using one of the following methods.

New console

To disassociate and reassociate an Elastic IP address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to disassociate, choose **Actions, Disassociate Elastic IP address**.
4. Choose **Disassociate**.

Old console

To disassociate and reassociate an Elastic IP address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address, choose **Actions**, and then select **Disassociate address**.
4. Choose **Disassociate address**.

AWS CLI

To disassociate an Elastic IP address

Use the [disassociate-address](#) AWS CLI command.

PowerShell

To disassociate an Elastic IP address

Use the [Unregister-EC2Address](#) AWS Tools for Windows PowerShell command.

Releasing an Elastic IP address

If you no longer need an Elastic IP address, we recommend that you release it using one of the following methods. The address to release must not be currently associated with an AWS resource, such as an EC2 instance, NAT gateway, or Network Load Balancer.

New console

To release an Elastic IP address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to release and choose **Actions, Release Elastic IP addresses**.
4. Choose **Release**.

Old console

To release an Elastic IP address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address, choose **Actions**, and then select **Release addresses**. Choose **Release** when prompted.

AWS CLI

To release an Elastic IP address

Use the [release-address](#) AWS CLI command.

PowerShell

To release an Elastic IP address

Use the [Remove-EC2Address](#) AWS Tools for Windows PowerShell command.

Recovering an Elastic IP address

If you have released your Elastic IP address, you might be able to recover it. The following rules apply:

- You cannot recover an Elastic IP address if it has been allocated to another AWS account, or if it will result in exceeding your Elastic IP address limit.
- You cannot recover tags associated with an Elastic IP address.
- You can recover an Elastic IP address using the Amazon EC2 API or a command line tool only.

AWS CLI

To recover an Elastic IP address

Use the [allocate-address](#) AWS CLI command and specify the IP address using the `--address` parameter as follows.

```
aws ec2 allocate-address --domain vpc --address 203.0.113.3
```

PowerShell

To recover an Elastic IP address

Use the [New-EC2Address](#) AWS Tools for Windows PowerShell command and specify the IP address using the `-Address` parameter as follows.

```
PS C:\> New-EC2Address -Address 203.0.113.3 -Domain vpc -Region us-east-1
```

Using reverse DNS for email applications

If you intend to send email to third parties from an instance, we suggest that you provision one or more Elastic IP addresses and provide them to AWS. AWS works with ISPs and internet anti-spam organizations to reduce the chance that your email sent from these addresses will be flagged as spam.

In addition, assigning a static reverse DNS record to your Elastic IP address that is used to send email can help avoid having email flagged as spam by some anti-spam organizations. Note that a corresponding forward DNS record (record type A) pointing to your Elastic IP address must exist before we can create your reverse DNS record.

If a reverse DNS record is associated with an Elastic IP address, the Elastic IP address is locked to your account and cannot be released from your account until the record is removed.

To remove email sending limits, or to provide us with your Elastic IP addresses and reverse DNS records, go to the [Request to Remove Email Sending Limitations](#) page.

Elastic IP address limit

By default, all AWS accounts are limited to five (5) Elastic IP addresses per Region, because public (IPv4) internet addresses are a scarce public resource. We strongly encourage you to use an Elastic IP address primarily for the ability to remap the address to another instance in the case of instance failure, and to use [DNS hostnames](#) for all other inter-node communication.

To verify how many Elastic IP addresses are in use

Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/> and choose **Elastic IPs** from the navigation pane.

To verify your current account limit for Elastic IP addresses

You can verify your limit in either the Amazon EC2 console or the Service Quotas console. Do one of the following:

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

Choose **Limits** from the navigation pane, and then enter **IP** in the search field. The limit is **EC2-VPC Elastic IPs**. If you have access to EC2-Classic, there is an additional limit, **EC2-Classic Elastic IPs**.

- Open the Service Quotas console at <https://console.aws.amazon.com/servicequotas/>.

On the Dashboard, choose **Amazon Elastic Compute Cloud (Amazon EC2)**. If Amazon Elastic Compute Cloud (Amazon EC2) is not listed on the Dashboard, choose **AWS services**, enter **EC2** in the search field, and then choose **Amazon Elastic Compute Cloud (Amazon EC2)**.

On the Amazon EC2 service quotas page, enter **IP** in the search field. The limit is **EC2-VPC Elastic IPs**. If you have access to EC2-Classic, there is an additional limit, **EC2-Classic Elastic IPs**. For more information, choose the limit.

If you think your architecture warrants additional Elastic IP addresses, you can request a quota increase directly from the Service Quotas console.

Elastic network interfaces

An *elastic network interface* is a logical networking component in a VPC that represents a virtual network card. It can include the following attributes:

- A primary private IPv4 address from the IPv4 address range of your VPC
- One or more secondary private IPv4 addresses from the IPv4 address range of your VPC
- One Elastic IP address (IPv4) per private IPv4 address
- One public IPv4 address
- One or more IPv6 addresses
- One or more security groups
- A MAC address
- A source/destination check flag
- A description

You can create and configure network interfaces in your account and attach them to instances in your VPC. Your account might also have *requester-managed* network interfaces, which are created and managed by AWS services to enable you to use other resources and services. You cannot manage these network interfaces yourself. For more information, see [Requester-managed network interfaces \(p. 812\)](#).

This AWS resource is referred to as a *network interface* in the AWS Management Console and the Amazon EC2 API. Therefore, we use "network interface" in this documentation instead of "elastic network interface". The term "network interface" in this documentation always means "elastic network interface".

Contents

- [Network interface basics \(p. 789\)](#)
- [IP addresses per network interface per instance type \(p. 791\)](#)
- [Working with network interfaces \(p. 802\)](#)
- [Scenarios for network interfaces \(p. 808\)](#)
- [Best practices for configuring network interfaces \(p. 810\)](#)
- [Requester-managed network interfaces \(p. 812\)](#)

Network interface basics

You can create a network interface, attach it to an instance, detach it from an instance, and attach it to another instance. The attributes of a network interface follow it as it's attached or detached from an instance and reattached to another instance. When you move a network interface from one instance to another, network traffic is redirected to the new instance.

Primary network interface

Each instance has a default network interface, called the *primary network interface*. You cannot detach a primary network interface from an instance. You can create and attach additional network interfaces. The maximum number of network interfaces that you can use varies by instance type. For more information, see [IP addresses per network interface per instance type \(p. 791\)](#).

Public IPv4 addresses for network interfaces

In a VPC, all subnets have a modifiable attribute that determines whether network interfaces created in that subnet (and therefore instances launched into that subnet) are assigned a public IPv4 address. For more information, see [IP addressing behavior for your subnet](#) in the *Amazon VPC User Guide*. The public IPv4 address is assigned from Amazon's pool of public IPv4 addresses. When you launch an instance, the IP address is assigned to the primary network interface that's created.

When you create a network interface, it inherits the public IPv4 addressing attribute from the subnet. If you later modify the public IPv4 addressing attribute of the subnet, the network interface keeps the setting that was in effect when it was created. If you launch an instance and specify an existing network interface as the primary network interface, the public IPv4 address attribute is determined by this network interface.

For more information, see [Public IPv4 addresses and external DNS hostnames \(p. 760\)](#).

Elastic IP addresses for network interface

If you have an Elastic IP address, you can associate it with one of the private IPv4 addresses for the network interface. You can associate one Elastic IP address with each private IPv4 address.

If you disassociate an Elastic IP address from a network interface, you can release it back to the address pool. This is the only way to associate an Elastic IP address with an instance in a different subnet or VPC, as network interfaces are specific to subnets.

IPv6 addresses for network interfaces

If you associate IPv6 CIDR blocks with your VPC and subnet, you can assign one or more IPv6 addresses from the subnet range to a network interface. Each IPv6 address can be assigned to one network interface.

All subnets have a modifiable attribute that determines whether network interfaces created in that subnet (and therefore instances launched into that subnet) are automatically assigned an IPv6 address from the range of the subnet. For more information, see [IP addressing behavior for your subnet](#) in the *Amazon VPC User Guide*. When you launch an instance, the IPv6 address is assigned to the primary network interface that's created.

For more information, see [IPv6 addresses \(p. 761\)](#).

Termination behavior

You can set the termination behavior for a network interface that's attached to an instance. You can specify whether the network interface should be automatically deleted when you terminate the instance to which it's attached.

Source/destination checking

Disabling source/destination checking enables an instance to handle network traffic that isn't specifically destined for the instance. For example, instances running services such as network address translation, routing, or a firewall should disable the source/destination check attribute. This attribute is enabled by default.

Monitoring IP traffic

You can enable a VPC flow log on your network interface to capture information about the IP traffic going to and from a network interface. After you've created a flow log, you can view and retrieve its data in Amazon CloudWatch Logs. For more information, see [VPC Flow Logs](#) in the *Amazon VPC User Guide*.

IP addresses per network interface per instance type

The following table lists the maximum number of network interfaces per instance type, and the maximum number of private IPv4 addresses and IPv6 addresses per network interface. The limit for IPv6 addresses is separate from the limit for private IPv4 addresses per network interface. Not all instance types support IPv6 addressing. Network interfaces, multiple private IPv4 addresses, and IPv6 addresses are only available for instances running in a VPC. IPv6 addresses are public and reachable over the Internet. For more information, see [Multiple IP addresses \(p. 767\)](#). For more information about IPv6 in VPC, see [IP Addressing in your VPC](#) in the *Amazon VPC User Guide*.

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
a1.medium	2	4	4
a1.large	3	10	10
a1.xlarge	4	15	15
a1.2xlarge	4	15	15
a1.4xlarge	8	30	30
a1.metal	8	30	30
c1.medium	2	6	IPv6 not supported
c1.xlarge	4	15	IPv6 not supported
c3.large	3	10	10
c3.xlarge	4	15	15
c3.2xlarge	4	15	15
c3.4xlarge	8	30	30
c3.8xlarge	8	30	30
c4.large	3	10	10
c4.xlarge	4	15	15
c4.2xlarge	4	15	15
c4.4xlarge	8	30	30
c4.8xlarge	8	30	30
c5.large	3	10	10
c5.xlarge	4	15	15
c5.2xlarge	4	15	15
c5.4xlarge	8	30	30

Amazon Elastic Compute Cloud
 User Guide for Linux Instances
 IP addresses per network interface per instance type

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
c5.9xlarge	8	30	30
c5.12xlarge	8	30	30
c5.18xlarge	15	50	50
c5.24xlarge	15	50	50
c5.metal	15	50	50
c5a.large	3	10	10
c5a.xlarge	4	15	15
c5a.2xlarge	4	15	15
c5a.4xlarge	8	30	30
c5a.8xlarge	8	30	30
c5a.12xlarge	8	30	30
c5a.16xlarge	15	50	50
c5a.24xlarge	15	50	50
c5ad.large	3	10	10
c5ad.xlarge	4	15	15
c5ad.2xlarge	4	15	15
c5ad.4xlarge	8	30	30
c5ad.8xlarge	8	30	30
c5ad.12xlarge	8	30	30
c5ad.16xlarge	15	50	50
c5ad.24xlarge	15	50	50
c5d.large	3	10	10
c5d.xlarge	4	15	15
c5d.2xlarge	4	15	15
c5d.4xlarge	8	30	30
c5d.9xlarge	8	30	30
c5d.12xlarge	8	30	30
c5d.18xlarge	15	50	50
c5d.24xlarge	15	50	50
c5d.metal	15	50	50

Amazon Elastic Compute Cloud
 User Guide for Linux Instances
 IP addresses per network interface per instance type

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
c5n.large	3	10	10
c5n.xlarge	4	15	15
c5n.2xlarge	4	15	15
c5n.4xlarge	8	30	30
c5n.9xlarge	8	30	30
c5n.18xlarge	15	50	50
c5n.metal	15	50	50
c6g.medium	2	4	4
c6g.large	3	10	10
c6g.xlarge	4	15	15
c6g.2xlarge	4	15	15
c6g.4xlarge	8	30	30
c6g.8xlarge	8	30	30
c6g.12xlarge	8	30	30
c6g.16xlarge	15	50	50
c6g.metal	15	50	50
c6gd.medium	2	4	4
c6gd.large	3	10	10
c6gd.xlarge	4	15	15
c6gd.2xlarge	4	15	15
c6gd.4xlarge	8	30	30
c6gd.8xlarge	8	30	30
c6gd.12xlarge	8	30	30
c6gd.16xlarge	15	50	50
c6gd.metal	15	50	50
cc2.8xlarge	8	30	IPv6 not supported
cr1.8xlarge	8	30	IPv6 not supported
d2.xlarge	4	15	15
d2.2xlarge	4	15	15
d2.4xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
d2.8xlarge	8	30	30
f1.2xlarge	4	15	15
f1.4xlarge	8	30	30
f1.16xlarge	8	50	50
g2.2xlarge	4	15	IPv6 not supported
g2.8xlarge	8	30	IPv6 not supported
g3s.xlarge	4	15	15
g3.4xlarge	8	30	30
g3.8xlarge	8	30	30
g3.16xlarge	15	50	50
g4dn.xlarge	3	10	10
g4dn.2xlarge	3	10	10
g4dn.4xlarge	3	10	10
g4dn.8xlarge	4	15	15
g4dn.12xlarge	8	30	30
g4dn.16xlarge	4	15	15
g4dn.metal	15	50	50
h1.2xlarge	4	15	15
h1.4xlarge	8	30	30
h1.8xlarge	8	30	30
h1.16xlarge	15	50	50
hs1.8xlarge	8	30	IPv6 not supported
i2.xlarge	4	15	15
i2.2xlarge	4	15	15
i2.4xlarge	8	30	30
i2.8xlarge	8	30	30
i3.large	3	10	10
i3.xlarge	4	15	15
i3.2xlarge	4	15	15
i3.4xlarge	8	30	30

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
i3.8xlarge	8	30	30
i3.16xlarge	15	50	50
i3.metal	15	50	50
i3en.large	3	10	10
i3en.xlarge	4	15	15
i3en.2xlarge	4	15	15
i3en.3xlarge	4	15	15
i3en.6xlarge	8	30	30
i3en.12xlarge	8	30	30
i3en.24xlarge	15	50	50
i3en.metal	15	50	50
inf1.xlarge	4	10	10
inf1.2xlarge	4	10	10
inf1.6xlarge	8	30	30
inf1.24xlarge	15	30	30
m1.small	2	4	IPv6 not supported
m1.medium	2	6	IPv6 not supported
m1.large	3	10	IPv6 not supported
m1.xlarge	4	15	IPv6 not supported
m2.xlarge	4	15	IPv6 not supported
m2.2xlarge	4	30	IPv6 not supported
m2.4xlarge	8	30	IPv6 not supported
m3.medium	2	6	IPv6 not supported
m3.large	3	10	IPv6 not supported
m3.xlarge	4	15	IPv6 not supported
m3.2xlarge	4	30	IPv6 not supported
m4.large	2	10	10
m4.xlarge	4	15	15
m4.2xlarge	4	15	15
m4.4xlarge	8	30	30

Amazon Elastic Compute Cloud
User Guide for Linux Instances
IP addresses per network interface per instance type

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m4.10xlarge	8	30	30
m4.16xlarge	8	30	30
m5.large	3	10	10
m5.xlarge	4	15	15
m5.2xlarge	4	15	15
m5.4xlarge	8	30	30
m5.8xlarge	8	30	30
m5.12xlarge	8	30	30
m5.16xlarge	15	50	50
m5.24xlarge	15	50	50
m5.metal	15	50	50
m5a.large	3	10	10
m5a.xlarge	4	15	15
m5a.2xlarge	4	15	15
m5a.4xlarge	8	30	30
m5a.8xlarge	8	30	30
m5a.12xlarge	8	30	30
m5a.16xlarge	15	50	50
m5a.24xlarge	15	50	50
m5ad.large	3	10	10
m5ad.xlarge	4	15	15
m5ad.2xlarge	4	15	15
m5ad.4xlarge	8	30	30
m5ad.8xlarge	8	30	30
m5ad.12xlarge	8	30	30
m5ad.16xlarge	15	50	50
m5ad.24xlarge	15	50	50
m5d.large	3	10	10
m5d.xlarge	4	15	15
m5d.2xlarge	4	15	15

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m5d.4xlarge	8	30	30
m5d.8xlarge	8	30	30
m5d.12xlarge	8	30	30
m5d.16xlarge	15	50	50
m5d.24xlarge	15	50	50
m5d.metal	15	50	50
m5dn.large	3	10	10
m5dn.xlarge	4	15	15
m5dn.2xlarge	4	15	15
m5dn.4xlarge	8	30	30
m5dn.8xlarge	8	30	30
m5dn.12xlarge	8	30	30
m5dn.16xlarge	15	50	50
m5dn.24xlarge	15	50	50
m5n.large	3	10	10
m5n.xlarge	4	15	15
m5n.2xlarge	4	15	15
m5n.4xlarge	8	30	30
m5n.8xlarge	8	30	30
m5n.12xlarge	8	30	30
m5n.16xlarge	15	50	50
m5n.24xlarge	15	50	50
m6g.medium	2	4	4
m6g.large	3	10	10
m6g.xlarge	4	15	15
m6g.2xlarge	4	15	15
m6g.4xlarge	8	30	30
m6g.8xlarge	8	30	30
m6g.12xlarge	8	30	30
m6g.16xlarge	15	50	50

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
m6g.metal	15	50	50
m6gd.medium	2	4	4
m6gd.large	3	10	10
m6gd.xlarge	4	15	15
m6gd.2xlarge	4	15	15
m6gd.4xlarge	8	30	30
m6gd.8xlarge	8	30	30
m6gd.12xlarge	8	30	30
m6gd.16xlarge	15	50	50
m6gd.metal	15	50	50
p2.xlarge	4	15	15
p2.8xlarge	8	30	30
p2.16xlarge	8	30	30
p3.2xlarge	4	15	15
p3.8xlarge	8	30	30
p3.16xlarge	8	30	30
p3dn.24xlarge	15	50	50
r3.large	3	10	10
r3.xlarge	4	15	15
r3.2xlarge	4	15	15
r3.4xlarge	8	30	30
r3.8xlarge	8	30	30
r4.large	3	10	10
r4.xlarge	4	15	15
r4.2xlarge	4	15	15
r4.4xlarge	8	30	30
r4.8xlarge	8	30	30
r4.16xlarge	15	50	50
r5.large	3	10	10
r5.xlarge	4	15	15

Amazon Elastic Compute Cloud
User Guide for Linux Instances
IP addresses per network interface per instance type

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r5.2xlarge	4	15	15
r5.4xlarge	8	30	30
r5.8xlarge	8	30	30
r5.12xlarge	8	30	30
r5.16xlarge	15	50	50
r5.24xlarge	15	50	50
r5.metal	15	50	50
r5a.large	3	10	10
r5a.xlarge	4	15	15
r5a.2xlarge	4	15	15
r5a.4xlarge	8	30	30
r5a.8xlarge	8	30	30
r5a.12xlarge	8	30	30
r5a.16xlarge	15	50	50
r5a.24xlarge	15	50	50
r5ad.large	3	10	10
r5ad.xlarge	4	15	15
r5ad.2xlarge	4	15	15
r5ad.4xlarge	8	30	30
r5ad.8xlarge	8	30	30
r5ad.12xlarge	8	30	30
r5ad.16xlarge	15	50	50
r5ad.24xlarge	15	50	50
r5d.large	3	10	10
r5d.xlarge	4	15	15
r5d.2xlarge	4	15	15
r5d.4xlarge	8	30	30
r5d.8xlarge	8	30	30
r5d.12xlarge	8	30	30
r5d.16xlarge	15	50	50

Amazon Elastic Compute Cloud
 User Guide for Linux Instances
 IP addresses per network interface per instance type

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r5d.24xlarge	15	50	50
r5d.metal	15	50	50
r5dn.large	3	10	10
r5dn.xlarge	4	15	15
r5dn.2xlarge	4	15	15
r5dn.4xlarge	8	30	30
r5dn.8xlarge	8	30	30
r5dn.12xlarge	8	30	30
r5dn.16xlarge	15	50	50
r5dn.24xlarge	15	50	50
r5n.large	3	10	10
r5n.xlarge	4	15	15
r5n.2xlarge	4	15	15
r5n.4xlarge	8	30	30
r5n.8xlarge	8	30	30
r5n.12xlarge	8	30	30
r5n.16xlarge	15	50	50
r5n.24xlarge	15	50	50
r6g.medium	2	4	4
r6g.large	3	10	10
r6g.xlarge	4	15	15
r6g.2xlarge	4	15	15
r6g.4xlarge	8	30	30
r6g.8xlarge	8	30	30
r6g.12xlarge	8	30	30
r6g.16xlarge	15	50	50
r6g.metal	15	50	50
r6gd.medium	2	4	4
r6gd.large	3	10	10
r6gd.xlarge	4	15	15

Amazon Elastic Compute Cloud
User Guide for Linux Instances
IP addresses per network interface per instance type

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
r6gd.2xlarge	4	15	15
r6gd.4xlarge	8	30	30
r6gd.8xlarge	8	30	30
r6gd.12xlarge	8	30	30
r6gd.16xlarge	15	50	50
r6gd.metal	15	50	50
t1.micro	2	2	IPv6 not supported
t2.nano	2	2	2
t2.micro	2	2	2
t2.small	3	4	4
t2.medium	3	6	6
t2.large	3	12	12
t2.xlarge	3	15	15
t2.2xlarge	3	15	15
t3.nano	2	2	2
t3.micro	2	2	2
t3.small	3	4	4
t3.medium	3	6	6
t3.large	3	12	12
t3.xlarge	4	15	15
t3.2xlarge	4	15	15
t3a.nano	2	2	2
t3a.micro	2	2	2
t3a.small	2	4	4
t3a.medium	3	6	6
t3a.large	3	12	12
t3a.xlarge	4	15	15
t3a.2xlarge	4	15	15
t4g.nano	2	2	2
t4g.micro	2	2	2

Instance type	Maximum network interfaces	Private IPv4 addresses per interface	IPv6 addresses per interface
t4g.small	2	4	4
t4g.medium	3	6	6
t4g.large	3	12	12
t4g.xlarge	4	15	15
t4g.2xlarge	4	15	15
u-6tb1.metal	5	30	30
u-9tb1.metal	5	30	30
u-12tb1.metal	5	30	30
u-18tb1.metal	15	50	50
u-24tb1.metal	15	50	50
x1.16xlarge	8	30	30
x1.32xlarge	8	30	30
xle.xlarge	3	10	10
xle.2xlarge	4	15	15
xle.4xlarge	4	15	15
xle.8xlarge	4	15	15
xle.16xlarge	8	30	30
xle.32xlarge	8	30	30
z1d.large	3	10	10
z1d.xlarge	4	15	15
z1d.2xlarge	4	15	15
z1d.3xlarge	8	30	30
z1d.6xlarge	8	30	30
z1d.12xlarge	15	50	50
z1d.metal	15	50	50

Working with network interfaces

You can work with network interfaces using the Amazon EC2 console or the command line.

Contents

- [Creating a network interface \(p. 803\)](#)
- [Viewing details about a network interface \(p. 803\)](#)

- [Attaching a network interface to an instance \(p. 804\)](#)
- [Detaching a network interface from an instance \(p. 805\)](#)
- [Managing IP addresses \(p. 805\)](#)
- [Modifying network interface attributes \(p. 807\)](#)
- [Adding or editing tags \(p. 808\)](#)
- [Deleting a network interface \(p. 808\)](#)

Creating a network interface

You can create a network interface in a subnet. You can't move the network interface to another subnet after it's created, and you can only attach the network interface to instances in the same Availability Zone.

To create a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Choose **Create Network Interface**.
4. For **Description**, enter a descriptive name.
5. For **Subnet**, select the subnet.
6. For **Private IP (or IPv4 Private IP)**, enter the primary private IPv4 address. If you don't specify an IPv4 address, we select an available private IPv4 address from within the selected subnet.
7. (IPv6 only) If you selected a subnet that has an associated IPv6 CIDR block, you can optionally specify an IPv6 address in the **IPv6 IP** field.
8. To create an Elastic Fabric Adapter, select **Elastic Fabric Adapter**.
9. For **Security groups**, select one or more security groups.
10. (Optional) Choose **Add Tag** and enter a tag key and a tag value.
11. Choose **Yes, Create**.

To create a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [create-network-interface \(AWS CLI\)](#)
- [New-EC2NetworkInterface \(AWS Tools for Windows PowerShell\)](#)

Viewing details about a network interface

You can view all the network interfaces in your account.

To describe a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface.
4. To view the details, choose **Details**.

To describe a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-network-interfaces](#) (AWS CLI)
- [Get-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

To describe a network interface attribute using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-network-interface-attribute](#) (AWS CLI)
- [Get-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

Attaching a network interface to an instance

You can attach a network interface to any of your stopped or running instances, using either the **Instances** or **Network Interfaces** pages of the Amazon EC2 console. Alternatively, you can specify an existing network interface or attach an additional network interface when you [launch an instance \(p. 499\)](#).

If the public IPv4 address on your instance is released, it does not receive a new one if there is more than one network interface attached to the instance. For more information about the behavior of public IPv4 addresses, see [Public IPv4 addresses and external DNS hostnames \(p. 760\)](#).

To attach a network interface to an instance using the Instances page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. Choose **Actions, Networking, Attach network interface**.
5. Select the network interface and choose **Attach**.

To attach a network interface to an instance using the Network Interfaces page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and choose **Attach**.
4. Select the instance and choose **Attach**.

To attach a network interface to an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [attach-network-interface](#) (AWS CLI)
- [Add-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Detaching a network interface from an instance

You can detach a secondary network interface that is attached to an EC2 instance at any time, using either the **Instances** or **Network Interfaces** page of the Amazon EC2 console.

To detach a network interface from an instance using the Instances page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. Choose **Actions, Networking, Detach network interface**.
5. Select the network interface and choose **Detach**.

You can't use the Amazon EC2 console to detach a network interface that is attached to a resource from another service, such as an Elastic Load Balancing load balancer, a Lambda function, a WorkSpace, or a NAT gateway. The network interfaces for those resources are deleted when the resource is deleted.

To detach a network interface from an instance using the Network Interfaces page

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and check the description to verify that the network interface is attached to an instance, not another type of resource. If the resource is an EC2 instance, choose **Detach**.

If the network interface is the primary network interface for the instance, the **Detach** button is disabled.
4. When prompted for confirmation, choose **Yes, Detach**.
5. If the network interface fails to detach from the instance, choose **Force detachment** and then try again. We recommend that you choose this option only as a last resort. Forcing a detachment can prevent you from attaching a different network interface on the same index until you restart the instance. It can also prevent the instance metadata from reflecting that the network interface was detached until you restart the instance.

To detach a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [detach-network-interface](#) (AWS CLI)
- [Dismount-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Managing IP addresses

You can manage the following IP addresses for your network interfaces:

- Elastic IP addresses (one per private IPv4 address)
- IPv4 addresses
- IPv6 addresses

To Elastic IP addresses of a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface.
4. To associate an Elastic IP address, do the following:
 - a. Choose **Actions, Associate Address**.
 - b. For **Address**, select the Elastic IP address.
 - c. For **Associate to private IP address**, select the private IPv4 address to associate with the Elastic IP address.
 - d. Choose **Allow reassociation** to allow the Elastic IP address to be associated with the specified network interface if it's currently associated with another instance or network interface, and then choose **Associate Address**.
5. To disassociate an Elastic IP address, do the following:
 - a. Choose **Actions, Disassociate Address**.
 - b. In the **Disassociate IP Address** dialog box, choose **Yes, Disassociate**.

To manage the IPv4 and IPv6 addresses of a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface.
4. Choose **Actions, Manage IP Addresses**.
5. For **IPv4 Addresses**, modify the IP addresses as needed. To assign an IPv4 address, choose **Assign new IP** and then specify an IPv4 address from the subnet range or let AWS choose one for you. To unassign an IPv4 address, choose **Unassign** next to the address.
6. For **IPv6 Addresses**, modify the IP addresses as needed. To assign an IPv6 address, choose **Assign new IP** and then specify an IPv6 address from the subnet range or let AWS choose one for you. To unassign an IPv6 address, choose **Unassign** next to the address.
7. Choose **Yes, Update**.

To manage the IP addresses of a network interface using the AWS CLI

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [assign-ipv6-addresses](#)
- [associate-address](#)
- [disassociate-address](#)
- [unassign-ipv6-addresses](#)

To manage the IP addresses of a network interface using the Tools for Windows PowerShell

You can use one of the following commands.

- [Register-EC2Address](#)
- [Register-EC2Ipv6AddressList](#)
- [Unregister-EC2Address](#)
- [Unregister-EC2Ipv6AddressList](#)

Modifying network interface attributes

You can change the following network interface attributes:

- [Description \(p. 807\)](#)
- [Security groups \(p. 807\)](#)
- [Delete on termination \(p. 807\)](#)
- [Source/destination check \(p. 807\)](#)

To change the description of a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and choose **Actions, Change Description**.
4. For **Change Description**, enter a description for the network interface, and then choose **Save**.

To change the security groups of a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and choose **Actions, Change Security Groups**.
4. For **Change Security Groups**, select the security groups to use, and then choose **Save**.

The security group and network interface must be created for the same VPC. To change the security group for interfaces owned by other services, such as Elastic Load Balancing, do so through that service.

To change the termination behavior of a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and choose **Actions, Change Termination Behavior**.
4. In the **Change Termination Behavior** dialog box, select the **Delete on termination** check box if you want the network interface to be deleted when you terminate an instance.

To change source/destination checking for a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and choose **Actions, Change Source/Dest Check**.
4. In the dialog box, choose **Enabled** (if enabling) or **Disabled** (if disabling), and **Save**.

To modify network interface attributes using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [modify-network-interface-attribute \(AWS CLI\)](#)

- [Edit-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

Adding or editing tags

Tags are metadata that you can add to a network interface. Tags are private and are only visible to your account. Each tag consists of a key and an optional value. For more information about tags, see [Tagging your Amazon EC2 resources \(p. 1229\)](#).

To add or edit tags for a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface.
4. In the details pane, choose **Tags, Add/Edit Tags**.
5. In the **Add/Edit Tags** dialog box, choose **Create Tag** for each tag to create, and enter a key and optional value. When you're done, choose **Save**.

To add or edit tags for a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [create-tags](#) (AWS CLI)
- [New-EC2Tag](#) (AWS Tools for Windows PowerShell)

Deleting a network interface

To delete an instance, you must first detach the network interface. Deleting a network interface releases all attributes associated with the interface and releases any private IP addresses or Elastic IP addresses to be used by another instance.

To delete a network interface using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select a network interface and choose **Delete**.
4. In the **Delete Network Interface** dialog box, choose **Yes, Delete**.

To delete a network interface using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [delete-network-interface](#) (AWS CLI)
- [Remove-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Scenarios for network interfaces

Attaching multiple network interfaces to an instance is useful when you want to:

- Create a management network.
- Use network and security appliances in your VPC.

- Create dual-homed instances with workloads/roles on distinct subnets.
- Create a low-budget, high-availability solution.

Creating a management network

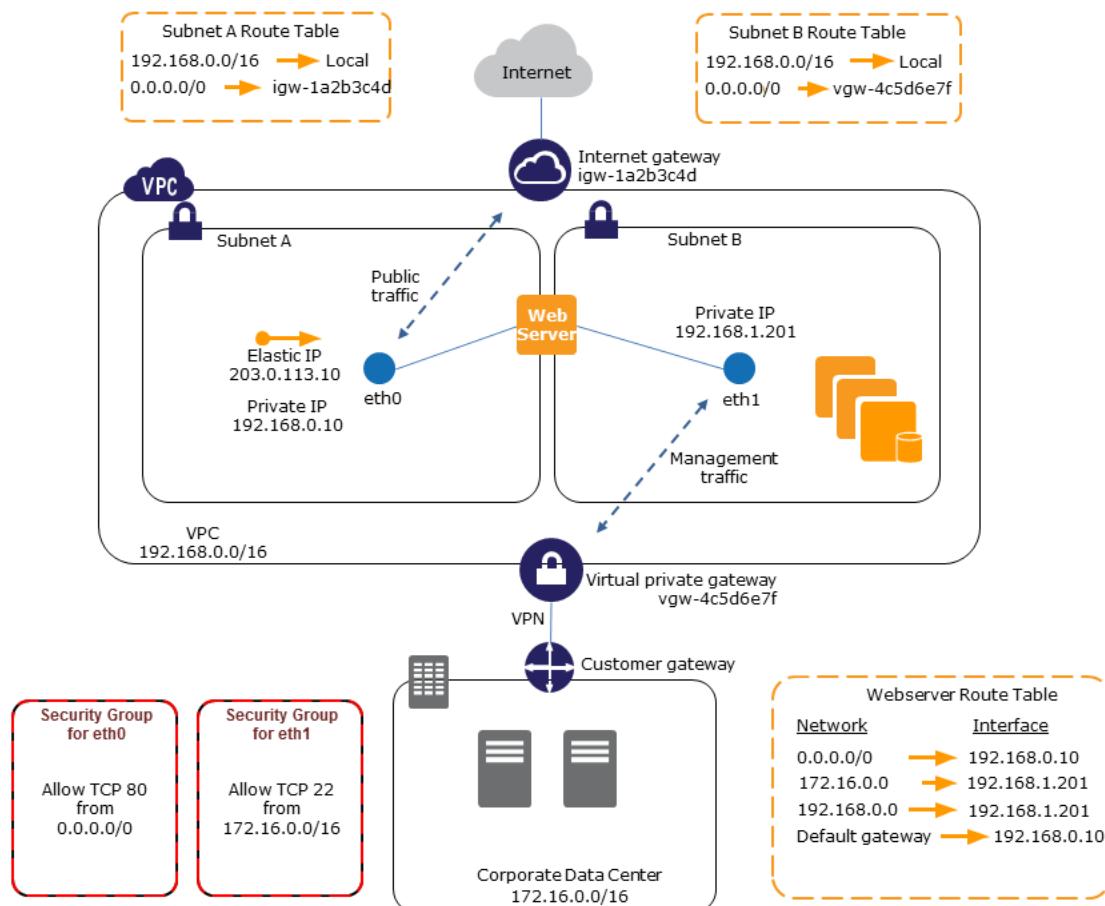
You can create a management network using network interfaces. In this scenario, as illustrated in the following image:

- The primary network interface (eth0) on the instance handles public traffic.
- The secondary network interface (eth1) handles backend management traffic, and is connected to a separate subnet in your VPC that has more restrictive access controls.

The public interface, which may or may not be behind a load balancer, has an associated security group that allows access to the server from the internet (for example, allow TCP port 80 and 443 from 0.0.0.0/0, or from the load balancer).

The private facing interface has an associated security group allowing SSH access only from an allowed range of IP addresses, either within the VPC, or from the internet, a private subnet within the VPC, or a virtual private gateway.

To ensure failover capabilities, consider using a secondary private IPv4 for incoming traffic on a network interface. In the event of an instance failure, you can move the interface and/or secondary private IPv4 address to a standby instance.



Use network and security appliances in your VPC

Some network and security appliances, such as load balancers, network address translation (NAT) servers, and proxy servers prefer to be configured with multiple network interfaces. You can create and attach secondary network interfaces to instances in a VPC that are running these types of applications and configure the additional interfaces with their own public and private IP addresses, security groups, and source/destination checking.

Creating dual-homed instances with workloads/roles on distinct subnets

You can place a network interface on each of your web servers that connects to a mid-tier network where an application server resides. The application server can also be dual-homed to a backend network (subnet) where the database server resides. Instead of routing network packets through the dual-homed instances, each dual-homed instance receives and processes requests on the front end, initiates a connection to the backend, and then sends requests to the servers on the backend network.

Create a low budget high availability solution

If one of your instances serving a particular function fails, its network interface can be attached to a replacement or hot standby instance pre-configured for the same role in order to rapidly recover the service. For example, you can use a network interface as your primary or secondary network interface to a critical service such as a database instance or a NAT instance. If the instance fails, you (or more likely, the code running on your behalf) can attach the network interface to a hot standby instance. Because the interface maintains its private IP addresses, Elastic IP addresses, and MAC address, network traffic begins flowing to the standby instance as soon as you attach the network interface to the replacement instance. Users experience a brief loss of connectivity between the time the instance fails and the time that the network interface is attached to the standby instance, but no changes to the VPC route table or your DNS server are required.

Best practices for configuring network interfaces

- You can attach a network interface to an instance when it's running (hot attach), when it's stopped (warm attach), or when the instance is being launched (cold attach).
- You can detach secondary network interfaces when the instance is running or stopped. However, you can't detach the primary network interface.
- You can move a network interface from one instance to another, if the instances are in the same Availability Zone and VPC but in different subnets.
- When launching an instance using the CLI, API, or an SDK, you can specify the primary network interface and additional network interfaces.
- Launching an Amazon Linux or Windows Server instance with multiple network interfaces automatically configures interfaces, private IPv4 addresses, and route tables on the operating system of the instance.
- A warm or hot attach of an additional network interface may require you to manually bring up the second interface, configure the private IPv4 address, and modify the route table accordingly. Instances running Amazon Linux or Windows Server automatically recognize the warm or hot attach and configure themselves.
- Attaching another network interface to an instance (for example, a NIC teaming configuration) cannot be used as a method to increase or double the network bandwidth to or from the dual-homed instance.
- If you attach two or more network interfaces from the same subnet to an instance, you might encounter networking issues such as asymmetric routing. If possible, use a secondary private IPv4 address on the primary network interface instead.

Configuring your network interface using ec2-net-utils

Amazon Linux AMIs may contain additional scripts installed by AWS, known as ec2-net-utils. These scripts optionally automate the configuration of your network interfaces. These scripts are available for Amazon Linux only.

Use the following command to install the package on Amazon Linux if it's not already installed, or update it if it's installed and additional updates are available:

```
$ yum install ec2-net-utils
```

The following components are part of ec2-net-utils:

udev rules (`/etc/udev/rules.d`)

Identifies network interfaces when they are attached, detached, or reattached to a running instance, and ensures that the hotplug script runs (`53-ec2-network-interfaces.rules`). Maps the MAC address to a device name (`75-persistent-net-generator.rules`, which generates `70-persistent-net.rules`).

hotplug script

Generates an interface configuration file suitable for use with DHCP (`/etc/sysconfig/network-scripts/ifcfg-ethN`). Also generates a route configuration file (`/etc/sysconfig/network-scripts/route-ethN`).

DHCP script

Whenever the network interface receives a new DHCP lease, this script queries the instance metadata for Elastic IP addresses. For each Elastic IP address, it adds a rule to the routing policy database to ensure that outbound traffic from that address uses the correct network interface. It also adds each private IP address to the network interface as a secondary address.

ec2ifup ethN

Extends the functionality of the standard **ifup**. After this script rewrites the configuration files `ifcfg-ethN` and `route-ethN`, it runs **ifup**.

ec2ifdown ethN

Extends the functionality of the standard **ifdown**. After this script removes any rules for the network interface from the routing policy database, it runs **ifdown**.

ec2ifscan

Checks for network interfaces that have not been configured and configures them.

This script isn't available in the initial release of ec2-net-utils.

To list any configuration files that were generated by ec2-net-utils, use the following command:

```
$ ls -l /etc/sysconfig/network-scripts/*-eth?
```

To disable the automation on a per-instance basis, you can add `EC2SYNC=no` to the corresponding `ifcfg-ethN` file. For example, use the following command to disable the automation for the `eth1` interface:

```
$ sed -i -e 's/^EC2SYNC=yes/EC2SYNC=no/' /etc/sysconfig/network-scripts/ifcfg-eth1
```

To disable the automation completely, you can remove the package using the following command:

```
$ yum remove ec2-net-utils
```

Requester-managed network interfaces

A requester-managed network interface is a network interface that an AWS service creates in your VPC. This network interface can represent an instance for another service, such as an Amazon RDS instance, or it can enable you to access another service or resource, such as an AWS PrivateLink service, or an Amazon ECS task.

You cannot modify or detach a requester-managed network interface. If you delete the resource that the network interface represents, the AWS service detaches and deletes the network interface for you. To change the security groups for a requester-managed network interface, you might have to use the console or command line tools for that service. For more information, see the service-specific documentation.

You can tag a requester-managed network interface. For more information, see [Adding or editing tags \(p. 808\)](#).

You can view the requester-managed network interfaces that are in your account.

To view requester-managed network interfaces using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface and view the following information on the details pane:
 - **Attachment owner:** If you created the network interface, this field displays your AWS account ID. Otherwise, it displays an alias or ID for the principal or service that created the network interface.
 - **Description:** Provides information about the purpose of the network interface; for example, "VPC Endpoint Interface".

To view requester-managed network interfaces using the command line

1. Use the [describe-network-interfaces](#) AWS CLI command to describe the network interfaces in your account.

```
aws ec2 describe-network-interfaces
```

2. In the output, the RequesterManaged field displays true if the network interface is managed by another AWS service.

```
{
    "Status": "in-use",
    ...
    "Description": "VPC Endpoint Interface vpce-089f2123488812123",
    "NetworkInterfaceId": "eni-c8fbcc27e",
    "VpcId": "vpc-1a2b3c4d",
    "PrivateIpAddresses": [
        {
            "PrivateDnsName": "ip-10-0-2-227.ec2.internal",
            "Primary": true,
            "PrivateIpAddress": "10.0.2.227"
        }
    ],
    "RequesterManaged": true,
```

...
}

Alternatively, use the [Get-EC2NetworkInterface](#) Tools for Windows PowerShell command.

Enhanced networking on Linux

Enhanced networking uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on [supported instance types \(p. 813\)](#). SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization when compared to traditional virtualized network interfaces. Enhanced networking provides higher bandwidth, higher packet per second (PPS) performance, and consistently lower inter-instance latencies. There is no additional charge for using enhanced networking.

For information about the supported network speed for each instance type, see [Amazon EC2 Instance Types](#).

Contents

- [Enhanced networking support \(p. 813\)](#)
- [Enabling enhanced networking on your instance \(p. 813\)](#)
- [Enabling enhanced networking with the Elastic Network Adapter \(ENA\) on Linux instances \(p. 814\)](#)
- [Enabling enhanced networking with the Intel 82599 VF interface on Linux instances \(p. 826\)](#)
- [Troubleshooting the Elastic Network Adapter \(ENA\) \(p. 832\)](#)

Enhanced networking support

All [current generation \(p. 201\)](#) instance types support enhanced networking, except for T2 instances.

You can enable enhanced networking using one of the following mechanisms:

Elastic Network Adapter (ENA)

The Elastic Network Adapter (ENA) supports network speeds of up to 100 Gbps for supported instance types.

The current generation instances use ENA for enhanced networking, except for C4, D2, and M4 instances smaller than `m4.16xlarge`.

Intel 82599 Virtual Function (VF) interface

The Intel 82599 Virtual Function interface supports network speeds of up to 10 Gbps for supported instance types.

The following instance types use the Intel 82599 VF interface for enhanced networking: C3, C4, D2, I2, M4 (excluding `m4.16xlarge`), and R3.

For a summary of the enhanced networking mechanisms by instance type, see [Summary of networking and storage features \(p. 206\)](#).

Enabling enhanced networking on your instance

If your instance type supports the Elastic Network Adapter for enhanced networking, follow the procedures in [Enabling enhanced networking with the Elastic Network Adapter \(ENA\) on Linux instances \(p. 814\)](#).

If your instance type supports the Intel 82599 VF interface for enhanced networking, follow the procedures in [Enabling enhanced networking with the Intel 82599 VF interface on Linux instances \(p. 826\)](#).

Enabling enhanced networking with the Elastic Network Adapter (ENA) on Linux instances

Amazon EC2 provides enhanced networking capabilities through the Elastic Network Adapter (ENA). To use enhanced networking, you must install the required ENA module and enable ENA support.

Contents

- [Requirements \(p. 814\)](#)
- [Enhanced networking performance \(p. 814\)](#)
- [Testing whether enhanced networking is enabled \(p. 815\)](#)
- [Enabling enhanced networking on the Amazon Linux AMI \(p. 817\)](#)
- [Enabling enhanced networking on Ubuntu \(p. 818\)](#)
- [Enabling enhanced networking on Linux \(p. 819\)](#)
- [Enabling enhanced networking on Ubuntu with DKMS \(p. 821\)](#)
- [Troubleshooting \(p. 823\)](#)
- [Operating system optimizations \(p. 823\)](#)

Requirements

To prepare for enhanced networking using the ENA, set up your instance as follows:

- Launch the instance using a [current generation \(p. 201\)](#) instance type, other than C4, D2, M4 instances smaller than `m4.16xlarge`, or T2.
- Launch the instance using a supported version of the Linux kernel and a supported distribution, so that ENA enhanced networking is enabled for your instance automatically. For more information, see [ENA Linux Kernel Driver Release Notes](#).
- Ensure that the instance has internet connectivity.
- Install and configure the [AWS CLI](#) or the [AWS Tools for Windows PowerShell](#) on any computer you choose, preferably your local desktop or laptop. For more information, see [Accessing Amazon EC2 \(p. 3\)](#). Enhanced networking cannot be managed from the Amazon EC2 console.
- If you have important data on the instance that you want to preserve, you should back that data up now by creating an AMI from your instance. Updating kernels and kernel modules, as well as enabling the `enaSupport` attribute, might render incompatible instances or operating systems unreachable. If you have a recent backup, your data will still be retained if this happens.

Enhanced networking performance

The following documentation provides a summary of the network performance for the instance types that support ENA enhanced networking:

- [Network Performance for Accelerated Computing Instances \(p. 282\)](#)
- [Network Performance for Compute Optimized Instances \(p. 256\)](#)
- [Network Performance for General Purpose Instances \(p. 214\)](#)
- [Network Performance for Memory Optimized Instances \(p. 266\)](#)

- [Network Performance for Storage Optimized Instances \(p. 274\)](#)

Testing whether enhanced networking is enabled

The following AMIs include the required ENA module and have ENA support enabled:

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (with `linux-aws` kernel) or later
- Red Hat Enterprise Linux 7.4 or later
- SUSE Linux Enterprise Server 12 SP2 or later
- CentOS 7.4.1708 or later
- FreeBSD 11.1 or later
- Debian GNU/Linux 9 or later

To test whether enhanced networking is already enabled, verify that the `ena` module is installed on your instance and that the `enaSupport` attribute is set. If your instance satisfies these two conditions, then the `ethtool -i ethn` command should show that the module is in use on the network interface.

Kernel module (`ena`)

To verify that the `ena` module is installed, use the `modinfo` command as shown in the following example.

```
[ec2-user ~]$ modinfo ena
filename:      /lib/modules/4.14.33-59.37.amzn2.x86_64/kernel/drivers/amazon/net/ena/
ena.ko
version:       1.5.0g
license:        GPL
description:   Elastic Network Adapter (ENA)
author:         Amazon.com, Inc. or its affiliates
srcversion:    692C7C68B8A9001CB3F31D0
alias:          pci:v00001D0Fd0000EC21sv*sd*bc*sc*i*
alias:          pci:v00001D0Fd0000EC20sv*sd*bc*sc*i*
alias:          pci:v00001D0Fd00001EC2sv*sd*bc*sc*i*
alias:          pci:v00001D0Fd00000EC2sv*sd*bc*sc*i*
depends:
retpoline:     Y
intree:        Y
name:          ena
...
```

In the above Amazon Linux case, the `ena` module is installed.

```
ubuntu:~$ modinfo ena
ERROR: modinfo: could not find module ena
```

In the above Ubuntu instance, the module is not installed, so you must first install it. For more information, see [Enabling enhanced networking on Ubuntu \(p. 818\)](#).

Instance attribute (`enaSupport`)

To check whether an instance has the enhanced networking `enaSupport` attribute set, use one of the following commands. If the attribute is set, the response is true.

- [describe-instances \(AWS CLI\)](#)

```
aws ec2 describe-instances --instance-ids instance_id --query  
"Reservations[].[Instances[]].EnaSupport"
```

- [Get-EC2Instance \(Tools for Windows PowerShell\)](#)

```
(Get-EC2Instance -InstanceId instance-id).Instances.EnaSupport
```

Image attribute (enaSupport)

To check whether an AMI has the enhanced networking enaSupport attribute set, use one of the following commands. If the attribute is set, the response is true.

- [describe-images \(AWS CLI\)](#)

```
aws ec2 describe-images --image-id ami_id --query "Images[].[EnaSupport]"
```

- [Get-EC2Image \(Tools for Windows PowerShell\)](#)

```
(Get-EC2Image -ImageId ami_id).EnaSupport
```

Network interface driver

Use the following command to verify that the ena module is being used on a particular interface, substituting the interface name that you want to check. If you are using a single interface (default), it this is eth0. If the operating system supports [predictable network names \(p. 820\)](#), this could be a name like ens5.

In the following example, the ena module is not loaded, because the listed driver is vif.

```
[ec2-user ~]$ ethtool -i eth0  
driver: vif  
version:  
firmware-version:  
bus-info: vif-0  
supports-statistics: yes  
supports-test: no  
supports-eeprom-access: no  
supports-register-dump: no  
supports-priv-flags: no
```

In this example, the ena module is loaded and at the minimum recommended version. This instance has enhanced networking properly configured.

```
[ec2-user ~]$ ethtool -i eth0  
driver: ena  
version: 1.5.0g  
firmware-version:  
expansion-rom-version:  
bus-info: 0000:00:05.0  
supports-statistics: yes  
supports-test: no  
supports-eeprom-access: no  
supports-register-dump: no  
supports-priv-flags: no
```

Enabling enhanced networking on the Amazon Linux AMI

Amazon Linux 2 and the latest versions of the Amazon Linux AMI include the module required for enhanced networking with ENA installed and have ENA support enabled. Therefore, if you launch an instance with an HVM version of Amazon Linux on a supported instance type, enhanced networking is already enabled for your instance. For more information, see [Testing whether enhanced networking is enabled \(p. 815\)](#).

If you launched your instance using an older Amazon Linux AMI and it does not have enhanced networking enabled already, use the following procedure to enable enhanced networking.

To enable enhanced networking on Amazon Linux AMI

1. Connect to your instance.
2. From the instance, run the following command to update your instance with the newest kernel and kernel modules, including ena:

```
[ec2-user ~]$ sudo yum update
```

3. From your local computer, reboot your instance using the Amazon EC2 console or one of the following commands: [reboot-instances](#) (AWS CLI), [Restart-EC2Instance](#) (AWS Tools for Windows PowerShell).
4. Connect to your instance again and verify that the ena module is installed and at the minimum recommended version using the [modinfo ena](#) command from [Testing whether enhanced networking is enabled \(p. 815\)](#).
5. [EBS-backed instance] From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: [stop-instances](#) (AWS CLI), [Stop-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

[Instance store-backed instance] You can't stop the instance to modify the attribute. Instead, proceed to this procedure: [To enable enhanced networking on Amazon Linux AMI \(instance store-backed instances\) \(p. 818\)](#).

6. From your local computer, enable the enhanced networking attribute using one of the following commands:

- [modify-instance-attribute](#) (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

- [Edit-EC2InstanceAttribute](#) (Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -EnaSupport $true
```

7. (Optional) Create an AMI from the instance, as described in [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#). The AMI inherits the enhanced networking enaSupport attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.
8. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: [start-instances](#) (AWS CLI), [Start-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
9. Connect to your instance and verify that the ena module is installed and loaded on your network interface using the [ethtool -i ethn](#) command from [Testing whether enhanced networking is enabled \(p. 815\)](#).

If you are unable to connect to your instance after enabling enhanced networking, see [Troubleshooting the Elastic Network Adapter \(ENA\) \(p. 832\)](#).

To enable enhanced networking on Amazon Linux AMI (instance store-backed instances)

Follow the previous procedure until the step where you stop the instance. Create a new AMI as described in [Creating an instance store-backed Linux AMI \(p. 127\)](#), making sure to enable the enhanced networking attribute when you register the AMI.

- [register-image \(AWS CLI\)](#)

```
aws ec2 register-image --ena-support ...
```

- [Register-EC2Image \(AWS Tools for Windows PowerShell\)](#)

```
Register-EC2Image -EnaSupport $true ...
```

Enabling enhanced networking on Ubuntu

The latest Ubuntu HVM AMIs include the module required for enhanced networking with ENA installed and have ENA support enabled. Therefore, if you launch an instance with the latest Ubuntu HVM AMI on a supported instance type, enhanced networking is already enabled for your instance. For more information, see [Testing whether enhanced networking is enabled \(p. 815\)](#).

If you launched your instance using an older AMI and it does not have enhanced networking enabled already, you can install the `linux-aws` kernel package to get the latest enhanced networking drivers and update the required attribute.

To install the `linux-aws` kernel package (Ubuntu 16.04 or later)

Ubuntu 16.04 and 18.04 ship with the Ubuntu custom kernel (`linux-aws` kernel package). To use a different kernel, contact [AWS Support](#).

To install the `linux-aws` kernel package (Ubuntu Trusty 14.04)

1. Connect to your instance.
2. Update the package cache and packages.

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y linux-aws
```

Important

If during the update process you are prompted to install `grub`, use `/dev/xvda` to install `grub` onto, and then choose to keep the current version of `/boot/grub/menu.lst`.

3. [EBS-backed instance] From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: [stop-instances \(AWS CLI\)](#), [Stop-EC2Instance \(AWS Tools for Windows PowerShell\)](#). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

[Instance store-backed instance] You can't stop the instance to modify the attribute. Instead, proceed to this procedure: [To enable enhanced networking on Ubuntu \(instance store-backed instances\) \(p. 819\)](#).

4. From your local computer, enable the enhanced networking attribute using one of the following commands:

- [modify-instance-attribute \(AWS CLI\)](#)

```
aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

- [Edit-EC2InstanceAttribute \(Tools for Windows PowerShell\)](#)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -EnaSupport $true
```

5. (Optional) Create an AMI from the instance, as described in [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#). The AMI inherits the enhanced networking enaSupport attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.
6. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: [start-instances \(AWS CLI\)](#), [Start-EC2Instance \(AWS Tools for Windows PowerShell\)](#). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.

To enable enhanced networking on Ubuntu (instance store-backed instances)

Follow the previous procedure until the step where you stop the instance. Create a new AMI as described in [Creating an instance store-backed Linux AMI \(p. 127\)](#), making sure to enable the enhanced networking attribute when you register the AMI.

- [register-image \(AWS CLI\)](#)

```
aws ec2 register-image --ena-support ...
```

- [Register-EC2Image \(AWS Tools for Windows PowerShell\)](#)

```
Register-EC2Image -EnaSupport $true ...
```

Enabling enhanced networking on Linux

The latest AMIs for Red Hat Enterprise Linux, SUSE Linux Enterprise Server, and CentOS include the module required for enhanced networking with ENA and have ENA support enabled. Therefore, if you launch an instance with the latest AMI on a supported instance type, enhanced networking is already enabled for your instance. For more information, see [Testing whether enhanced networking is enabled \(p. 815\)](#).

The following procedure provides the general steps for enabling enhanced networking on a Linux distribution other than Amazon Linux AMI or Ubuntu. For more information, such as detailed syntax for commands, file locations, or package and tool support, see the documentation for your Linux distribution.

To enable enhanced networking on Linux

1. Connect to your instance.
2. Clone the source code for the ena module on your instance from GitHub at <https://github.com/amzn/amzn-drivers>. (SUSE Linux Enterprise Server 12 SP2 and later include ENA 2.02 by default, so you are not required to download and compile the ENA driver. For SUSE Linux Enterprise Server 12 SP2 and later, you should file a request to add the driver version you want to the stock kernel).

```
git clone https://github.com/amzn/amzn-drivers
```

3. Compile and install the ena module on your instance. These steps depend on the Linux distribution. For more information about compiling the module on Red Hat Enterprise Linux, see the [AWS Knowledge Center article](#).
4. Run the **sudo depmod** command to update module dependencies.
5. Update *initramfs* on your instance to ensure that the new module loads at boot time. For example, if your distribution supports **dracut**, you can use the following command.

```
dracut -f -v
```

6. Determine if your system uses predictable network interface names by default. Systems that use **systemd** or **udev** versions 197 or greater can rename Ethernet devices and they do not guarantee that a single network interface will be named *eth0*. This behavior can cause problems connecting to your instance. For more information and to see other configuration options, see [Predictable Network Interface Names](#) on the freedesktop.org website.
 - a. You can check the **systemd** or **udev** versions on RPM-based systems with the following command.

```
rpm -qa | grep -e '^systemd-[0-9]+\| ^udev-[0-9]+\'
systemd-208-11.el7_0.2.x86_64
```

In the above Red Hat Enterprise Linux 7 example, the **systemd** version is 208, so predictable network interface names must be disabled.

- b. Disable predictable network interface names by adding the *net.ifnames=0* option to the **GRUB_CMDLINE_LINUX** line in */etc/default/grub*.

```
sudo sed -i '/^GRUB_CMDLINE_LINUX/s/"$/ net.ifnames=0"/' /etc/default/grub
```

- c. Rebuild the grub configuration file.

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. [EBS-backed instance] From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: [stop-instances](#) (AWS CLI), [Stop-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

[Instance store-backed instance] You can't stop the instance to modify the attribute. Instead, proceed to this procedure: [To enable enhanced networking on Linux \(instance store-backed instances\) \(p. 821\)](#).

8. From your local computer, enable the enhanced networking *enaSupport* attribute using one of the following commands:

- [modify-instance-attribute](#) (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

- [Edit-EC2InstanceAttribute](#) (Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -EnaSupport $true
```

9. (Optional) Create an AMI from the instance, as described in [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#). The AMI inherits the enhanced networking *enaSupport* attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.

Important

If your instance operating system contains an `/etc/udev/rules.d/70-persistent-net.rules` file, you must delete it before creating the AMI. This file contains the MAC address for the Ethernet adapter of the original instance. If another instance boots with this file, the operating system will be unable to find the device and `eth0` might fail, causing boot issues. This file is regenerated at the next boot cycle, and any instances launched from the AMI create their own version of the file.

10. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: [start-instances \(AWS CLI\)](#), [Start-EC2Instance \(AWS Tools for Windows PowerShell\)](#). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
11. (Optional) Connect to your instance and verify that the module is installed.

If you are unable to connect to your instance after enabling enhanced networking, see [Troubleshooting the Elastic Network Adapter \(ENA\) \(p. 832\)](#).

To enable enhanced networking on Linux (instance store-backed instances)

Follow the previous procedure until the step where you stop the instance. Create a new AMI as described in [Creating an instance store-backed Linux AMI \(p. 127\)](#), making sure to enable the enhanced networking attribute when you register the AMI.

- [register-image \(AWS CLI\)](#)

```
aws ec2 register-image --ena-support ...
```

- [Register-EC2Image \(AWS Tools for Windows PowerShell\)](#)

```
Register-EC2Image -EnaSupport ...
```

Enabling enhanced networking on Ubuntu with DKMS

This method is for testing and feedback purposes only. It is not intended for use with production deployments. For production deployments, see [Enabling enhanced networking on Ubuntu \(p. 818\)](#).

Important

Using DKMS voids the support agreement for your subscription. It should not be used for production deployments.

To enable enhanced networking with ENA on Ubuntu (EBS-backed instances)

1. Follow steps 1 and 2 in [Enabling enhanced networking on Ubuntu \(p. 818\)](#).
2. Install the `build-essential` packages to compile the kernel module and the `dkms` package so that your `ena` module is rebuilt every time your kernel is updated.

```
ubuntu:~$ sudo apt-get install -y build-essential dkms
```

3. Clone the source for the `ena` module on your instance from GitHub at <https://github.com/amzn/amzn-drivers>.

```
ubuntu:~$ git clone https://github.com/amzn/amzn-drivers
```

4. Move the `amzn-drivers` package to the `/usr/src/` directory so DKMS can find it and build it for each kernel update. Append the version number (you can find the current version number in the

release notes) of the source code to the directory name. For example, version 1.0.0 is shown in the following example.

```
ubuntu:~$ sudo mv amzn-drivers /usr/src/amzn-drivers-1.0.0
```

5. Create the DKMS configuration file with the following values, substituting your version of ena.

Create the file.

```
ubuntu:~$ sudo touch /usr/src/amzn-drivers-1.0.0/dkms.conf
```

Edit the file and add the following values.

```
ubuntu:~$ sudo vim /usr/src/amzn-drivers-1.0.0/dkms.conf
PACKAGE_NAME="ena"
PACKAGE_VERSION="1.0.0"
CLEAN="make -C kernel/linux/ena clean"
MAKE="make -C kernel/linux/ena/ BUILD_KERNEL=${kernelver}"
BUILT_MODULE_NAME[0]="ena"
BUILT_MODULE_LOCATION="kernel/linux/ena"
DEST_MODULE_LOCATION[0]="/updates"
DEST_MODULE_NAME[0]="ena"
AUTOINSTALL="yes"
```

6. Add, build, and install the ena module on your instance using DKMS.

Add the module to DKMS.

```
ubuntu:~$ sudo dkms add -m amzn-drivers -v 1.0.0
```

Build the module using the **dkms** command.

```
ubuntu:~$ sudo dkms build -m amzn-drivers -v 1.0.0
```

Install the module using **dkms**.

```
ubuntu:~$ sudo dkms install -m amzn-drivers -v 1.0.0
```

7. Rebuild initramfs so the correct module is loaded at boot time.

```
ubuntu:~$ sudo update-initramfs -u -k all
```

8. Verify that the ena module is installed using the modinfo ena command from [Testing whether enhanced networking is enabled \(p. 815\)](#).

```
ubuntu:~$ modinfo ena
filename:      /lib/modules/3.13.0-74-generic/updates/dkms/ena.ko
version:       1.0.0
license:        GPL
description:   Elastic Network Adapter (ENA)
author:        Amazon.com, Inc. or its affiliates
srcversion:    9693C876C54CA64AE48F0CA
alias:         pci:v00001D0Fd0000EC21sv*sd*bc*sc*i*
alias:         pci:v00001D0Fd0000EC20sv*sd*bc*sc*i*
alias:         pci:v00001D0Fd00001EC2sv*sd*bc*sc*i*
alias:         pci:v00001D0Fd00000EC2sv*sd*bc*sc*i*
depends:
vermagic:      3.13.0-74-generic SMP mod_unload modversions
```

```
parm:      debug:Debug level (0=none,...,16=all) (int)
parm:      push_mode:Descriptor / header push mode
(0=automatic,1=disable,3=enable)
  0 - Automatically choose according to device capability (default)
  1 - Don't push anything to device memory
  3 - Push descriptors and header buffer to device memory (int)
parm:      enable_wd:Enable keepalive watchdog (0=disable,1=enable,default=1)
(int)
parm:      enable_missing_tx_detection:Enable missing Tx completions. (default=1)
(int)
parm:      numa_node_override_array:Numa node override map
(array of int)
parm:      numa_node_override:Enable/Disable numa node override (0=disable)
(int)
```

9. Continue with Step 3 in [Enabling enhanced networking on Ubuntu \(p. 818\)](#).

Troubleshooting

For additional information about troubleshooting your ENA adapter, see [Troubleshooting the Elastic Network Adapter \(ENA\) \(p. 832\)](#).

Operating system optimizations

To achieve the maximum network performance on instances with enhanced networking, you may need to modify the default operating system configuration. We recommend the following configuration changes for applications that require high network performance.

In addition to these operating system optimizations, you should also consider the maximum transmission unit (MTU) of your network traffic, and adjust according to your workload and network architecture. For more information, see [Network maximum transmission unit \(MTU\) for your EC2 instance \(p. 881\)](#).

AWS regularly measures average round trip latencies between instances launched in a cluster placement group of 50us and tail latencies of 200us at the 99.9 percentile. If your applications require consistently low latencies, we recommend using the latest version of the ENA drivers on fixed performance Nitro-based instances.

These procedures were written for Amazon Linux 2 and Amazon Linux AMI. However, they may also work for other Linux distributions with kernel version 3.9 or newer. For more information, see your system-specific documentation.

To optimize your Amazon Linux instance for enhanced networking

1. Check the clock source for your instance:

```
cat /sys/devices/system/clocksource/clocksource0/current_clocksource
```

2. If the clock source is `xen`, complete the following substeps. Otherwise, skip to [Step 3 \(p. 824\)](#).

- a. Edit the GRUB configuration and add `xen_nopvspin=1` and `clocksource=tsc` to the kernel boot options.
 - For Amazon Linux 2, edit the `/etc/default/grub` file and add these options to the `GRUB_CMDLINE_LINUX_DEFAULT` line, as shown below:

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
biosdevname=0 nvme_core.io_timeout=4294967295 xen_nopvspin=1 clocksource=tsc"
GRUB_TIMEOUT=0
```

- For Amazon Linux AMI, edit the /boot/grub/grub.conf file and add these options to the kernel line, as shown below:

```
kernel /boot/vmlinuz-4.14.62-65.117.amzn1.x86_64 root=LABEL=/ console=tty1
       console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295 xen_nopvspin=1
       clocksource=tsc
```

- (Amazon Linux 2 only) Rebuild your GRUB configuration file to pick up these changes:

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

- If your instance type is listed as supported on [Processor state control for your EC2 instance \(p. 622\)](#), prevent the system from using deeper C-states to ensure low-latency system performance. For more information, see [High performance and low latency by limiting deeper C-states \(p. 624\)](#).
 - Edit the GRUB configuration and add intel_idle.max_cstate=1 to the kernel boot options.

- For Amazon Linux 2, edit the /etc/default/grub file and add this option to the GRUB_CMDLINE_LINUX_DEFAULT line, as shown below:

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8
                           net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 xen_nopvspin=1
                           clocksource=tsc intel_idle.max_cstate=1"
                           GRUB_TIMEOUT=0
```

- For Amazon Linux AMI, edit the /boot/grub/grub.conf file and add this option to the kernel line, as shown below:

```
kernel /boot/vmlinuz-4.14.62-65.117.amzn1.x86_64 root=LABEL=/ console=tty1
       console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295 xen_nopvspin=1
       clocksource=tsc intel_idle.max_cstate=1
```

- (Amazon Linux 2 only) Rebuild your GRUB configuration file to pick up these changes:

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

- Ensure that your reserved kernel memory is sufficient to sustain a high rate of packet buffer allocations (the default value may be too small).
 - Open (as root or with sudo) the /etc/sysctl.conf file with the editor of your choice.
 - Add the vm.min_free_kbytes line to the file with the reserved kernel memory value (in kilobytes) for your instance type. As a rule of thumb, you should set this value to between 1-3% of available system memory, and adjust this value up or down to meet the needs of your application.

```
vm.min_free_kbytes = 1048576
```

- Apply this configuration with the following command:

```
sudo sysctl -p
```

- Verify that the setting was applied with the following command:

```
sudo sysctl -a 2>&1 | grep min_free_kbytes
```

- Reboot your instance to load the new configuration:

```
sudo reboot
```

6. (Optional) Manually distribute packet receive interrupts so that they are associated with different CPUs that all belong to the same NUMA node. Use this carefully, however, because **irqbalancer** is disabled globally.

Note

The configuration change in this step does not survive a reboot.

- a. Create a file called **smp_affinity.sh** and paste the following code block into it:

```
#!/bin/sh
service irqbalance stop
affinity_values=(00000001 00000002 00000004 00000008 00000010 00000020 00000040
00000080)
irqs=($(grep eth /proc/interrupts|awk '{print $1}'|cut -d : -f 1))
irqLen=${#irqs[@]}
for (( i=0; i<${irqLen}; i++ )); do
    echo $(printf "0000,00000000,00000000,00000000,%s" ${affinity_values[$i]}) > /proc/
irq/${irqs[$i]}/smp_affinity;
    echo "IRQ ${irqs[$i]} =" $(cat /proc/irq/${irqs[$i]}/smp_affinity);
done
```

- b. Run the script with the following command:

```
sudo bash ./smp_affinity.sh
```

7. (Optional) If the vCPUs that handle receive IRQs are overloaded, or if your application network processing is demanding on CPU, you can offload part of the network processing to other cores with receive packet steering (RPS). Ensure that cores used for RPS belong to the same NUMA node to avoid inter-NUMA node locks. For example, to use cores 8-15 for packet processing, use the following command.

Note

The configuration change in this step does not survive a reboot.

```
for i in `seq 0 7`; do echo $(printf "0000,00000000,00000000,00000000,0000ff00") | sudo
tee /sys/class/net/eth0/queues/rx-$i/rps_cpus; done
```

8. (Optional) If possible, keep all processing on the same NUMA node.

- a. Install **numactl**:

```
sudo yum install -y numactl
```

- b. When you run your network processing program, bind it to a single NUMA node. For example, the following command binds the shell script, **run.sh**, to NUMA node 0:

```
numactl --cpunodebind=0 --membind=0 run.sh
```

- c. If you have hyperthreading enabled, you can configure your application to only use a single hardware thread per CPU core.

- You can view which CPU cores map to a NUMA node with the **lscpu** command:

```
lscpu | grep NUMA
```

Output:

```
NUMA node(s):      2
NUMA node0 CPU(s): 0-15,32-47
NUMA node1 CPU(s): 16-31,48-63
```

- You can view which hardware threads belong to a physical CPU with the following command:

```
cat /sys/devices/system/cpu/cpu0/topology/thread_siblings_list
```

Output:

```
0,32
```

In this example, threads 0 and 32 map to CPU 0.

- To avoid running on threads 32-47 (which are actually hardware threads of the same CPUs as 0-15, use the following command:

```
numactl --physcpubind=+0-15 --membind=0 ./run.sh
```

9. Use multiple elastic network interfaces for different classes of traffic. For example, if you are running a web server that uses a backend database, use one elastic network interfaces for the web server front end, and another for the database connection.

Enabling enhanced networking with the Intel 82599 VF interface on Linux instances

Amazon EC2 provides enhanced networking capabilities through the Intel 82599 VF interface, which uses the Intel ixgbevf driver.

Contents

- [Requirements \(p. 826\)](#)
- [Testing whether enhanced networking is enabled \(p. 827\)](#)
- [Enabling enhanced networking on Amazon Linux \(p. 828\)](#)
- [Enabling enhanced networking on Ubuntu \(p. 829\)](#)
- [Enabling enhanced networking on other Linux distributions \(p. 830\)](#)
- [Troubleshooting connectivity issues \(p. 832\)](#)

Requirements

To prepare for enhanced networking using the Intel 82599 VF interface, set up your instance as follows:

- Select from the following supported instance types: C3, C4, D2, I2, M4 (excluding m4.16xlarge), and R3.
- Launch the instance from an HVM AMI using Linux kernel version of 2.6.32 or later. The latest Amazon Linux HVM AMIs have the modules required for enhanced networking installed and have the required attributes set. Therefore, if you launch an Amazon EBS-backed, enhanced networking-supported instance using a current Amazon Linux HVM AMI, enhanced networking is already enabled for your instance.

Warning

Enhanced networking is supported only for HVM instances. Enabling enhanced networking with a PV instance can make it unreachable. Setting this attribute without the proper module or module version can also make your instance unreachable.

- Ensure that the instance has internet connectivity.
- Install and configure the [AWS CLI](#) or the [AWS Tools for Windows PowerShell](#) on any computer you choose, preferably your local desktop or laptop. For more information, see [Accessing Amazon EC2 \(p. 3\)](#). Enhanced networking cannot be managed from the Amazon EC2 console.
- If you have important data on the instance that you want to preserve, you should back that data up now by creating an AMI from your instance. Updating kernels and kernel modules, as well as enabling the `sriovNetSupport` attribute, might render incompatible instances or operating systems unreachable. If you have a recent backup, your data will still be retained if this happens.

Testing whether enhanced networking is enabled

Enhanced networking with the Intel 82599 VF interface is enabled if the `ixgbevf` module is installed on your instance and the `sriovNetSupport` attribute is set.

Instance attribute (`sriovNetSupport`)

To check whether an instance has the enhanced networking `sriovNetSupport` attribute set, use one of the following commands:

- [describe-instance-attribute](#) (AWS CLI)

```
aws ec2 describe-instance-attribute --instance-id instance_id --attribute sriovNetSupport
```

- [Get-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

```
Get-EC2InstanceAttribute -InstanceId instance-id -Attribute sriovNetSupport
```

If the attribute isn't set, `SriovNetSupport` is empty. If the attribute is set, the value is simple, as shown in the following example output.

```
"SriovNetSupport": {  
    "Value": "simple"  
},
```

Image attribute (`sriovNetSupport`)

To check whether an AMI already has the enhanced networking `sriovNetSupport` attribute set, use one of the following commands:

- [describe-images](#) (AWS CLI)

```
aws ec2 describe-images --image-id ami_id --query "Images[].SriovNetSupport"
```

- [Get-EC2Image](#) (AWS Tools for Windows PowerShell)

```
(Get-EC2Image -ImageId ami-id).SriovNetSupport
```

If the attribute isn't set, `SriovNetSupport` is empty. If the attribute is set, the value is simple.

Network interface driver

Use the following command to verify that the module is being used on a particular interface, substituting the interface name that you want to check. If you are using a single interface (default), this is `eth0`. If the operating system supports [predictable network names \(p. 830\)](#), this could be a name like `ens5`.

In the following example, the `ixgbevf` module is not loaded, because the listed driver is `vif`.

```
[ec2-user ~]$ ethtool -i eth0
driver: vif
version:
firmware-version:
bus-info: vif-0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

In this example, the `ixgbevf` module is loaded. This instance has enhanced networking properly configured.

```
[ec2-user ~]$ ethtool -i eth0
driver: ixgbevf
version: 4.0.3
firmware-version: N/A
bus-info: 0000:00:03.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: no
supports-register-dump: yes
supports-priv-flags: no
```

Enabling enhanced networking on Amazon Linux

The latest Amazon Linux HVM AMIs have the `ixgbevf` module required for enhanced networking installed and have the required `sriovNetSupport` attribute set. Therefore, if you launch an instance type using a current Amazon Linux HVM AMI, enhanced networking is already enabled for your instance. For more information, see [Testing whether enhanced networking is enabled \(p. 827\)](#).

If you launched your instance using an older Amazon Linux AMI and it does not have enhanced networking enabled already, use the following procedure to enable enhanced networking.

Warning

There is no way to disable the enhanced networking attribute after you've enabled it.

To enable enhanced networking

1. Connect to your instance.
2. From the instance, run the following command to update your instance with the newest kernel and kernel modules, including `ixgbevf`:

```
[ec2-user ~]$ sudo yum update
```

3. From your local computer, reboot your instance using the Amazon EC2 console or one of the following commands: [reboot-instances](#) (AWS CLI), [Restart-EC2Instance](#) (AWS Tools for Windows PowerShell).

4. Connect to your instance again and verify that the `ixgbevf` module is installed and at the minimum recommended version using the `modinfo ixgbevf` command from [Testing whether enhanced networking is enabled \(p. 827\)](#).
5. [EBS-backed instance] From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: `stop-instances` (AWS CLI), `Stop-EC2Instance` (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

[Instance store-backed instance] You can't stop the instance to modify the attribute. Instead, proceed to this procedure: [To enable enhanced networking \(instance store-backed instances\) \(p. 829\)](#).
6. From your local computer, enable the enhanced networking attribute using one of the following commands:
 - [modify-instance-attribute](#) (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id instance_id --srivnet-support simple
```
 - [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -SriovNetSupport "simple"
```
7. (Optional) Create an AMI from the instance, as described in [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#). The AMI inherits the enhanced networking attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.
8. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: `start-instances` (AWS CLI), `Start-EC2Instance` (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
9. Connect to your instance and verify that the `ixgbevf` module is installed and loaded on your network interface using the `ethtool -i ethn` command from [Testing whether enhanced networking is enabled \(p. 827\)](#).

To enable enhanced networking (instance store-backed instances)

Follow the previous procedure until the step where you stop the instance. Create a new AMI as described in [Creating an instance store-backed Linux AMI \(p. 127\)](#), making sure to enable the enhanced networking attribute when you register the AMI.

- [register-image](#) (AWS CLI)

```
aws ec2 register-image --srivnet-support simple ...
```
- [Register-EC2Image](#) (AWS Tools for Windows PowerShell)

```
Register-EC2Image -SriovNetSupport "simple" ...
```

Enabling enhanced networking on Ubuntu

Before you begin, [check if enhanced networking is already enabled \(p. 827\)](#) on your instance.

The Quick Start Ubuntu HVM AMIs include the necessary drivers for enhanced networking. If you have a version of `ixgbevf` earlier than 2.16.4, you can install the `linux-aws` kernel package to get the latest enhanced networking drivers.

The following procedure provides the general steps for compiling the `ixgbevf` module on an Ubuntu instance.

To install the `linux-aws` kernel package

1. Connect to your instance.
2. Update the package cache and packages.

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y linux-aws
```

Important

If during the update process, you are prompted to install `grub`, use `/dev/xvda` to install `grub`, and then choose to keep the current version of `/boot/grub/menu.lst`.

Enabling enhanced networking on other Linux distributions

Before you begin, [check if enhanced networking is already enabled \(p. 827\)](#) on your instance. The latest Quick Start HVM AMIs include the necessary drivers for enhanced networking, therefore you do not need to perform additional steps.

The following procedure provides the general steps if you need to enable enhanced networking with the Intel 82599 VF interface on a Linux distribution other than Amazon Linux or Ubuntu. For more information, such as detailed syntax for commands, file locations, or package and tool support, see the specific documentation for your Linux distribution.

To enable enhanced networking on Linux

1. Connect to your instance.
2. Download the source for the `ixgbevf` module on your instance from Sourceforge at <https://sourceforge.net/projects/e1000/files/ixgbevf%20stable/>.

Versions of `ixgbevf` earlier than 2.16.4, including version 2.14.2, do not build properly on some Linux distributions, including certain versions of Ubuntu.

3. Compile and install the `ixgbevf` module on your instance.

Warning

If you compile the `ixgbevf` module for your current kernel and then upgrade your kernel without rebuilding the driver for the new kernel, your system might revert to the distribution-specific `ixgbevf` module at the next reboot. This could make your system unreachable if the distribution-specific version is incompatible with enhanced networking.

4. Run the `sudo depmod` command to update module dependencies.
5. Update `initramfs` on your instance to ensure that the new module loads at boot time.
6. Determine if your system uses predictable network interface names by default. Systems that use `systemd` or `udev` versions 197 or greater can rename Ethernet devices and they do not guarantee that a single network interface will be named `eth0`. This behavior can cause problems connecting to your instance. For more information and to see other configuration options, see [Predictable Network Interface Names](#) on the freedesktop.org website.
 - a. You can check the `systemd` or `udev` versions on RPM-based systems with the following command:

```
[ec2-user ~]$ rpm -qa | grep -e '^systemd-[0-9]+\+|^\udev-[0-9]+\+'  
systemd-208-11.el7_0.2.x86_64
```

In the above Red Hat Enterprise Linux 7 example, the **systemd** version is 208, so predictable network interface names must be disabled.

- b. Disable predictable network interface names by adding the `net.ifnames=0` option to the `GRUB_CMDLINE_LINUX` line in `/etc/default/grub`.

```
[ec2-user ~]$ sudo sed -i '/^GRUB_CMDLINE_LINUX/s/"$/ net.ifnames=0"/' /etc/default/grub
```

- c. Rebuild the grub configuration file.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. [EBS-backed instance] From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: [stop-instances](#) (AWS CLI), [Stop-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

[Instance store-backed instance] You can't stop the instance to modify the attribute. Instead, proceed to this procedure: [To enable enhanced networking \(instance store-backed instances\) \(p. 831\)](#).

8. From your local computer, enable the enhanced networking attribute using one of the following commands:

- [modify-instance-attribute](#) (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id instance_id --srivnet-support simple
```

- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -SriovNetSupport "simple"
```

9. (Optional) Create an AMI from the instance, as described in [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#). The AMI inherits the enhanced networking attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.

Important

If your instance operating system contains an `/etc/udev/rules.d/70-persistent-net.rules` file, you must delete it before creating the AMI. This file contains the MAC address for the Ethernet adapter of the original instance. If another instance boots with this file, the operating system will be unable to find the device and `eth0` might fail, causing boot issues. This file is regenerated at the next boot cycle, and any instances launched from the AMI create their own version of the file.

10. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: [start-instances](#) (AWS CLI), [Start-EC2Instance](#) (AWS Tools for Windows PowerShell). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
11. (Optional) Connect to your instance and verify that the module is installed.

To enable enhanced networking (instance store-backed instances)

Follow the previous procedure until the step where you stop the instance. Create a new AMI as described in [Creating an instance store-backed Linux AMI \(p. 127\)](#), making sure to enable the enhanced networking attribute when you register the AMI.

- [register-image](#) (AWS CLI)

```
aws ec2 register-image --srivnet-support simple ...
```

- [Register-EC2Image \(AWS Tools for Windows PowerShell\)](#)

```
Register-EC2Image -SriovNetSupport "simple" ...
```

Troubleshooting connectivity issues

If you lose connectivity while enabling enhanced networking, the `ixgbevf` module might be incompatible with the kernel. Try installing the version of the `ixgbevf` module included with the distribution of Linux for your instance.

If you enable enhanced networking for a PV instance or AMI, this can make your instance unreachable.

For more information, see [How do I enable and configure enhanced networking on my EC2 instances?](#).

Troubleshooting the Elastic Network Adapter (ENA)

The Elastic Network Adapter (ENA) is designed to improve operating system health and reduce the chances of long-term disruption because of unexpected hardware behavior and or failures. The ENA architecture keeps device or driver failures as transparent to the system as possible. This topic provides troubleshooting information for ENA.

If you are unable to connect to your instance, start with the [Troubleshooting connectivity issues \(p. 832\)](#) section.

If you are able to connect to your instance, you can gather diagnostic information by using the failure detection and recovery mechanisms that are covered in the later sections of this topic.

Contents

- [Troubleshooting connectivity issues \(p. 832\)](#)
- [Keep-alive mechanism \(p. 833\)](#)
- [Register read timeout \(p. 834\)](#)
- [Statistics \(p. 835\)](#)
- [Driver error logs in syslog \(p. 837\)](#)

Troubleshooting connectivity issues

If you lose connectivity while enabling enhanced networking, the `ena` module might be incompatible with your instance's current running kernel. This can happen if you install the module for a specific kernel version (without `dkms`, or with an improperly configured `dkms.conf` file) and then your instance kernel is updated. If the instance kernel that is loaded at boot time does not have the `ena` module properly installed, your instance will not recognize the network adapter and your instance becomes unreachable.

If you enable enhanced networking for a PV instance or AMI, this can also make your instance unreachable.

If your instance becomes unreachable after enabling enhanced networking with ENA, you can disable the `enaSupport` attribute for your instance and it will fall back to the stock network adapter.

To disable enhanced networking with ENA (EBS-backed instances)

1. From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: [stop-instances \(AWS CLI\)](#), [Stop-EC2Instance \(AWS Tools for Windows PowerShell\)](#). If your

instance is managed by AWS OpsWorks, you should stop the instance in the AWS OpsWorks console so that the instance state remains in sync.

Important

If you are using an instance store-backed instance, you can't stop the instance.

Instead, proceed to [To disable enhanced networking with ENA \(instance store-backed instances\) \(p. 833\)](#).

2. From your local computer, disable the enhanced networking attribute using the following command.
 - [modify-instance-attribute \(AWS CLI\)](#)

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --no-ena-support
```

3. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: [start-instances \(AWS CLI\)](#), [Start-EC2Instance \(AWS Tools for Windows PowerShell\)](#). If your instance is managed by AWS OpsWorks, you should start the instance in the AWS OpsWorks console so that the instance state remains in sync.
4. (Optional) Connect to your instance and try reinstalling the ena module with your current kernel version by following the steps in [Enabling enhanced networking with the Elastic Network Adapter \(ENA\) on Linux instances \(p. 814\)](#).

To disable enhanced networking with ENA (instance store-backed instances)

If your instance is an instance store-backed instance, create a new AMI as described in [Creating an instance store-backed Linux AMI \(p. 127\)](#). Be sure to disable the enhanced networking `enaSupport` attribute when you register the AMI.

- [register-image \(AWS CLI\)](#)

```
$ aws ec2 register-image --no-ena-support ...
```

- [Register-EC2Image \(AWS Tools for Windows PowerShell\)](#)

```
C:\> Register-EC2Image -EnaSupport $false ...
```

Keep-alive mechanism

The ENA device posts keep-alive events at a fixed rate (usually once every second). The ENA driver implements a watchdog mechanism, which checks for the presence of these keep-alive messages. If a message or messages are present, the watchdog is rearmed, otherwise the driver concludes that the device experienced a failure and then does the following:

- Dumps its current statistics to syslog
- Resets the ENA device
- Resets the ENA driver state

The above reset procedure may result in some traffic loss for a short period of time (TCP connections should be able to recover), but should not otherwise affect the user.

The ENA device may also indirectly request a device reset procedure, by not sending a keep-alive notification, for example, if the ENA device reaches an unknown state after loading an irrecoverable configuration.

Below is an example of the reset procedure:

```
[18509.800135] ena 0000:00:07.0 eth1: Keep alive watchdog timeout. // The watchdog process initiates a reset
[18509.815244] ena 0000:00:07.0 eth1: Trigger reset is on
[18509.825589] ena 0000:00:07.0 eth1: tx_timeout: 0 // The driver logs the current statistics
[18509.834253] ena 0000:00:07.0 eth1: io_suspend: 0
[18509.842674] ena 0000:00:07.0 eth1: io_resume: 0
[18509.850275] ena 0000:00:07.0 eth1: wd_expired: 1
[18509.857855] ena 0000:00:07.0 eth1: interface_up: 1
[18509.865415] ena 0000:00:07.0 eth1: interface_down: 0
[18509.873468] ena 0000:00:07.0 eth1: admin_q_pause: 0
[18509.881075] ena 0000:00:07.0 eth1: queue_0_tx_cnt: 0
[18509.888629] ena 0000:00:07.0 eth1: queue_0_tx_bytes: 0
[18509.895286] ena 0000:00:07.0 eth1: queue_0_tx_queue_stop: 0
.....
.....
[18511.280972] ena 0000:00:07.0 eth1: free uncompleted tx skb qid 3 idx 0x7 // At the end of the down process, the driver discards incomplete packets.
[18511.420112] [ENA_COM: ena_com_validate_version] ena device version: 0.10 //The driver begins its up process
[18511.420119] [ENA_COM: ena_com_validate_version] ena controller version: 0.0.1 implementation version 1
[18511.420127] [ENA_COM: ena_com_admin_init] ena_defs : Version:[b9692e8] Build date [Wed Apr 6 09:54:21 IDT 2016]
[18512.252108] ena 0000:00:07.0: Device watchdog is Enabled
[18512.674877] ena 0000:00:07.0: irq 46 for MSI/MSI-X
[18512.674933] ena 0000:00:07.0: irq 47 for MSI/MSI-X
[18512.674990] ena 0000:00:07.0: irq 48 for MSI/MSI-X
[18512.675037] ena 0000:00:07.0: irq 49 for MSI/MSI-X
[18512.675085] ena 0000:00:07.0: irq 50 for MSI/MSI-X
[18512.675141] ena 0000:00:07.0: irq 51 for MSI/MSI-X
[18512.675188] ena 0000:00:07.0: irq 52 for MSI/MSI-X
[18512.675233] ena 0000:00:07.0: irq 53 for MSI/MSI-X
[18512.675279] ena 0000:00:07.0: irq 54 for MSI/MSI-X
[18512.772641] [ENA_COM: ena_com_set_hash_function] Feature 10 isn't supported
[18512.772647] [ENA_COM: ena_com_set_hash_ctrl] Feature 18 isn't supported
[18512.775945] ena 0000:00:07.0: Device reset completed successfully // The reset process is complete
```

Register read timeout

The ENA architecture suggests a limited usage of memory mapped I/O (MMIO) read operations. MMIO registers are accessed by the ENA device driver only during its initialization procedure.

If the driver logs (available in **dmesg** output) indicate failures of read operations, this may be caused by an incompatible or incorrectly compiled driver, a busy hardware device, or hardware failure.

Intermittent log entries that indicate failures on read operations should not be considered an issue; the driver will retry them in this case. However, a sequence of log entries containing read failures indicate a driver or hardware problem.

Below is an example of driver log entry indicating a read operation failure due to a timeout:

```
[ 47.113698] [ENA_COM: ena_com_reg_bar_read32] reading reg failed for timeout. expected: req id[1] offset[88] actual: req id[57006] offset[0]
[ 47.333715] [ENA_COM: ena_com_reg_bar_read32] reading reg failed for timeout. expected: req id[2] offset[8] actual: req id[57007] offset[0]
[ 47.346221] [ENA_COM: ena_com_dev_reset] Reg read32 timeout occurred
```

Statistics

If you experience insufficient network performance or latency issues, you should retrieve the device statistics and examine them. These statistics can be obtained using **ethtool**, as shown below:

```
[ec2-user ~]$ ethtool -S ethN
NIC statistics:
    tx_timeout: 0
    io_suspend: 0
    io_resume: 0
    wd_expired: 0
    interface_up: 1
    interface_down: 0
    admin_q_pause: 0
    queue_0_tx_cnt: 4329
    queue_0_tx_bytes: 1075749
    queue_0_tx_queue_stop: 0
...
```

The following command output parameters are described below:

tx_timeout: *N*

The number of times that the Netdev watchdog was activated.

io_suspend: *N*

Unsupported. This value should always be zero.

io_resume: *N*

Unsupported. This value should always be zero.

wd_expired: *N*

The number of times that the driver did not receive the keep-alive event in the preceding 3 seconds.

interface_up: *N*

The number of times that the ENA interface was brought up.

interface_down: *N*

The number of times that the ENA interface was brought down.

admin_q_pause: *N*

The admin queue is in an unstable state. This value should always be zero.

queue_N_tx_cnt: *N*

The number of transmitted packets for queue *N*.

queue_N_tx_bytes: *N*

The number of transmitted bytes for queue *N*.

queue_N_tx_queue_stop: *N*

The number of times that queue *N* was full and stopped.

queue_N_tx_queue_wakeup: *N*

The number of times that queue *N* resumed after being stopped.

queue_N_tx_dma_mapping_err: *N*

Direct memory access error count. If this value is not 0, it indicates low system resources.

queue_*N*_tx_napi_comp: *N*

The number of times the napi handler called napi_complete for queue *N*.

queue_*N*_tx_poll: *N*

The number of times the napi handler was scheduled for queue *N*.

queue_*N*_tx_doorbells: *N*

The number of transmission doorbells for queue *N*.

queue_*N*_tx_linearize: *N*

The number of times SKB linearization was attempted for queue *N*.

queue_*N*_tx_linearize_failed: *N*

The number of times SKB linearization failed for queue *N*.

queue_*N*_tx_prepare_ctx_err: *N*

The number of times ena_com_prepare_tx failed for queue *N*. This value should always be zero; if not, see the driver logs.

queue_*N*_tx_missing_tx_comp: code*N*

The number of packets that were left uncompleted for queue *N*. This value should always be zero.

queue_*N*_tx_bad_req_id: *N*

Invalid req_id for queue *N*. The valid req_id is zero, minus the queue_size, minus 1.

queue_*N*_rx_cnt: *N*

The number of received packets for queue *N*.

queue_*N*_rx_bytes: *N*

The number of received bytes for queue *N*.

queue_*N*_rx_refil_partial: *N*

The number of times the driver did not succeed in refilling the empty portion of the rx queue with the buffers for queue *N*. If this value is not zero, it indicates low memory resources.

queue_*N*_rx_bad_csum: *N*

The number of times the rx queue had a bad checksum for queue *N* (only if rx checksum offload is supported).

queue_*N*_rx_page_alloc_fail: *N*

The number of time that page allocation failed for queue *N*. If this value is not zero, it indicates low memory resources.

queue_*N*_rx_skb_alloc_fail: *N*

The number of time that SKB allocation failed for queue *N*. If this value is not zero, it indicates low system resources.

queue_*N*_rx_dma_mapping_err: *N*

Direct memory access error count. If this value is not 0, it indicates low system resources.

queue_*N*_rx_bad_desc_num: *N*

Too many buffers per packet. If this value is not 0, it indicates usage of very small buffers.

queue_N_rx_small_copy_len_pkt: N

Optimization: For packets smaller than this threshold, which is set by sysfs, the packet is copied directly to the stack to avoid allocation of a new page.

ena_admin_q_aborted_cmd: N

The number of admin commands that were aborted. This usually happens during the auto-recovery procedure.

ena_admin_q_submitted_cmd: N

The number of admin queue doorbells.

ena_admin_q_completed_cmd: N

The number of admin queue completions.

ena_admin_q_out_of_space: N

The number of times that the driver tried to submit new admin command, but the queue was full.

ena_admin_q_no_completion: N

The number of times that the driver did not get an admin completion for a command.

Driver error logs in syslog

The ENA driver writes log messages to **syslog** during system boot. You can examine these logs to look for errors if you are experiencing issues. Below is an example of information logged by the ENA driver in **syslog** during system boot, along with some annotations for select messages.

```
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  478.416939] [ENA_COM: ena_com_validate_version]
ena device version: 0.10
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  478.420915] [ENA_COM: ena_com_validate_version]
ena controller version: 0.0.1 implementation version 1
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.256831] ena 0000:00:03.0: Device watchdog is
Enabled
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.672947] ena 0000:00:03.0: creating 8 io
queues. queue size: 1024
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.680885] [ENA_COM:
ena_com_init_interrupt_moderation] Feature 20 isn't supported // Interrupt moderation is
not supported by the device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.691609] [ENA_COM: ena_com_get_feature_ex]
Feature 10 isn't supported // RSS HASH function configuration is not supported by the
device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.694583] [ENA_COM: ena_com_get_feature_ex]
Feature 18 isn't supported //RSS HASH input source configuration is not supported by the
device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.697433] [ENA_COM:
ena_com_set_host_attributes] Set host attribute isn't supported
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.701064] ena 0000:00:03.0 (unnamed
net_device) (uninitialized): Cannot set host attributes
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.704917] ena 0000:00:03.0: Elastic Network
Adapter (ENA) found at mem f3000000, mac addr 02:8a:3c:1e:13:b5 Queues 8
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  480.805037] EXT4-fs (xvda1): re-mounted. Opts:
(null)
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  481.025842] NET: Registered protocol family 10
```

Which errors can I ignore?

The following warnings that may appear in your system's error logs can be ignored for the Elastic Network Adapter:

Set host attribute isn't supported

Host attributes are not supported for this device.
failed to alloc buffer for rx queue

This is a recoverable error, and it indicates that there may have been a memory pressure issue when the error was thrown.

Feature **X** isn't supported

The referenced feature is not supported by the Elastic Network Adapter. Possible values for **X** include:

- **10**: RSS Hash function configuration is not supported for this device.
- **12**: RSS Indirection table configuration is not supported for this device.
- **18**: RSS Hash Input configuration is not supported for this device.
- **20**: Interrupt moderation is not supported for this device.
- **27**: The Elastic Network Adapter driver does not support polling the Ethernet capabilities from snmpd.

Failed to config AENQ

The Elastic Network Adapter does not support AENQ configuration.

Trying to set unsupported AENQ events

This error indicates an attempt to set an AENQ events group that is not supported by the Elastic Network Adapter.

Elastic Fabric Adapter

An Elastic Fabric Adapter (EFA) is a network device that you can attach to your Amazon EC2 instance to accelerate High Performance Computing (HPC) and machine learning applications. EFA enables you to achieve the application performance of an on-premises HPC cluster, with the scalability, flexibility, and elasticity provided by the AWS Cloud.

EFA provides lower and more consistent latency and higher throughput than the TCP transport traditionally used in cloud-based HPC systems. It enhances the performance of inter-instance communication that is critical for scaling HPC and machine learning applications. It is optimized to work on the existing AWS network infrastructure and it can scale depending on application requirements.

EFA integrates with Libfabric 1.11.1 and it supports Open MPI 4.0.5 and Intel MPI 2019 Update 7 for HPC applications, and Nvidia Collective Communications Library (NCCL) for machine learning applications.

Note

The OS-bypass capabilities of EFAs are not supported on Windows instances. If you attach an EFA to a Windows instance, the instance functions as an Elastic Network Adapter, without the added EFA capabilities.

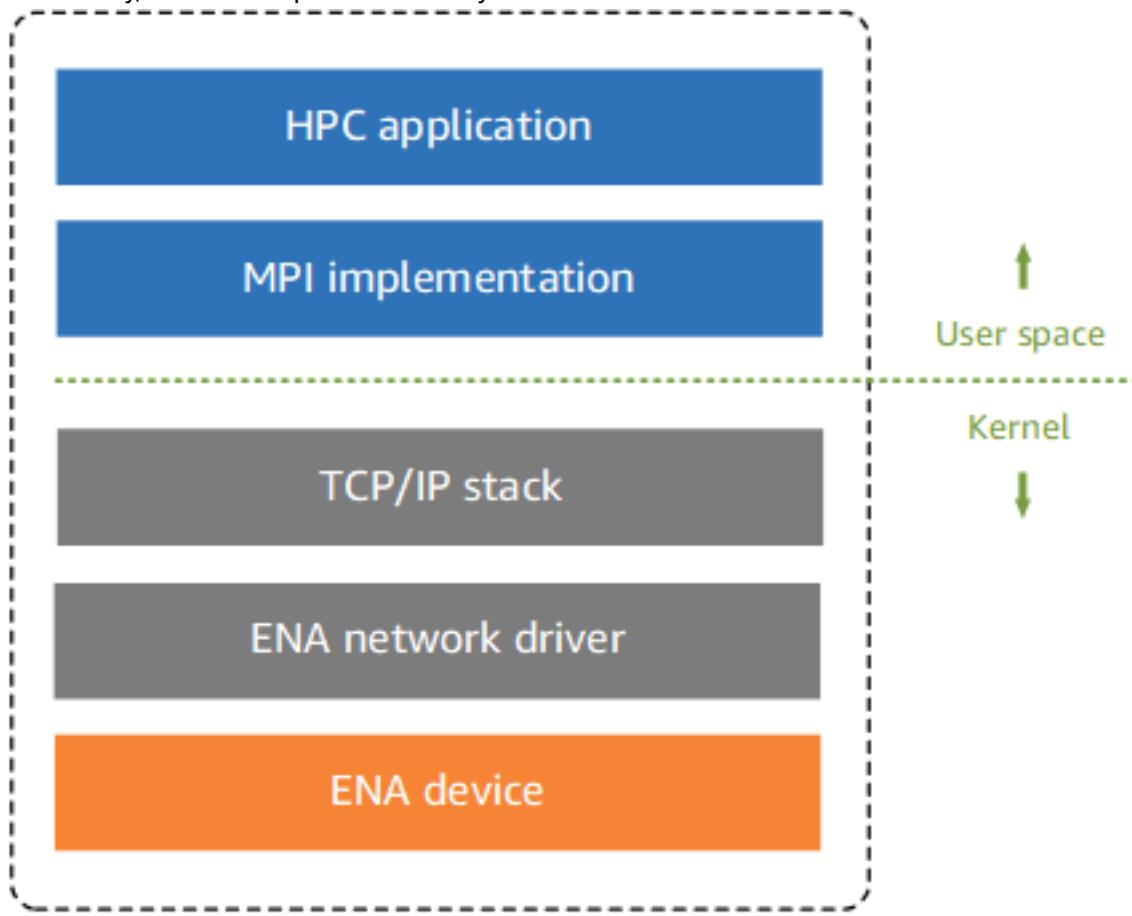
Contents

- [EFA basics \(p. 839\)](#)
- [Supported interfaces and libraries \(p. 840\)](#)
- [Supported instance types \(p. 840\)](#)
- [Supported AMIs \(p. 840\)](#)
- [EFA limitations \(p. 841\)](#)
- [Getting started with EFA and MPI \(p. 841\)](#)

- [Getting started with EFA and NCCL \(p. 849\)](#)
- [Working with EFA \(p. 865\)](#)
- [Monitoring an EFA \(p. 867\)](#)
- [Verifying the EFA installer using a checksum \(p. 868\)](#)

EFA basics

An EFA is an Elastic Network Adapter (ENA) with added capabilities. It provides all of the functionality of an ENA, with an additional OS-bypass functionality. OS-bypass is an access model that allows HPC and machine learning applications to communicate directly with the network interface hardware to provide low-latency, reliable transport functionality.



Traditional HPC software stack in EC2

Traditionally, HPC applications use the Message Passing Interface (MPI) to interface with the system's network transport. In the AWS Cloud, this has meant that applications interface with MPI, which then uses the operating system's TCP/IP stack and the ENA device driver to enable network communication between instances.

With an EFA, HPC applications use MPI or NCCL to interface with the *Libfabric* API. The *Libfabric* API bypasses the operating system kernel and communicates directly with the EFA device to put packets on the network. This reduces overhead and enables the HPC application to run more efficiently.

Note

Libfabric is a core component of the OpenFabrics Interfaces (OFI) framework, which defines and exports the user-space API of OFI. For more information, see the [Libfabric OpenFabrics](#) website.

Differences between EFAs and ENAs

Elastic Network Adapters (ENAs) provide traditional IP networking features that are required to support VPC networking. EFAs provide all of the same traditional IP networking features as ENAs, and they also support OS-bypass capabilities. OS-bypass enables HPC and machine learning applications to bypass the operating system kernel and to communicate directly with the EFA device.

Supported interfaces and libraries

EFA supports the following interfaces and libraries:

- Open MPI 4.0.5
- Intel MPI 2019 Update 7
- NVIDIA Collective Communications Library (NCCL) 2.4.2 and later

Supported instance types

The following instance types support EFAs:

- General purpose: m5dn.24xlarge | m5n.24xlarge
- Compute optimized: c5n.18xlarge | c5n.metal
- Memory optimized: r5dn.24xlarge | r5n.24xlarge
- Storage optimized: i3en.24xlarge | i3en.metal
- Accelerated computing: g4dn.metal | inf1.24xlarge | p3dn.24xlarge

The available instance types vary by Region. To see the available instance types that support EFA in a Region, use the `describe-instance-types` command with the `--region` option and the appropriate Region code.

```
aws ec2 describe-instance-types --region us-east-2 --filters Name=network-info.efa-supported,Values=true --query InstanceTypes[*].[InstanceType] --output text
```

The following is example output.

```
g4dn.metal
i3en.24xlarge
r5n.24xlarge
c5n.18xlarge
m5n.24xlarge
inf1.24xlarge
m5dn.24xlarge
c5n.metal
p3dn.24xlarge
i3en.metal
r5dn.24xlarge
```

Supported AMIs

The following AMIs support EFAs:

- Amazon Linux and Amazon Linux 2
- CentOS 7
- RHEL 7.6, 7.7, and 7.8
- Ubuntu 16.04 and 18.04

EFA limitations

EFA has the following limitations:

- You can attach only one EFA per instance.
- EFA OS-bypass traffic is limited to a single subnet. In other words, EFA traffic cannot be sent from one subnet to another. Normal IP traffic from the EFA can be sent from one subnet to another.
- EFA OS-bypass traffic is not routable. Normal IP traffic from the EFA remains routable.
- The EFA must be a member of a security group that allows all inbound and outbound traffic to and from the security group itself.

Getting started with EFA and MPI

This tutorial helps you to launch an EFA and MPI-enabled instance cluster for HPC workloads. In this tutorial, you will perform the following steps:

Contents

- Step 1: Prepare an EFA-enabled security group (p. 841)
- Step 2: Launch a temporary instance (p. 842)
- Step 3: Install the EFA software (p. 842)
- Step 4: Disable ptrace protection (p. 845)
- Step 5: (Optional) Install Intel MPI (p. 845)
- Step 6: Install your HPC application (p. 846)
- Step 7: Create an EFA-enabled AMI (p. 847)
- Step 8: Launch EFA-enabled instances into a cluster placement group (p. 847)
- Step 9: Terminate the temporary instance (p. 848)
- Step 10: Enable passwordless SSH (p. 848)

Step 1: Prepare an EFA-enabled security group

An EFA requires a security group that allows all inbound and outbound traffic to and from the security group itself.

To create an EFA-enabled security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups** and then choose **Create Security Group**.
3. In the **Create Security Group** window, do the following:
 - a. For **Security group name**, enter a descriptive name for the security group, such as **EFA-enabled security group**.
 - b. (Optional) For **Description**, enter a brief description of the security group.

- c. For **VPC**, select the VPC into which you intend to launch your EFA-enabled instances.
- d. Choose **Create**.
4. Select the security group that you created, and on the **Description** tab, copy the **Group ID**.
5. On the **Inbound** tab, do the following:
 - a. Choose **Edit**.
 - b. For **Type**, choose **All traffic**.
 - c. For **Source**, choose **Custom** and paste the security group ID that you copied into the field.
 - d. Choose **Save**.
6. On the **Outbound** tab, do the following:
 - a. Choose **Edit**.
 - b. For **Type**, choose **All traffic**.
 - c. For **Destination**, choose **Custom** and paste the security group ID that you copied into the field.
 - d. Choose **Save**.

Step 2: Launch a temporary instance

Launch a temporary instance that you can use to install and configure the EFA software components. You use this instance to create an EFA-enabled AMI from which you can launch your EFA-enabled instances.

To launch a temporary instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. On the **Choose an AMI** page, choose **Select** for one of the supported AMIs (p. 840).
4. On the **Choose an Instance Type** page, select one of the supported instance types (p. 840) and then choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, do the following:
 - a. For **Subnet**, choose the subnet in which to launch the instance.
 - b. For **Elastic Fabric Adapter**, choose **Enable**.
 - c. In the **Network Interfaces** section, for device **eth0**, choose **New network interface**.
 - d. Choose **Next: Add Storage**.
6. On the **Add Storage** page, specify the volumes to attach to the instances in addition to the volumes that are specified by the AMI (such as the root device volume). Then choose **Next: Add Tags**.
7. On the **Add Tags** page, specify a tag that you can use to identify the temporary instance, and then choose **Next: Configure Security Group**.
8. On the **Configure Security Group** page, for **Assign a security group**, select **Select an existing security group**, and then select the security group that you created in **Step 1**.
9. On the **Review Instance Launch** page, review the settings, and then choose **Launch** to choose a key pair and to launch your instance.

Step 3: Install the EFA software

Install the EFA-enabled kernel, EFA drivers, Libfabric, and Open MPI stack that is required to support EFA on your temporary instance.

The steps differ depending on whether you intend to use EFA with Open MPI, with Intel MPI, or with Open MPI and Intel MPI.

To install the EFA software

1. Connect to the instance you launched. For more information, see [Connect to your Linux instance \(p. 563\)](#).
2. To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process may take a few minutes.

- Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7/7.8, CentOS 7

```
$ sudo yum update -y
```

- Ubuntu 16.04 and Ubuntu 18.04

```
$ sudo apt-get update -y
```

```
$ sudo apt-get upgrade -y
```

3. Download the EFA software installation files. The software installation files are packaged into a compressed tarball (.tar.gz) file. To download the latest *stable* version, use the following command.

```
$ curl -O https://efa-installer.amazonaws.com/aws-efa-installer-1.10.0.tar.gz
```

You can also get the latest version by replacing the version number with `latest` in the preceding command.

4. (Optional) Verify the authenticity and integrity of the EFA tarball (.tar.gz) file. We recommend that you do this to verify the identity of the software publisher and to check that the file has not been altered or corrupted since it was published. If you do not want to verify the tarball file, skip this step.

Note

Alternatively, if you prefer to verify the tarball file by using an MD5 or SHA256 checksum instead, see [Verifying the EFA installer using a checksum \(p. 868\)](#).

- a. Download the public GPG key and import it into your keyring.

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer.key
```

```
$ gpg --import aws-efa-installer.key
```

The command should return a key value. Make a note of the key value, because you need it in the next step.

- b. Verify the GPG key's fingerprint. Run the following command and specify the key value from the previous step.

```
$ gpg --fingerprint key_value
```

The command should return a fingerprint that is identical to `4E90 91BC BB97 A96B 26B1 5E59 A054 80B1 DD2D 3CCC`. If the fingerprint does not match, don't run the EFA installation script, and contact AWS Support.

- c. Download the signature file and verify the signature of the EFA tarball file.

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer-1.10.0.tar.gz.sig
```

```
$ gpg --verify ./aws-efa-installer-1.10.0.tar.gz.sig
```

The following shows example output.

```
gpg: Signature made Wed 29 Jul 2020 12:50:13 AM UTC using RSA key ID DD2D3CCC
gpg: Good signature from "Amazon EC2 EFA <ec2-efa-maintainers@amazon.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:                 There is no indication that the signature belongs to the owner.
Primary key fingerprint: 4E90 91BC BB97 A96B 26B1  5E59 A054 80B1 DD2D 3CCC
```

If the result includes `Good signature`, and the fingerprint matches the fingerprint returned in the previous step, proceed to the next step. If not, don't run the EFA installation script, and contact AWS Support.

5. Extract the files from the compressed `.tar.gz` file and navigate into the extracted directory.

```
$ tar -xf aws-efa-installer-1.10.0.tar.gz
```

```
$ cd aws-efa-installer
```

6. Install the EFA software. Do one of the following depending on your use case.

- **Open MPI and Intel MPI**

If you intend to use EFA with Open MPI and Intel MPI, you must install the EFA software with Libfabric and Open MPI, and you must complete Step 5: (Optional) Install Intel MPI. To install the EFA software with Libfabric and Open MPI, run the following command.

```
$ sudo ./efa_installer.sh -y
```

Libfabric is installed in the `/opt/amazon/efa` directory, while Open MPI is installed in the `/opt/amazon/openmpi` directory.

- **Open MPI only**

If you intend to use EFA with Open MPI only, you must install the EFA software with Libfabric and Open MPI, and you can skip Step 5: (Optional) Install Intel MPI. To install the EFA software with Libfabric and Open MPI, run the following command.

```
$ sudo ./efa_installer.sh -y
```

Libfabric is installed in the `/opt/amazon/efa` directory, while Open MPI is installed in the `/opt/amazon/openmpi` directory.

- **Intel MPI only**

If you intend to use EFA with Intel MPI only, you can install the EFA software without Libfabric and Open MPI. In this case, Intel MPI uses its embedded Libfabric. If you choose to do this, you must complete Step 5: (Optional) Install Intel MPI.

To install the EFA software without Libfabric and Open MPI, run the following command.

```
$ sudo ./efa_installer.sh -y --minimal
```

7. Log out of the instance and then log back in.
8. Confirm that the EFA software components were successfully installed.

```
$ fi_info -p efa
```

The command should return information about the Libfabric EFA interfaces. The following example shows the command output.

```
provider: efa
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-rdm
    version: 2.0
    type: FI_EP_RDM
    protocol: FI_PROTO_EFA
provider: efa
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-dgram
    version: 2.0
    type: FI_EP_DGRAM
    protocol: FI_PROTO_EFA
provider: efa;ofi_rxd
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-dgram
    version: 1.0
    type: FI_EP_RDM
    protocol: FI_PROTO_RXD
```

Step 4: Disable ptrace protection

To improve your HPC application's performance, Libfabric uses the instance's local memory for interprocess communications when the processes are running on the same instance.

The shared memory feature uses Cross Memory Attach (CMA), which is not supported with *ptrace protection*. If you are using a Linux distribution that has ptrace protection enabled by default, such as Ubuntu, you must disable it. If your Linux distribution does not have ptrace protection enabled by default, skip this step.

To disable ptrace protection

Do one of the following:

- To temporarily disable ptrace protection for testing purposes, run the following command.

```
$ sudo sysctl -w kernel.yama.ptrace_scope=0
```

- To permanently disable ptrace protection, add `kernel.yama.ptrace_scope = 0` to `/etc/sysctl.d/10-ptrace.conf` and reboot the instance.

Step 5: (Optional) Install Intel MPI

Important

If you intend to only use Open MPI, skip this step. Perform this step only if you intend to use Intel MPI.

Intel MPI requires an additional installation and environment variable configuration.

Prerequisites

Ensure that the user performing the following steps has sudo permissions.

To install Intel MPI

1. To download the Intel MPI installation files, see the [Intel Developer Zone website](#).

You must register before you can download the installation files. After you have registered, do the following:

- a. For Product, choose **Intel MPI Library for Linux**.
- b. For Version, choose **2019 Update 7**, and then choose **Full Product**.
2. The installation files are packaged into a compressed `.tar.gz` file. Extract the files from the compressed `.tar.gz` file and navigate into the extracted directory.

```
$ tar -xf file_name.tgz
```

```
$ cd directory_name
```

3. Open `silent.cfg` using your preferred text editor. In line 10, change `ACCEPT_EULA=decline` to `ACCEPT_EULA=accept`. Save the changes and close the file.
4. Run the installation script.

```
$ sudo ./install.sh -s silent.cfg
```

Intel MPI is installed in the `/opt/intel/impi/` directory by default.

5. Add the Intel MPI environment variables to the corresponding shell startup scripts to ensure that they are set each time that the instance starts. Do one of the following depending on your shell.
 - For **bash**, add the following environment variable to `/home/username/.bashrc` and `/home/username/.bash_profile`.

```
source /opt/intel/compilers_and_libraries/linux/mpi/intel64/bin/mpivars.sh
```

- For **csh** and **tcsh**, add the following environment variable to `/home/username/.cshrc`.

```
source /opt/intel/compilers_and_libraries/linux/mpi/intel64/bin/mpivars.csh
```

6. Log out of the instance and then log back in.
7. Run the following command to confirm that Intel MPI was successfully installed.

```
$ which mpicc
```

Ensure that the returned path includes the `/opt/intel/` subdirectory.

Note

If you no longer want to use Intel MPI, remove the environment variables from the shell startup scripts.

Step 6: Install your HPC application

Install the HPC application on the temporary instance. The installation procedure varies depending on the specific HPC application. For more information about installing software on your Linux instance, see [Managing Software on Your Linux Instance](#).

Note

You might need to refer to your HPC application's documentation for installation instructions.

Step 7: Create an EFA-enabled AMI

After you have installed the required software components, you create an AMI that you can reuse to launch your EFA-enabled instances.

To create an AMI from your temporary instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the temporary instance that you created and choose **Actions, Image, Create image**.
4. For **Create image**, do the following:
 - a. For **Image name**, enter a descriptive name for the AMI.
 - b. (Optional) For **Image description**, enter a brief description of the purpose of the AMI.
 - c. Choose **Create image**.
5. In the navigation pane, choose **AMIs**.
6. Locate the AMI that you created in the list. Wait for the status to change from pending to available before continuing to the next step.

Step 8: Launch EFA-enabled instances into a cluster placement group

Launch your EFA-enabled instances into a cluster placement group using the EFA-enabled AMI that you created in **Step 7**, and the EFA-enabled security group that you created in **Step 1**.

Note

It is not an absolute requirement to launch your EFA-enabled instances into a cluster placement group. However, we do recommend running your EFA-enabled instances in a cluster placement group as it launches the instances into a low-latency group in a single Availability Zone.

To launch your EFA-enabled instances into a cluster placement group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. On the **Choose an AMI** page, choose **My AMIs**, find the AMI that you created in **Step 7**, and then choose **Select**.
4. On the **Choose an Instance Type** page, select one of the [supported instance types \(p. 840\)](#) and then choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, do the following:
 - a. For **Number of instances**, enter the number of EFA-enabled instances that you want to launch.
 - b. For **Network and Subnet**, select the VPC and subnet into which to launch the instances.
 - c. For **Placement group**, select **Add instance to placement group**.
 - d. For **Placement group name**, select **Add to a new placement group**, enter a descriptive name for the placement group, and then for **Placement group strategy**, select **cluster**.
 - e. For **EFA**, choose **Enable**.
 - f. In the **Network Interfaces** section, for device **eth0**, choose **New network interface**. You can optionally specify a primary IPv4 address and one or more secondary IPv4 addresses. If you're launching the instance into a subnet that has an associated IPv6 CIDR block, you can optionally specify a primary IPv6 address and one or more secondary IPv6 addresses.
 - g. Choose **Next: Add Storage**.

6. On the **Add Storage** page, specify the volumes to attach to the instances in addition to the volumes specified by the AMI (such as the root device volume), and then choose **Next: Add Tags**.
7. On the **Add Tags** page, specify tags for the instances, such as a user-friendly name, and then choose **Next: Configure Security Group**.
8. On the **Configure Security Group** page, for **Assign a security group**, select **Select an existing security group**, and then select the security group that you created in **Step 1**.
9. Choose **Review and Launch**.
10. On the **Review Instance Launch** page, review the settings, and then choose **Launch** to choose a key pair and to launch your instances.

Step 9: Terminate the temporary instance

At this point, you no longer need the temporary instance that you launched. You can terminate the instance to stop incurring charges for it.

To terminate the temporary instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the temporary instance that you created and then choose **Actions, Instance state, Terminate instance**.
4. When prompted for confirmation, choose **Terminate**.

Step 10: Enable passwordless SSH

To enable your applications to run across all of the instances in your cluster, you must enable passwordless SSH access from the leader node to the member nodes. The leader node is the instance from which you run your applications. The remaining instances in the cluster are the member nodes.

To enable passwordless SSH between the instances in the cluster

1. Select one instance in the cluster as the leader node, and connect to it.
2. Disable `strictHostKeyChecking` and enable `ForwardAgent` on the leader node. Open `~/.ssh/config` using your preferred text editor and add the following.

```
Host *
  ForwardAgent yes
Host *
  StrictHostKeyChecking no
```

3. Generate an RSA key pair.

```
$ ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
```

The key pair is created in the `$HOME/.ssh/` directory.

4. Change the permissions of the private key on the leader node.

```
$ chmod 600 ~/.ssh/id_rsa
chmod 600 ~/.ssh/config
```

5. Open `~/.ssh/id_rsa.pub` using your preferred text editor and copy the key.

6. For each member node in the cluster, do the following:
 - a. Connect to the instance.
 - b. Open `~/.ssh/authorized_keys` using your preferred text editor and add the public key that you copied earlier.
7. To test that the passwordless SSH is functioning as expected, connect to your leader node and run the following command.

```
$ ssh member_node_private_ip
```

You should connect to the member node without being prompted for a key or password.

Getting started with EFA and NCCL

The Nvidia Collective Communications Library (NCCL) is a library of standard collective communication routines for multiple GPUs across a single node or multiple nodes. NCCL can be used together with EFA, Libfabric, and MPI to support various machine learning workloads. For more information, see the [NCCL](#) website.

Note

- NCCL with EFA is supported with p3dn.24xlarge instances only.
- Only NCCL 2.4.2 and later is supported with EFA.

The following tutorials help you to launch an EFA and NCCL-enabled instance cluster for machine learning workloads.

- [Using a base AMI \(p. 849\)](#)
- [Using an AWS Deep Learning AMI \(p. 860\)](#)

Using a base AMI

The following steps help you to get started using one of the [supported base AMIs \(p. 840\)](#).

Contents

- [Step 1: Prepare an EFA-enabled security group \(p. 850\)](#)
- [Step 2: Launch a temporary instance \(p. 850\)](#)
- [Step 3: Install the EFA software \(p. 851\)](#)
- [Step 4: Install Nvidia GPU drivers and the Nvidia CUDA toolkit \(p. 853\)](#)
- [Step 5: Install cuDNN \(p. 854\)](#)
- [Step 6: Install NCCL \(p. 855\)](#)
- [Step 7: Install the aws-ofi-nccl plugin \(p. 855\)](#)
- [Step 8: Install the NCCL tests \(p. 856\)](#)
- [Step 9: Test your EFA and NCCL configuration \(p. 857\)](#)
- [Step 10: Install your machine learning applications \(p. 858\)](#)
- [Step 11: Create an EFA and NCCL-enabled AMI \(p. 858\)](#)
- [Step 12: Terminate the temporary instance \(p. 859\)](#)
- [Step 13: Launch EFA and NCCL-enabled instances into a cluster placement group \(p. 859\)](#)

- Step 14: Enable passwordless SSH (p. 860)

Step 1: Prepare an EFA-enabled security group

An EFA requires a security group that allows all inbound and outbound traffic to and from the security group itself.

To create an EFA-enabled security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups** and then choose **Create Security Group**.
3. In the **Create Security Group** window, do the following:
 - a. For **Security group name**, enter a descriptive name for the security group, such as **EFA-enabled security group**.
 - b. (Optional) For **Description**, enter a brief description of the security group.
 - c. For **VPC**, select the VPC into which you intend to launch your EFA-enabled instances.
 - d. Choose **Create**.
4. Select the security group that you created, and on the **Description** tab, copy the **Group ID**.
5. On the **Inbound** tab, do the following:
 - a. Choose **Edit**.
 - b. For **Type**, choose **All traffic**.
 - c. For **Source**, choose **Custom** and paste the security group ID that you copied into the field.
 - d. Choose **Save**.
6. On the **Outbound** tab, do the following:
 - a. Choose **Edit**.
 - b. For **Type**, choose **All traffic**.
 - c. For **Destination**, choose **Custom** and paste the security group ID that you copied into the field.
 - d. Choose **Save**.

Step 2: Launch a temporary instance

Launch a temporary instance that you can use to install and configure the EFA software components. You use this instance to create an EFA-enabled AMI from which you can launch your EFA-enabled instances.

To launch a temporary instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. On the **Choose an AMI** page, choose one of the supported AMIs.
4. On the **Choose an Instance Type** page, select **p3dn.24xlarge** and then choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, do the following:
 - a. For **Subnet**, choose the subnet in which to launch the instance.
 - b. For **Elastic Fabric Adapter**, choose **Enable**.
 - c. In the **Network Interfaces** section, for device **eth0**, choose **New network interface**.
 - d. Choose **Next: Add Storage**.

6. On the **Add Storage** page, specify the volumes to attach to the instances, in addition to the volumes specified by the AMI (such as the root device volume). Then choose **Next: Add Tags**.
7. On the **Add Tags** page, specify a tag that you can use to identify the temporary instance, and then choose **Next: Configure Security Group**.
8. On the **Configure Security Group** page, for **Assign a security group**, select **Select an existing security group**. Then select the security group that you created in **Step 1**.
9. On the **Review Instance Launch** page, review the settings, and then choose **Launch** to choose a key pair and to launch your instance.

Step 3: Install the EFA software

Install the EFA-enabled kernel, EFA drivers, Libfabric, and Open MPI stack that is required to support EFA on your temporary instance.

To install the EFA software

1. Connect to the instance you launched. For more information, see [Connect to your Linux instance \(p. 563\)](#).
2. To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process may take a few minutes.
 - Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7/7.8, CentOS 7

```
$ sudo yum update -y
```

- Ubuntu 16.04 and Ubuntu 18.04

```
$ sudo apt-get update -y
```

```
$ sudo apt-get upgrade -y
```

3. Download the EFA software installation files. The software installation files are packaged into a compressed tarball (.tar.gz) file. To download the latest *stable* version, use the following command.

```
$ curl -O https://efa-installer.amazonaws.com/aws-efa-installer-1.10.0.tar.gz
```

You can also get the latest version by replacing the version number with `latest` in the preceding command.

4. (Optional) Verify the authenticity and integrity of the EFA tarball (.tar.gz) file. We recommend that you do this to verify the identity of the software publisher and to check that the file has not been altered or corrupted since it was published. If you do not want to verify the tarball file, skip this step.

Note

Alternatively, if you prefer to verify the tarball file by using an MD5 or SHA256 checksum instead, see [Verifying the EFA installer using a checksum \(p. 868\)](#).

- a. Download the public GPG key and import it into your keyring.

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer.key
```

```
$ gpg --import aws-efa-installer.key
```

The command should return a key value. Make a note of the key value, because you need it in the next step.

- b. Verify the GPG key's fingerprint. Run the following command and specify the key value from the previous step.

```
$ gpg --fingerprint key_value
```

The command should return a fingerprint that is identical to **4E90 91BC BB97 A96B 26B1 5E59 A054 80B1 DD2D 3CCC**. If the fingerprint does not match, don't run the EFA installation script, and contact AWS Support.

- c. Download the signature file and verify the signature of the EFA tarball file.

```
$ wget https://efa-installer.amazonaws.com/aws-efa-installer-1.10.0.tar.gz.sig
```

```
$ gpg --verify ./aws-efa-installer-1.10.0.tar.gz.sig
```

The following shows example output.

```
gpg: Signature made Wed 29 Jul 2020 12:50:13 AM UTC using RSA key ID DD2D3CCC
gpg: Good signature from "Amazon EC2 EFA <ec2-efa-maintainers@amazon.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:           There is no indication that the signature belongs to the owner.
Primary key fingerprint: 4E90 91BC BB97 A96B 26B1 5E59 A054 80B1 DD2D 3CCC
```

If the result includes **Good signature**, and the fingerprint matches the fingerprint returned in the previous step, proceed to the next step. If not, don't run the EFA installation script, and contact AWS Support.

5. Extract the files from the compressed **.tar.gz** file and navigate into the extracted directory.

```
$ tar -xf aws-efa-installer-1.10.0.tar.gz
```

```
$ cd aws-efa-installer
```

6. Run the EFA software installation script.

```
$ sudo ./efa_installer.sh -y
```

Libfabric is installed in the **/opt/amazon/efa** directory, while Open MPI is installed in the **/opt/amazon/openmpi** directory.

7. Log out of the instance and then log back in.
8. Confirm that the EFA software components were successfully installed.

```
$ fi_info -p efa
```

The command should return information about the Libfabric EFA interfaces. The following example shows the command output.

```
provider: efa
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-rdm
    version: 2.0
```

```
type: FI_EP_RDM
protocol: FI_PROTO_EFA
provider: efa
fabric: EFA-fe80::94:3dff:fe89:1b70
domain: efa_0-dgram
version: 2.0
type: FI_EP_DGRAM
protocol: FI_PROTO_EFA
provider: efa;ofi_rxrd
fabric: EFA-fe80::94:3dff:fe89:1b70
domain: efa_0-dgram
version: 1.0
type: FI_EP_RDM
protocol: FI_PROTO_RXD
```

Step 4: Install Nvidia GPU drivers and the Nvidia CUDA toolkit

To install the Nvidia GPU Drivers and the Nvidia CUDA toolkit

1. Install the utilities that are needed to install the Nvidia GPU drivers and the Nvidia CUDA toolkit.

- Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7/7.8, CentOS 7

```
$ sudo yum groupinstall 'Development Tools' -y
```

- Ubuntu 16.04 and Ubuntu 18.04

```
$ sudo apt-get install build-essential -y
```

2. To use the Nvidia GPU driver, you must first disable the nouveau open source drivers.

- a. Install the **gcc** compiler and the kernel headers package for the version of the kernel that you are currently running.

- Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7/7.8, CentOS 7

```
$ sudo yum install -y gcc kernel-devel-$(uname -r)
```

- Ubuntu 16.04 and Ubuntu 18.04

```
$ sudo apt-get install -y gcc make linux-headers-$(uname -r)
```

- b. Add nouveau to the `/etc/modprobe.d/blacklist.conf` deny list file.

```
$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

- c. Open `/etc/default/grub` using your preferred text editor and add the following.

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

- d. Rebuild the Grub configuration.

- Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7/7.8, CentOS 7

```
$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

- Ubuntu 16.04 and Ubuntu 18.04

```
$ sudo update-grub
```

- Reboot the instance and reconnect to it.
- Download the Nvidia CUDA Toolkit installer.

```
$ wget https://developer.download.nvidia.com/compute/cuda/11.0.3/local_installers/cuda_11.0.3_450.51.06_linux.run
```

- Run the Nvidia CUDA Toolkit installer.

```
$ sudo sh cuda_11.0.3_450.51.06_linux.run
```

When prompted to accept the license agreement, enter **accept** and press **Enter**.

- At the CUDA Installer menu, ensure that all of the items are selected, highlight **Install**, and then press **Enter**.
- Add the following statements to the shell startup scripts to ensure that the CUDA paths are set each time that the instance starts.

```
export PATH=/usr/local/cuda/bin:/usr/local/cuda/NsightCompute-2019.1${PATH:+:$PATH}  
export LD_LIBRARY_PATH=/usr/local/cuda/lib64\${LD_LIBRARY_PATH:+:$LD_LIBRARY_PATH}
```

- For *bash* shells, add the statements to */home/username/.bashrc* and */home/username/.bash_profile*.
 - For *tcsh* shells, add the statements to */home/username/.cshrc*.
- To confirm that the Nvidia GPU drivers are functional, run the following command.

```
$ nvidia-smi -q | head
```

The command should return information about the Nvidia GPUs, Nvidia GPU drivers, and Nvidia CUDA toolkit.

Step 5: Install cuDNN

Install the cuDNN library. The cuDNN library comes in two packages, a developer and runtime library. Depending on your use case, you might need to install only one, or both libraries.

To install cuDNN

- Navigate to your home directory.

```
$ cd $HOME
```

- Install the Nvidia machine learning repositories.

- Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7/7.8, CentOS 7

```
$ sudo yum install -y https://developer.download.nvidia.com/compute/machine-learning/repos/rhel7/x86_64/nvidia-machine-learning-repo-rhel7-1.0.0-1.x86_64.rpm
```

- Ubuntu 16.04 and Ubuntu 18.04

```
$ wget https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64/nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb
```

```
$ sudo apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64/7fa2af80.pub
```

```
$ sudo apt install ./nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb
```

3. Install the libraries.

- Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7/7.8, CentOS 7

```
$ sudo yum install -y libcudnn8 libcudnn8-devel
```

- Ubuntu 16.04 and Ubuntu 18.04

```
$ sudo apt update && sudo apt install libcudnn8 libcudnn8-dev -y
```

Step 6: Install NCCL

Install NCCL. For more information about NCCL, see the [NCCL repository](#).

To install NCCL

1. Navigate to the /opt directory.

```
$ cd /opt
```

2. Clone the official NCCL repository to the instance and navigate into the local cloned repository.

```
$ sudo git clone https://github.com/NVIDIA/nccl.git
```

```
$ cd nccl
```

3. Build and install NCCL and specify the CUDA installation directory.

```
$ sudo make -j src.build CUDA_HOME=/usr/local/cuda
```

Step 7: Install the aws-ofi-nccl plugin

The aws-ofi-nccl plugin maps NCCL's connection-oriented transport APIs to Libfabric's connection-less reliable interface. This enables you to use Libfabric as a network provider while running NCCL-based applications. For more information about the aws-ofi-nccl plugin, see the [aws-ofi-nccl repository](#).

To install the aws-ofi-nccl plugin

1. Navigate to your home directory.

```
$ cd $HOME
```

2. (Ubuntu 16.04 and Ubuntu 18.04) Install the utilities that are required to install the **aws-ofi-nccl** plugin. To install the required utilities, run the following command.

```
$ sudo apt-get install libtool autoconf -y
```

3. Clone the **aws** branch of the official AWS aws-ofi-nccl repository to the instance and navigate into the local cloned repository.

```
$ git clone https://github.com/aws/aws-ofi-nccl.git -b aws
```

```
$ cd aws-ofi-nccl
```

4. To generate the **configure** script, run the **autogen.sh** script.

```
$ ./autogen.sh
```

5. To generate the **make** files, run the **configure** script and specify the MPI, Libfabric, NCCL, and CUDA installation directories.

```
$ ./configure --prefix=/opt/aws-ofi-nccl --with-mpi=/opt/amazon/openmpi \  
--with-libfabric=/opt/amazon/efa --with-nccl=/opt/nccl/build \  
--with-cuda=/usr/local/cuda
```

6. Add the Open MPI directory to the **PATH** variable.

```
$ export PATH=/opt/amazon/openmpi/bin/:$PATH
```

7. Install the **aws-ofi-nccl** plugin.

```
$ make
```

```
$ sudo make install
```

Step 8: Install the NCCL tests

Install the NCCL tests. The NCCL tests enable you to confirm that NCCL is properly installed and that it is operating as expected. For more information about the NCCL tests, see the [nccl-tests repository](#).

To install the NCCL tests

1. Navigate to your home directory.

```
$ cd $HOME
```

2. Clone the official nccl-tests repository to the instance and navigate into the local cloned repository.

```
$ git clone https://github.com/NVIDIA/nccl-tests.git
```

```
$ cd nccl-tests
```

3. Add the Libfabric directory to the **LD_LIBRARY_PATH** variable.

- Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7/7.8, CentOS 7

```
$ export LD_LIBRARY_PATH=/opt/amazon/efa/lib64:$LD_LIBRARY_PATH
```

- Ubuntu 16.04 and Ubuntu 18.04

```
$ export LD_LIBRARY_PATH=/opt/amazon/efa/lib:$LD_LIBRARY_PATH
```

- Install the NCCL tests and specify the MPI, NCCL, and CUDA installation directories.

```
$ make MPI=1 MPI_HOME=/opt/amazon/openmpi NCCL_HOME=/opt/nccl/build CUDA_HOME=/usr/local/cuda
```

Step 9: Test your EFA and NCCL configuration

Run a test to ensure that your temporary instance is properly configured for EFA and NCCL.

To test your EFA and NCCL configuration

- Create a host file that specifies the hosts on which to run the tests. The following command creates a host file named my-hosts that includes a reference to the instance itself.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/local-ipv4 >> my-hosts
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4 >> my-hosts
```

- Run the test and specify the host file (--hostfile) and the number of GPUs to use (-n). The following command runs the all_reduce_perf test on 8 GPUs on the instance itself, and specifies the following environment variables.

- FI_PROVIDER="efa"—specifies the fabric interface provider. This must be set to "efa".
- FI_EFA_TX_MIN_CREDITS=64—specifies the minimum number of send credits that the sender requests from the receiver. 64 is the recommended value for NCCL jobs using EFA. The value should only be increased for message transfers that are larger than 256 MB.
- FI_EFA_ENABLE_SHM_TRANSFER=0—enables the SHM provider to provide the communication across all intra-node processes.
- NCCL_DEBUG=INFO—enables detailed debugging output. You can also specify VERSION to print only the NCCL version at the start of the test, or WARN to receive only error messages.
- NCCL_ALGO=ring—uses ring algorithm for collective operations.

For more information about the NCCL test arguments, see the [NCCL Tests README](#) in the official nccl-tests repository.

- Amazon Linux, Amazon Linux 2, RHEL 7.6/7.7/7.8, CentOS 7

```
$ /opt/amazon/openmpi/bin/mpirun \
-x FI_PROVIDER="efa" \
-x FI_EFA_TX_MIN_CREDITS=64 \
-x FI_EFA_ENABLE_SHM_TRANSFER=0 \
```

```
-x LD_LIBRARY_PATH=/opt/nccl/build/lib:/usr/local/cuda/lib64:/opt/amazon/efa/
lib64:/opt/amazon/openmpi/lib64:/opt/aws-ofi-nccl/lib:$LD_LIBRARY_PATH \
-x NCCL_DEBUG=INFO \
-x NCCL_ALGO=ring \
-x RDMAV_FORK_SAFE=1 \
--hostfile my-hosts -n 8 -N 8 \
--mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --bind-to none \
$HOME/nccl-tests/build/all_reduce_perf -b 8 -e 1G -f 2 -g 1 -c 1 -n 100
```

- Ubuntu 16.04 and Ubuntu 18.04

```
$ /opt/amazon/openmpi/bin/mpirun \
-x FI_PROVIDER="efa" \
-x FI_EFA_TX_MIN_CREDITS=64 \
-x FI_EFA_ENABLE_SHM_TRANSFER=0 \
-x LD_LIBRARY_PATH=/opt/nccl/build/lib:/usr/local/cuda/lib64:/opt/amazon/efa/
lib64:/opt/amazon/openmpi/lib64:/opt/aws-ofi-nccl/lib:$LD_LIBRARY_PATH \
-x NCCL_DEBUG=INFO \
-x NCCL_ALGO=ring \
-x RDMAV_FORK_SAFE=1 \
--hostfile my-hosts -n 8 -N 8 \
--mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --bind-to none \
$HOME/nccl-tests/build/all_reduce_perf -b 8 -e 1G -f 2 -g 1 -c 1 -n 100
```

3. You can confirm that EFA is active as the underlying provider for NCCL when the NCCL_DEBUG log is printed.

```
$ ip-192-168-2-54:14:14 [0] NCCL INFO NET/OFI Forcing AWS OFI ndev 4
*ip-192-168-2-54:14:14 [0] NCCL INFO NET/OFI Selected Provider is efa*
```

Step 10: Install your machine learning applications

Install the machine learning applications on the temporary instance. The installation procedure varies depending on the specific machine learning application. For more information about installing software on your Linux instance, see [Managing Software on Your Linux Instance](#).

Note

You might need to refer to your machine learning application's documentation for installation instructions.

Step 11: Create an EFA and NCCL-enabled AMI

After you have installed the required software components, you create an AMI that you can reuse to launch your EFA-enabled instances.

To create an AMI from your temporary instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the temporary instance that you created and choose **Actions, Image, Create image**.
4. For **Create image**, do the following:
 - a. For **Image name**, enter a descriptive name for the AMI.
 - b. (Optional) For **Image description**, enter a brief description of the purpose of the AMI.
 - c. Choose **Create image**.
5. In the navigation pane, choose **AMIs**.
6. Locate the AMI that you created in the list. Wait for the status to change from pending to available before continuing to the next step.

Step 12: Terminate the temporary instance

At this point, you no longer need the temporary instance that you launched. You can terminate the instance to stop incurring charges for it.

To terminate the temporary instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the temporary instance that you created and then choose **Actions, Instance state, Terminate instance**.
4. When prompted for confirmation, choose **Terminate**.

Step 13: Launch EFA and NCCL-enabled instances into a cluster placement group

Launch your EFA and NCCL-enabled instances into a cluster placement group using the EFA-enabled AMI and the EFA-enabled security group that you created earlier.

To launch your EFA and NCCL-enabled instances into a cluster placement group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. On the **Choose an AMI** page, choose **My AMIs**, find the AMI that you created earlier, and then choose **Select**.
4. On the **Choose an Instance Type** page, select **p3dn.24xlarge** and then choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, do the following:
 - a. For **Number of instances**, enter the number of EFA and NCCL-enabled instances that you want to launch.
 - b. For **Network** and **Subnet**, select the VPC and subnet into which to launch the instances.
 - c. For **Placement group**, select **Add instance to placement group**.
 - d. For **Placement group name**, select **Add to a new placement group**, and then enter a descriptive name for the placement group. Then for **Placement group strategy**, select **cluster**.
 - e. For **EFA**, choose **Enable**.
 - f. In the **Network Interfaces** section, for device **eth0**, choose **New network interface**. You can optionally specify a primary IPv4 address and one or more secondary IPv4 addresses. If you are launching the instance into a subnet that has an associated IPv6 CIDR block, you can optionally specify a primary IPv6 address and one or more secondary IPv6 addresses.
 - g. Choose **Next: Add Storage**.
6. On the **Add Storage** page, specify the volumes to attach to the instances in addition to the volumes specified by the AMI (such as the root device volume). Then choose **Next: Add Tags**.
7. On the **Add Tags** page, specify tags for the instances, such as a user-friendly name, and then choose **Next: Configure Security Group**.
8. On the **Configure Security Group** page, for **Assign a security group**, select **Select an existing security group**, and then select the security group that you created earlier.
9. Choose **Review and Launch**.
10. On the **Review Instance Launch** page, review the settings, and then choose **Launch** to choose a key pair and to launch your instances.

Step 14: Enable passwordless SSH

To enable your applications to run across all of the instances in your cluster, you must enable passwordless SSH access from the leader node to the member nodes. The leader node is the instance from which you run your applications. The remaining instances in the cluster are the member nodes.

To enable passwordless SSH between the instances in the cluster

1. Select one instance in the cluster as the leader node, and connect to it.
2. Disable `strictHostKeyChecking` and enable `ForwardAgent` on the leader node. Open `~/.ssh/config` using your preferred text editor and add the following.

```
Host *
  ForwardAgent yes
Host *
  StrictHostKeyChecking no
```

3. Generate an RSA key pair.

```
$ ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
```

The key pair is created in the `$HOME/.ssh/` directory.

4. Change the permissions of the private key on the leader node.

```
$ chmod 600 ~/.ssh/id_rsa
chmod 600 ~/.ssh/config
```

5. Open `~/.ssh/id_rsa.pub` using your preferred text editor and copy the key.
6. For each member node in the cluster, do the following:
 - a. Connect to the instance.
 - b. Open `~/.ssh/authorized_keys` using your preferred text editor and add the public key that you copied earlier.
7. To test that the passwordless SSH is functioning as expected, connect to your leader node and run the following command.

```
$ ssh member_node_private_ip
```

You should connect to the member node without being prompted for a key or password.

Using an AWS Deep Learning AMI

The following steps help you to get started with one of the following AWS Deep Learning AMIs:

- Deep Learning AMI (Amazon Linux 2) Version 25.0 and later
- Deep Learning AMI (Amazon Linux) Version 25.0 and later
- Deep Learning AMI (Ubuntu 18.04) Version 25.0 and later
- Deep Learning AMI (Ubuntu 16.04) Version 25.0 and later

For more information, see the [AWS Deep Learning AMI User Guide](#).

Contents

- [Step 1: Prepare an EFA-enabled security group \(p. 861\)](#)

- Step 2: Launch a temporary instance (p. 861)
- Step 3: Test your EFA and NCCL configuration (p. 862)
- Step 4: Install your machine learning applications (p. 863)
- Step 5: Create an EFA and NCCL-enabled AMI (p. 863)
- Step 6: Terminate the temporary instance (p. 863)
- Step 7: Launch EFA and NCCL-enabled instances into a cluster placement group (p. 863)
- Step 8: Enable passwordless SSH (p. 864)

Step 1: Prepare an EFA-enabled security group

An EFA requires a security group that allows all inbound and outbound traffic to and from the security group itself.

To create an EFA-enabled security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups** and then choose **Create Security Group**.
3. In the **Create Security Group** window, do the following:
 - a. For **Security group name**, enter a descriptive name for the security group, such as **EFA-enabled security group**.
 - b. (Optional) For **Description**, enter a brief description of the security group.
 - c. For **VPC**, select the VPC into which you intend to launch your EFA-enabled instances.
 - d. Choose **Create**.
4. Select the security group that you created, and on the **Description** tab, copy the **Group ID**.
5. On the **Inbound** tab, do the following:
 - a. Choose **Edit**.
 - b. For **Type**, choose **All traffic**.
 - c. For **Source**, choose **Custom** and paste the security group ID that you copied into the field.
 - d. Choose **Save**.
6. On the **Outbound** tab, do the following:
 - a. Choose **Edit**.
 - b. For **Type**, choose **All traffic**.
 - c. For **Destination**, choose **Custom** and paste the security group ID that you copied into the field.
 - d. Choose **Save**.

Step 2: Launch a temporary instance

Launch a temporary instance that you can use to install and configure the EFA software components. You use this instance to create an EFA-enabled AMI from which you can launch your EFA-enabled instances.

To launch a temporary instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. On the **Choose an AMI** page, choose a supported **AWS Deep Learning AMI Version 25.0** or later.
4. On the **Choose an Instance Type** page, select **p3dn.24xlarge** and then choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, do the following:

- a. For **Subnet**, choose the subnet in which to launch the instance.
- b. For **Elastic Fabric Adapter**, choose **Enable**.
- c. In the **Network Interfaces** section, for device **eth0**, choose **New network interface**.
- d. Choose **Next: Add Storage**.
6. On the **Add Storage** page, specify the volumes to attach to the instances, in addition to the volumes specified by the AMI (such as the root device volume). Then choose **Next: Add Tags**.
7. On the **Add Tags** page, specify a tag that you can use to identify the temporary instance, and then choose **Next: Configure Security Group**.
8. On the **Configure Security Group** page, for **Assign a security group**, select **Select an existing security group**. Then select the security group that you created in **Step 1**.
9. On the **Review Instance Launch** page, review the settings, and then choose **Launch** to choose a key pair and to launch your instance.

Step 3: Test your EFA and NCCL configuration

Run a test to ensure that your temporary instance is properly configured for EFA and NCCL.

To test your EFA and NCCL configuration

1. Create a host file that specifies the hosts on which to run the tests. The following command creates a host file named `my-hosts` that includes a reference to the instance itself.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/local-ipv4 >> my-hosts
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4 >> my-hosts
```

2. Run the test and specify the host file (`--hostfile`) and the number of GPUs to use (`-n`). The following command runs the `all_reduce_perf` test on 8 GPUs on the instance itself, and specifies the following environment variables.
 - `FI_PROVIDER="efa"`—specifies the fabric interface provider. This must be set to "efa".
 - `FI_EFA_TX_MIN_CREDITS=64`—specifies the minimum number of send credits that the sender requests from the receiver. 64 is the recommended value for NCCL jobs using EFA. The value should only be increased for message transfers that are larger than 256 MB.
 - `NCCL_DEBUG=INFO`—enables detailed debugging output. You can also specify `VERSION` to print only the NCCL version at the start of the test, or `WARN` to receive only error messages.
 - `NCCL_TREE_THRESHOLD=0`—disables tree algorithms for the test.

For more information about the NCCL test arguments, see the [NCCL Tests README](#) in the official `nccl-tests` repository.

```
/opt/amazon/openmpi/bin/mpirun \
-x FI_PROVIDER="efa" \
-x FI_EFA_TX_MIN_CREDITS=64 \
-x NCCL_DEBUG=INFO \
-x NCCL_TREE_THRESHOLD=0 \
```

```
--hostfile my-hosts -n 8 -N 8 \
--mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --bind-to none \
$HOME/src/bin/efa-tests/efa-cuda-10.0/all_reduce_perf -b 8 -e 1G -f 2 -g 1 -c 1 -n 100
```

Step 4: Install your machine learning applications

Install the machine learning applications on the temporary instance. The installation procedure varies depending on the specific machine learning application. For more information about installing software on your Linux instance, see [Managing Software on Your Linux Instance](#).

Note

You might need to refer to your machine learning application's documentation for installation instructions.

Step 5: Create an EFA and NCCL-enabled AMI

After you have installed the required software components, you create an AMI that you can reuse to launch your EFA-enabled instances.

To create an AMI from your temporary instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the temporary instance that you created and choose **Actions, Image, Create image**.
4. For **Create image**, do the following:
 - a. For **Image name**, enter a descriptive name for the AMI.
 - b. (Optional) For **Image description**, enter a brief description of the purpose of the AMI.
 - c. Choose **Create image**.
5. In the navigation pane, choose **AMIs**.
6. Locate the AMI that you created in the list. Wait for the status to change from pending to available before continuing to the next step.

Step 6: Terminate the temporary instance

At this point, you no longer need the temporary instance that you launched. You can terminate the instance to stop incurring charges for it.

To terminate the temporary instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the temporary instance that you created and then choose **Actions, Instance state, Terminate instance**.
4. When prompted for confirmation, choose **Terminate**.

Step 7: Launch EFA and NCCL-enabled instances into a cluster placement group

Launch your EFA and NCCL-enabled instances into a cluster placement group using the EFA-enabled AMI and the EFA-enabled security group that you created earlier.

To launch your EFA and NCCL-enabled instances into a cluster placement group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. Choose **Launch Instance**.
3. On the **Choose an AMI** page, choose **My AMIs**, find the AMI that you created earlier, and then choose **Select**.
4. On the **Choose an Instance Type** page, select **p3dn.24xlarge** and then choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, do the following:
 - a. For **Number of instances**, enter the number of EFA and NCCL-enabled instances that you want to launch.
 - b. For **Network and Subnet**, select the VPC and subnet into which to launch the instances.
 - c. For **Placement group**, select **Add instance to placement group**.
 - d. For **Placement group name**, select **Add to a new placement group**, and then enter a descriptive name for the placement group. Then for **Placement group strategy**, select **cluster**.
 - e. For **EFA**, choose **Enable**.
 - f. In the **Network Interfaces** section, for device **eth0**, choose **New network interface**. You can optionally specify a primary IPv4 address and one or more secondary IPv4 addresses. If you are launching the instance into a subnet that has an associated IPv6 CIDR block, you can optionally specify a primary IPv6 address and one or more secondary IPv6 addresses.
 - g. Choose **Next: Add Storage**.
6. On the **Add Storage** page, specify the volumes to attach to the instances in addition to the volumes specified by the AMI (such as the root device volume). Then choose **Next: Add Tags**.
7. On the **Add Tags** page, specify tags for the instances, such as a user-friendly name, and then choose **Next: Configure Security Group**.
8. On the **Configure Security Group** page, for **Assign a security group**, select **Select an existing security group**, and then select the security group that you created earlier.
9. Choose **Review and Launch**.
10. On the **Review Instance Launch** page, review the settings, and then choose **Launch** to choose a key pair and to launch your instances.

Step 8: Enable passwordless SSH

To enable your applications to run across all of the instances in your cluster, you must enable passwordless SSH access from the leader node to the member nodes. The leader node is the instance from which you run your applications. The remaining instances in the cluster are the member nodes.

To enable passwordless SSH between the instances in the cluster

1. Select one instance in the cluster as the leader node, and connect to it.
2. Disable `strictHostKeyChecking` and enable `ForwardAgent` on the leader node. Open `~/.ssh/config` using your preferred text editor and add the following.

```
Host *
  ForwardAgent yes
Host *
  StrictHostKeyChecking no
```

3. Generate an RSA key pair.

```
$ ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
```

The key pair is created in the `$HOME/.ssh/` directory.

4. Change the permissions of the private key on the leader node.

```
$ chmod 600 ~/.ssh/id_rsa
chmod 600 ~/.ssh/config
```

5. Open `~/.ssh/id_rsa.pub` using your preferred text editor and copy the key.
6. For each member node in the cluster, do the following:
 - a. Connect to the instance.
 - b. Open `~/.ssh/authorized_keys` using your preferred text editor and add the public key that you copied earlier.
7. To test that the passwordless SSH is functioning as expected, connect to your leader node and run the following command.

```
$ ssh member_node_private_ip
```

You should connect to the member node without being prompted for a key or password.

Working with EFA

You can create, use, and manage an EFA much like any other elastic network interface in Amazon EC2. However, unlike elastic network interfaces, EFAs cannot be attached to or detached from an instance in a running state.

EFA requirements

To use an EFA, you must do the following:

- Choose one of the [supported instance types](#) (p. 840).
- Use one of the [supported AMIs](#) (p. 840).
- Install the EFA software components. For more information, see [Step 3: Install the EFA software](#) (p. 842) and [Step 5: \(Optional\) Install Intel MPI](#) (p. 845).
- Use a security group that allows all inbound and outbound traffic to and from the security group itself. For more information, see [Step 1: Prepare an EFA-enabled security group](#) (p. 841).

Contents

- [Creating an EFA](#) (p. 865)
- [Attaching an EFA to a stopped instance](#) (p. 866)
- [Attaching an EFA when launching an instance](#) (p. 866)
- [Adding an EFA to a launch template](#) (p. 866)
- [Managing IP addresses for an EFA](#) (p. 867)
- [Changing the security group for an EFA](#) (p. 867)
- [Detaching an EFA](#) (p. 867)
- [Viewing EFAs](#) (p. 867)
- [Deleting an EFA](#) (p. 867)

Creating an EFA

You can create an EFA in a subnet in a VPC. You can't move the EFA to another subnet after it's created, and you can only attach it to stopped instances in the same Availability Zone.

To create a new EFA using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Choose **Create Network Interface**.
4. For **Description**, enter a descriptive name for the EFA.
5. For **Subnet**, select the subnet in which to create the EFA.
6. For **Private IP**, enter the primary private IPv4 address. If you don't specify an IPv4 address, we select an available private IPv4 address from the selected subnet.
7. (IPv6 only) If you selected a subnet that has an associated IPv6 CIDR block, you can optionally specify an IPv6 address in the **IPv6 IP** field.
8. For **Security groups**, select one or more security groups.
9. For **EFA**, choose **Enabled**.
10. Choose **Yes, Create**.

To create a new EFA using the AWS CLI

Use the `create-network-interface` command and for `interface-type`, specify `efa`, as shown in the following example.

```
aws ec2 create-network-interface --subnet-id subnet-01234567890 --description example_efa  
--interface-type efa
```

Attaching an EFA to a stopped instance

You can attach an EFA to any supported instance that is in the `stopped` state. You cannot attach an EFA to an instance that is in the `running` state. For more information about the supported instance types, see [Supported instance types \(p. 840\)](#).

You attach an EFA to an instance in the same way that you attach a network interface to an instance. For more information, see [Attaching a network interface to an instance \(p. 804\)](#).

Attaching an EFA when launching an instance

To attach an existing EFA when launching an instance (AWS CLI)

Use the `run-instances` command and for `NetworkInterfaceId`, specify the ID of the EFA, as shown in the following example.

```
aws ec2 run-instances --image-id ami_id --count 1 --instance-  
type c5n.18xlarge --key-name my_key_pair --network-interfaces  
DeviceIndex=0,NetworkInterfaceId=efa_id,Groups=sg_id,SubnetId=subnet_id
```

To attach a new EFA when launching an instance (AWS CLI)

Use the `run-instances` command and for `InterfaceType`, specify `efa`, as shown in the following example.

```
aws ec2 run-instances --image-id ami_id --count 1 --instance-  
type c5n.18xlarge --key-name my_key_pair --network-interfaces  
DeviceIndex=0,InterfaceType=efa,Groups=sg_id,SubnetId=subnet_id
```

Adding an EFA to a launch template

You can create a launch template that contains the configuration information needed to launch EFA-enabled instances. To create an EFA-enabled launch template, create a new launch template and

specify a supported instance type, your EFA-enabled AMI, and an EFA-enabled security group. For more information, see [Getting started with EFA and MPI \(p. 841\)](#).

You can leverage launch templates to launch EFA-enabled instances with other AWS services, such as AWS Batch.

For more information about creating launch templates, see [Creating a launch template \(p. 506\)](#).

Managing IP addresses for an EFA

You can change the IP addresses associated with an EFA. If you have an Elastic IP address, you can associate it with an EFA. If your EFA is provisioned in a subnet that has an associated IPv6 CIDR block, you can assign one or more IPv6 addresses to the EFA.

You assign an Elastic IP (IPv4) and IPv6 address to an EFA in the same way that you assign an IP address to an elastic network interface. For more information, see [Managing IP addresses \(p. 805\)](#).

Changing the security group for an EFA

You can change the security group that is associated with an EFA. To enable OS-bypass functionality, the EFA must be a member of a security group that allows all inbound and outbound traffic to and from the security group itself.

You change the security group that is associated with an EFA in the same way that you change the security group that is associated with an elastic network interface. For more information, see [Changing the security group \(p. 807\)](#).

Detaching an EFA

To detach an EFA from an instance, you must first stop the instance. You cannot detach an EFA from an instance that is in the running state.

You detach an EFA from an instance in the same way that you detach an elastic network interface from an instance. For more information, see [Detaching a network interface from an instance \(p. 805\)](#).

Viewing EFAs

You can view all of the EFAs in your account.

You view EFAs in the same way that you view elastic network interfaces. For more information, see [Viewing details about a network interface \(p. 803\)](#).

Deleting an EFA

To delete an EFA, you must first detach it from the instance. You cannot delete an EFA while it is attached to an instance.

You delete EFAs in the same way that you delete elastic network interfaces. For more information, see [Deleting a network interface \(p. 808\)](#).

Monitoring an EFA

You can use the following features to monitor the performance of your Elastic Fabric Adapters.

Amazon VPC flow logs

You can create an Amazon VPC Flow Log to capture information about the traffic going to and from an EFA. Flow log data can be published to Amazon CloudWatch Logs and Amazon S3. After you create a

flow log, you can retrieve and view its data in the chosen destination. For more information, see [VPC Flow Logs](#) in the *Amazon VPC User Guide*.

You create a flow log for an EFA in the same way that you create a flow log for an elastic network interface. For more information, see [Creating a Flow Log](#) in the *Amazon VPC User Guide*.

In the flow log entries, EFA traffic is identified by the `srcAddress` and `destAddress`, which are both formatted as MAC addresses, as shown in the following example.

```
version accountId eniId      srcAddress      destAddress      sourcePort destPort
  protocol packets bytes start      end      action log-status
2          3794735123 eni-10000001 01:23:45:67:89:ab 05:23:45:67:89:ab -           -
9          5689     1521232534 1524512343 ACCEPT OK
```

Amazon CloudWatch

Amazon CloudWatch provides metrics that enable you to monitor your EFAs in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For more information, see [Monitoring your instances using CloudWatch \(p. 715\)](#).

Verifying the EFA installer using a checksum

You can optionally verify the EFA tarball (.tar.gz file) using an MD5 or SHA256 checksum. We recommend that you do this to verify the identity of the software publisher and to check that the application has not been altered or corrupted since it was published.

To verify the tarball

Use the **md5sum** utility for the MD5 checksum, or the **sha256sum** utility for the SHA256 checksum, and specify the tarball filename. You must run the command from the directory in which you saved the tarball file.

- MD5

```
$ md5sum tarball_filename.tar.gz
```

- SHA256

```
$ sha256sum tarball_filename.tar.gz
```

The commands should return a checksum value in the following format.

```
checksum_value tarball_filename.tar.gz
```

Compare the checksum value returned by the command with the checksum value provided in the table below. If the checksums match, then it is safe to run the installation script. If the checksums do not match, do not run the installation script, and contact AWS Support.

For example, the following command verifies the EFA 1.9.4 tarball using the SHA256 checksum.

```
$ sha256sum aws-efa-installer-1.9.4.tar.gz
```

```
1009b5182693490d908ef0ed2c1dd4f813cc310a5d2062ce9619c4c12b5a7f14 aws-efa-
installer-1.9.4.tar.gz
```

The following table lists the checksums for recent versions of EFA.

Version	Download URL	Checksums
EFA 1.10.0	https://efa-installer.amazonaws.com/aws-efa-installer-1.10.0.tar.gz	MD5: 46f73f5a7afe41b4bb918c81888fef9 SHA256: 136612f96f2a085a7d98296da0afb6fa807b38142
EFA 1.9.5	https://efa-installer.amazonaws.com/aws-efa-installer-1.9.5.tar.gz	MD5: 95edb8a209c18ba8d250409846eb6ef4 SHA256: a4343308d7ea4dc943ccc21bcebed913e8868e59b
EFA 1.9.4	https://efa-installer.amazonaws.com/aws-efa-installer-1.9.4.tar.gz	MD5: f26dd5c350422c1a985e35947fa5aa28 SHA256: 1009b5182693490d908ef0ed2c1dd4f813cc310a5
EFA 1.9.3	https://efa-installer.amazonaws.com/aws-efa-installer-1.9.3.tar.gz	MD5: 95755765a097802d3e6d5018d1a5d3d6 SHA256: 46ce732d6f3fcc9edf6a6e9f9df0ad136054328e2
EFA 1.8.4	https://efa-installer.amazonaws.com/aws-efa-installer-1.8.4.tar.gz	MD5: 85d594c41e831afc6c9305263140457e SHA256: 0d974655a09b213d7859e658965e56dc4f23a0eee

Placement groups

When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures. You can use *placement groups* to influence the placement of a group of *interdependent* instances to meet the needs of your workload. Depending on the type of workload, you can create a placement group using one of the following placement strategies:

- *Cluster* – packs instances close together inside an Availability Zone. This strategy enables workloads to achieve the low-latency network performance necessary for tightly-coupled node-to-node communication that is typical of HPC applications.
- *Partition* – spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions. This strategy is typically used by large distributed and replicated workloads, such as Hadoop, Cassandra, and Kafka.
- *Spread* – strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.

There is no charge for creating a placement group.

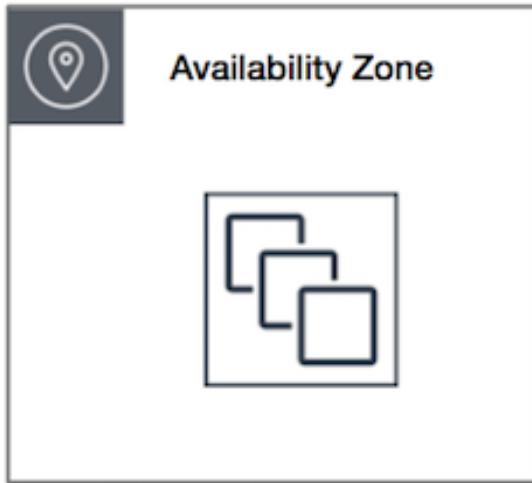
Contents

- [Cluster placement groups \(p. 870\)](#)
- [Partition placement groups \(p. 871\)](#)
- [Spread placement groups \(p. 871\)](#)
- [Placement group rules and limitations \(p. 872\)](#)
- [Creating a placement group \(p. 873\)](#)
- [Tagging a placement group \(p. 874\)](#)
- [Launching instances in a placement group \(p. 876\)](#)
- [Describing instances in a placement group \(p. 877\)](#)
- [Changing the placement group for an instance \(p. 879\)](#)
- [Deleting a placement group \(p. 880\)](#)

Cluster placement groups

A cluster placement group is a logical grouping of instances within a single Availability Zone. A cluster placement group can span peered VPCs in the same Region. Instances in the same cluster placement group enjoy a higher per-flow throughput limit for TCP/IP traffic and are placed in the same high-bisection bandwidth segment of the network.

The following image shows instances that are placed into a cluster placement group.



Cluster placement groups are recommended for applications that benefit from low network latency, high network throughput, or both. They are also recommended when the majority of the network traffic is between the instances in the group. To provide the lowest latency and the highest packet-per-second network performance for your placement group, choose an instance type that supports enhanced networking. For more information, see [Enhanced Networking \(p. 813\)](#).

We recommend that you launch your instances in the following way:

- Use a single launch request to launch the number of instances that you need in the placement group.
- Use the same instance type for all instances in the placement group.

If you try to add more instances to the placement group later, or if you try to launch more than one instance type in the placement group, you increase your chances of getting an insufficient capacity error.

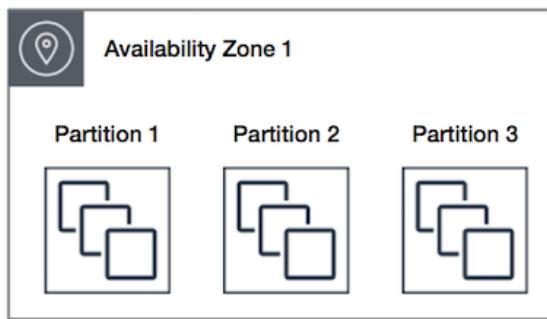
If you stop an instance in a placement group and then start it again, it still runs in the placement group. However, the start fails if there isn't enough capacity for the instance.

If you receive a capacity error when launching an instance in a placement group that already has running instances, stop and start all of the instances in the placement group, and try the launch again. Starting the instances may migrate them to hardware that has capacity for all of the requested instances.

Partition placement groups

Partition placement groups help reduce the likelihood of correlated hardware failures for your application. When using partition placement groups, Amazon EC2 divides each group into logical segments called partitions. Amazon EC2 ensures that each partition within a placement group has its own set of racks. Each rack has its own network and power source. No two partitions within a placement group share the same racks, allowing you to isolate the impact of hardware failure within your application.

The following image is a simple visual representation of a partition placement group in a single Availability Zone. It shows instances that are placed into a partition placement group with three partitions—**Partition 1**, **Partition 2**, and **Partition 3**. Each partition comprises multiple instances. The instances in a partition do not share racks with the instances in the other partitions, allowing you to contain the impact of a single hardware failure to only the associated partition.



Partition placement groups can be used to deploy large distributed and replicated workloads, such as HDFS, HBase, and Cassandra, across distinct racks. When you launch instances into a partition placement group, Amazon EC2 tries to distribute the instances evenly across the number of partitions that you specify. You can also launch instances into a specific partition to have more control over where the instances are placed.

A partition placement group can have partitions in multiple Availability Zones in the same Region. A partition placement group can have a maximum of seven partitions per Availability Zone. The number of instances that can be launched into a partition placement group is limited only by the limits of your account.

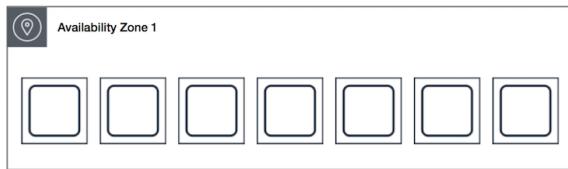
In addition, partition placement groups offer visibility into the partitions — you can see which instances are in which partitions. You can share this information with topology-aware applications, such as HDFS, HBase, and Cassandra. These applications use this information to make intelligent data replication decisions for increasing data availability and durability.

If you start or launch an instance in a partition placement group and there is insufficient unique hardware to fulfill the request, the request fails. Amazon EC2 makes more distinct hardware available over time, so you can try your request again later.

Spread placement groups

A spread placement group is a group of instances that are each placed on distinct racks, with each rack having its own network and power source.

The following image shows seven instances in a single Availability Zone that are placed into a spread placement group. The seven instances are placed on seven different racks.



Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other. Launching instances in a spread placement group reduces the risk of simultaneous failures that might occur when instances share the same racks. Spread placement groups provide access to distinct racks, and are therefore suitable for mixing instance types or launching instances over time.

A spread placement group can span multiple Availability Zones in the same Region. You can have a maximum of seven running instances per Availability Zone per group.

If you start or launch an instance in a spread placement group and there is insufficient unique hardware to fulfill the request, the request fails. Amazon EC2 makes more distinct hardware available over time, so you can try your request again later.

Placement group rules and limitations

General rules and limitations

Before you use placement groups, be aware of the following rules:

- The name that you specify for a placement group must be unique within your AWS account for the Region.
- You can't merge placement groups.
- An instance can be launched in one placement group at a time; it cannot span multiple placement groups.
- [On-Demand Capacity Reservation \(p. 476\)](#) and [zonal Reserved Instances \(p. 311\)](#) provide a capacity reservation for EC2 instances in a specific Availability Zone. The capacity reservation can be used by instances in a placement group. However, it is not possible to explicitly reserve capacity for a placement group.
- You cannot launch Dedicated Hosts in placement groups.

Cluster placement group rules and limitations

The following rules apply to cluster placement groups:

- Instances in a cluster placement group you must use the following supported instance types:
 - [Current generation \(p. 201\)](#) instances, except for the [burstable performance \(p. 219\)](#) instances (for example, T2).
 - The following [previous generation \(p. 204\)](#) instances: C3, cc2.8xlarge, cr1.8xlarge, G2, hs1.8xlarge, I2, and R3.
- A cluster placement group can't span multiple Availability Zones.
- The maximum network throughput speed of traffic between two instances in a cluster placement group is limited by the slower of the two instances. For applications with high-throughput requirements, choose an instance type with network connectivity that meets your requirements.
- For instances that are enabled for enhanced networking, the following rules apply:
 - Instances within a cluster placement group can use up to 10 Gbps for single-flow traffic. Instances that are not within a cluster placement group can use up to 5 Gbps for single-flow traffic.

- Traffic to and from Amazon S3 buckets within the same Region over the public IP address space or through a VPC endpoint can use all available instance aggregate bandwidth.
- You can launch multiple instance types into a cluster placement group. However, this reduces the likelihood that the required capacity will be available for your launch to succeed. We recommend using the same instance type for all instances in a cluster placement group.
- Network traffic to the internet and over an AWS Direct Connect connection to on-premises resources is limited to 5 Gbps.

Partition placement group rules and limitations

The following rules apply to partition placement groups:

- A partition placement group supports a maximum of seven partitions per Availability Zone. The number of instances that you can launch in a partition placement group is limited only by your account limits.
- When instances are launched into a partition placement group, Amazon EC2 tries to evenly distribute the instances across all partitions. Amazon EC2 doesn't guarantee an even distribution of instances across all partitions.
- A partition placement group with Dedicated Instances can have a maximum of two partitions.

Spread placement group rules and limitations

The following rules apply to spread placement groups:

- A spread placement group supports a maximum of seven running instances per Availability Zone. For example, in a Region with three Availability Zones, you can run a total of 21 instances in the group (seven per zone). If you try to start an eighth instance in the same Availability Zone and in the same spread placement group, the instance will not launch. If you need to have more than seven instances in an Availability Zone, then the recommendation is to use multiple spread placement groups. Using multiple spread placement groups does not provide guarantees about the spread of instances between groups, but it does ensure the spread for each group, thus limiting impact from certain classes of failures.
- Spread placement groups are not supported for Dedicated Instances.

Creating a placement group

You can create a placement group using one of the following methods.

Note

You can tag a placement group on creation using the command line tools only.

New console

To create a placement group using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Placement Groups**, **Create placement group**.
3. Specify a name for the group.
4. Choose the placement strategy for the group. If you choose **Partition**, choose the number of partitions within the group.
5. Choose **Create group**.

Old console

To create a placement group using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Placement Groups**, **Create Placement Group**.
3. Specify a name for the group.
4. Choose the placement strategy for the group. If you choose **Partition**, specify the number of partitions within the group.
5. Choose **Create**.

AWS CLI

To create a placement group using the AWS CLI

Use the [create-placement-group](#) command. The following example creates a placement group named `my-cluster` that uses the `cluster` placement strategy, and it applies a tag with a key of `purpose` and a value of `production`.

```
aws ec2 create-placement-group --group-name my-cluster --strategy cluster --tag-specifications 'ResourceType=placement-group,Tags={Key=purpose,Value=production}'
```

To create a partition placement group using the AWS CLI

Use the [create-placement-group](#) command. Specify the `--strategy` parameter with the value `partition`, and specify the `--partition-count` parameter with the desired number of partitions. In this example, the partition placement group is named `HDFS-Group-A` and is created with five partitions.

```
aws ec2 create-placement-group --group-name HDFS-Group-A --strategy partition --partition-count 5
```

PowerShell

To create a placement group using the AWS Tools for Windows PowerShell

Use the [New-EC2PlacementGroup](#) command.

Tagging a placement group

To help categorize and manage your existing placement groups, you can tag them with custom metadata. For more information about how tags work, see [Tagging your Amazon EC2 resources \(p. 1229\)](#).

When you tag a placement group, the instances that are launched into the placement group are not automatically tagged. You need to explicitly tag the instances that are launched into the placement group. For more information, see [Adding a tag when you launch an instance \(p. 1237\)](#).

You can view, add, and delete tags using the *new* console and the command line tools.

New console

To view, add, or delete a tag for an existing placement group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Placement Groups**.

3. Select a placement group, and then choose **Actions, Manage tags**.
4. The **Manage tags** section displays any tags that are assigned to the placement group. Do the following to add or remove tags:
 - To add a tag, choose **Add tag**, and then enter the tag key and value. You can add up to 50 tags per placement group. For more information, see [Tag restrictions \(p. 1233\)](#).
 - To delete a tag, choose **Remove** next to the tag that you want to delete.
5. Choose **Save changes**.

AWS CLI

To view placement group tags

Use the [describe-tags](#) command to view the tags for the specified resource. In the following example, you describe the tags for all of your placement groups.

```
aws ec2 describe-tags \
--filters Name=resource-type,Values=placement-group
```

```
{
  "Tags": [
    {
      "Key": "Environment",
      "ResourceId": "pg-0123456789EXAMPLE",
      "ResourceType": "placement-group",
      "Value": "Production"
    },
    {
      "Key": "Environment",
      "ResourceId": "pg-9876543210EXAMPLE",
      "ResourceType": "placement-group",
      "Value": "Production"
    }
  ]
}
```

You can also use the [describe-tags](#) command to view the tags for a placement group by specifying its ID. In the following example, you describe the tags for pg-0123456789EXAMPLE.

```
aws ec2 describe-tags \
--filters Name=resource-id,Values=pg-0123456789EXAMPLE
```

```
{
  "Tags": [
    {
      "Key": "Environment",
      "ResourceId": "pg-0123456789EXAMPLE",
      "ResourceType": "placement-group",
      "Value": "Production"
    }
  ]
}
```

You can also view the tags of a placement group by describing the placement group.

Use the [describe-placement-groups](#) command to view the configuration of the specified placement group, which includes any tags that were specified for the placement group.

```
aws ec2 describe-placement-groups \
--group-name my-cluster
```



```
{
    "PlacementGroups": [
        {
            "GroupName": "my-cluster",
            "State": "available",
            "Strategy": "cluster",
            "GroupId": "pg-0123456789EXAMPLE",
            "Tags": [
                {
                    "Key": "Environment",
                    "Value": "Production"
                }
            ]
        }
    ]
}
```

To tag an existing placement group using the AWS CLI

You can use the [create-tags](#) command to tag existing resources. In the following example, the existing placement group is tagged with Key=Cost-Center and Value=CC-123.

```
aws ec2 create-tags \
--resources pg-0123456789EXAMPLE \
--tags Key=Cost-Center,Value=CC-123
```

To delete a tag from a placement group using the AWS CLI

You can use the [delete-tags](#) command to delete tags from existing resources. For examples, see [Examples](#) in the *AWS CLI Command Reference*.

PowerShell

To view placement group tags

Use the [Get-EC2Tag](#) command.

To describe the tags for a specific placement group

Use the [Get-EC2PlacementGroup](#) command.

To tag an existing placement group

Use the [New-EC2Tag](#) command.

To delete a tag from a placement group

Use the [Remove-EC2Tag](#) command.

Launching instances in a placement group

You can launch an instance into a placement group if the [placement group rules and limitations are met \(p. 872\)](#) using one of the following methods.

Console

To launch instances into a placement group using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Choose **Launch Instance**. Complete the wizard as directed, taking care to do the following:
 - On the **Choose an Instance Type** page, select an instance type that can be launched into a placement group.
 - On the **Configure Instance Details** page, the following fields are applicable to placement groups:
 - For **Number of instances**, enter the total number of instances that you need in this placement group, because you might not be able to add instances to the placement group later.
 - For **Placement group**, select the **Add instance to placement group** check box. If you do not see **Placement group** on this page, verify that you have selected an instance type that can be launched into a placement group. Otherwise, this option is not available.
 - For **Placement group name**, you can choose to add the instances to an existing placement group or to a new placement group that you create.
 - For **Placement group strategy**, choose the appropriate strategy. If you choose **partition**, for **Target partition**, choose **Auto distribution** to have Amazon EC2 do a best effort to distribute the instances evenly across all the partitions in the group. Alternatively, specify the partition in which to launch the instances.

AWS CLI

To launch instances into a placement group using the AWS CLI

Use the `run-instances` command and specify the placement group name using the `--placement "GroupName = my-cluster"` parameter. In this example, the placement group is named `my-cluster`.

```
aws ec2 run-instances --placement "GroupName = my-cluster"
```

To launch instances into a specific partition of a partition placement group using the AWS CLI

Use the `run-instances` command and specify the placement group name and partition using the `--placement "GroupName = HDFS-Group-A, PartitionNumber = 3"` parameter. In this example, the placement group is named `HDFS-Group-A` and the partition number is 3.

```
aws ec2 run-instances --placement "GroupName = HDFS-Group-A, PartitionNumber = 3"
```

PowerShell

To launch instances into a placement group using AWS Tools for Windows PowerShell

Use the `New-EC2Instance` command and specify the placement group name using the `-Placement_GroupName` parameter.

Describing instances in a placement group

You can view the placement information of your instances using one of the following methods. You can also filter partition placement groups by the partition number using the AWS CLI.

New console

To view the placement group and partition number of an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. In the **Description** tab, under **Host and placement group**, find **Placement group**. If the instance is not in a placement group, the field is empty. Otherwise, it contains the name of the placement group name. If the placement group is a partition placement group, **Partition number** contains the partition number for the instance.

Old console

To view the placement group and partition number of an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. In the **Description** tab, find **Placement group**. If the instance is not in a placement group, the field is empty. Otherwise, it contains the name of the placement group name. If the placement group is a partition placement group, **Partition number** contains the partition number for the instance.

AWS CLI

To view the partition number for an instance in a partition placement group using the AWS CLI

Use the [describe-instances](#) command and specify the `--instance-id` parameter.

```
aws ec2 describe-instances --instance-id i-0123a456700123456
```

The response contains the placement information, which includes the placement group name and the partition number for the instance.

```
"Placement": {  
    "AvailabilityZone": "us-east-1c",  
    "GroupName": "HDFS-Group-A",  
    "PartitionNumber": 3,  
    "Tenancy": "default"  
}
```

To filter instances for a specific partition placement group and partition number using the AWS CLI

Use the [describe-instances](#) command and specify the `--filters` parameter with the `placement-group-name` and `placement-partition-number` filters. In this example, the placement group is named `HDFS-Group-A` and the partition number is `7`.

```
aws ec2 describe-instances --filters "Name = placement-group-name, Values = HDFS-Group-A" "Name = placement-partition-number, Values = 7"
```

The response lists all the instances that are in the specified partition within the specified placement group. The following is example output showing only the instance ID, instance type, and placement information for the returned instances.

```
"Instances": [
    {
        "InstanceId": "i-0a1bc23d4567e8f90",
        "InstanceType": "r4.large",
    },
    {
        "Placement": {
            "AvailabilityZone": "us-east-1c",
            "GroupName": "HDFS-Group-A",
            "PartitionNumber": 7,
            "Tenancy": "default"
        }
    },
    {
        "InstanceId": "i-0a9b876cd5d4ef321",
        "InstanceType": "r4.large",
    },
    {
        "Placement": {
            "AvailabilityZone": "us-east-1c",
            "GroupName": "HDFS-Group-A",
            "PartitionNumber": 7,
            "Tenancy": "default"
        }
    }
],
```

Changing the placement group for an instance

You can change the placement group for an instance in any of the following ways:

- Move an existing instance to a placement group
- Move an instance from one placement group to another
- Remove an instance from a placement group

Before you move or remove the instance, the instance must be in the `stopped` state. You can move or remove an instance using the AWS CLI or an AWS SDK.

AWS CLI

To move an instance to a placement group using the AWS CLI

1. Stop the instance using the [stop-instances](#) command.
2. Use the [modify-instance-placement](#) command and specify the name of the placement group to which to move the instance.

```
aws ec2 modify-instance-placement --instance-id i-0123a456700123456 --group-name MySpreadGroup
```

3. Start the instance using the [start-instances](#) command.

PowerShell

To move an instance to a placement group using the AWS Tools for Windows PowerShell

1. Stop the instance using the [Stop-EC2Instance](#) command.
2. Use the [Edit-EC2InstancePlacement](#) command and specify the name of the placement group to which to move the instance.

3. Start the instance using the [Start-EC2Instance](#) command.

AWS CLI

To remove an instance from a placement group using the AWS CLI

1. Stop the instance using the [stop-instances](#) command.
2. Use the [modify-instance-placement](#) command and specify an empty string for the placement group name.

```
aws ec2 modify-instance-placement --instance-id i-0123a456700123456 --group-name ""
```

3. Start the instance using the [start-instances](#) command.

PowerShell

To remove an instance from a placement group using the AWS Tools for Windows PowerShell

1. Stop the instance using the [Stop-EC2Instance](#) command.
2. Use the [Edit-EC2InstancePlacement](#) command and specify an empty string for the placement group name.
3. Start the instance using the [Start-EC2Instance](#) command.

Deleting a placement group

If you need to replace a placement group or no longer need one, you can delete it. You can delete a placement group using one of the following methods.

Requirement

Before you can delete a placement group, it must contain no instances. You can [terminate \(p. 608\)](#) all instances that you launched into the placement group, [move \(p. 879\)](#) them to another placement group, or [remove \(p. 880\)](#) them from the placement group.

New console

To delete a placement group using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Placement Groups**.
3. Select the placement group and choose **Actions, Delete**.
4. When prompted for confirmation, enter **Delete** and then choose **Delete**.

Old console

To delete a placement group using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Placement Groups**.
3. Select the placement group and choose **Actions, Delete Placement Group**.
4. When prompted for confirmation, choose **Delete**.

AWS CLI

To delete a placement group using the AWS CLI

Use the [delete-placement-group](#) command and specify the placement group name to delete the placement group. In this example, the placement group name is `my-cluster`.

```
aws ec2 delete-placement-group --group-name my-cluster
```

PowerShell

To delete a placement group using the AWS Tools for Windows PowerShell

Use the [Remove-EC2PlacementGroup](#) command to delete the placement group.

Network maximum transmission unit (MTU) for your EC2 instance

The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The larger the MTU of a connection, the more data that can be passed in a single packet. Ethernet packets consist of the frame, or the actual data you are sending, and the network overhead information that surrounds it.

Ethernet frames can come in different formats, and the most common format is the standard Ethernet v2 frame format. It supports 1500 MTU, which is the largest Ethernet packet size supported over most of the internet. The maximum supported MTU for an instance depends on its instance type. All Amazon EC2 instance types support 1500 MTU, and many current instance sizes support 9001 MTU, or jumbo frames.

If your instance runs in a Wavelength Zone, the maximum MTU value is 1300.

To see Network MTU information for Windows instances, switch to this page in the *Amazon EC2 User Guide for Windows Instances* guide: [Network maximum transmission unit \(MTU\) for your EC2 instance](#).

Contents

- [Jumbo frames \(9001 MTU\) \(p. 881\)](#)
- [Path MTU Discovery \(p. 882\)](#)
- [Check the path MTU between two hosts \(p. 882\)](#)
- [Check and set the MTU on your Linux instance \(p. 883\)](#)
- [Troubleshooting \(p. 883\)](#)

Jumbo frames (9001 MTU)

Jumbo frames allow more than 1500 bytes of data by increasing the payload size per packet, and thus increasing the percentage of the packet that is not packet overhead. Fewer packets are needed to send the same amount of usable data. However, outside of a given AWS Region (EC2-Classic), a single VPC, or a VPC peering connection, you will experience a maximum path of 1500 MTU. VPN connections and traffic sent over an internet gateway are limited to 1500 MTU. If packets are over 1500 bytes, they are fragmented, or they are dropped if the `Don't Fragment` flag is set in the IP header.

Jumbo frames should be used with caution for internet-bound traffic or any traffic that leaves a VPC. Packets are fragmented by intermediate systems, which slows down this traffic. To use jumbo frames inside a VPC and not slow traffic that's bound for outside the VPC, you can configure the MTU size by route, or use multiple elastic network interfaces with different MTU sizes and different routes.

For instances that are collocated inside a cluster placement group, jumbo frames help to achieve the maximum network throughput possible, and they are recommended in this case. For more information, see [Placement groups \(p. 869\)](#).

You can use jumbo frames for traffic between your VPCs and your on-premises networks over AWS Direct Connect. For more information, and for how to verify Jumbo Frame capability, see [Setting Network MTU](#) in the [AWS Direct Connect User Guide](#).

All [current generation instances \(p. 206\)](#) support jumbo frames. The following previous generation instances support jumbo frames: C3, G2, I2, M3, and R3.

For more information about supported MTU sizes for transit gateways, see [MTU](#) in [Amazon VPC Transit Gateways](#).

Path MTU Discovery

Path MTU Discovery (PMTUD) is used to determine the maximum transmission unit (MTU) of a network path. Path MTU is the maximum packet size between the originating host and the receiving host. If a host sends a packet that's larger than the MTU of the receiving host or that's larger than the MTU of a device along the path, the receiving host or device returns the following ICMP message: Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (Type 3, Code 4). This instructs the original host to adjust the MTU until the packet can be transmitted.

By default, security groups do not allow any inbound ICMP traffic. However, security groups are stateful, therefore ICMP responses to outbound requests are allowed to flow in, regardless of security group rules. Therefore, you do not need to explicitly add an inbound ICMP rule to ensure that your instance can receive the ICMP message response. For more information about configuring ICMP rules in a network ACL, see [Path MTU Discovery](#) in the [Amazon VPC User Guide](#).

Important

Path MTU Discovery does not guarantee that jumbo frames will not be dropped by some routers. An internet gateway in your VPC will forward packets up to 1500 bytes only. 1500 MTU packets are recommended for internet traffic.

Check the path MTU between two hosts

You can check the path MTU between two hosts using the **tracepath** command, which is part of the **iputils** package that is available by default on many Linux distributions, including Amazon Linux.

To check path MTU using tracepath

Use the following command to check the path MTU between your EC2 instance and another host. You can use a DNS name or an IP address as the destination. If the destination is another EC2 instance, verify that the security group allows inbound UDP traffic. This example checks the path MTU between an EC2 instance and `amazon.com`.

```
[ec2-user ~]$ tracepath amazon.com
1?: [LOCALHOST]          pmtu 9001
1: ip-172-31-16-1.us-west-1.compute.internal (172.31.16.1)    0.187ms pmtu 1500
1: no reply
2: no reply
3: no reply
4: 100.64.16.241 (100.64.16.241)                                0.574ms
5: 72.21.222.221 (72.21.222.221)                                84.447ms asymm 21
6: 205.251.229.97 (205.251.229.97)                            79.970ms asymm 19
7: 72.21.222.194 (72.21.222.194)                                96.546ms asymm 16
8: 72.21.222.239 (72.21.222.239)                            79.244ms asymm 15
9: 205.251.225.73 (205.251.225.73)                            91.867ms asymm 16
...
31: no reply
```

```
Too many hops: pmtu 1500
Resume: pmtu 1500
```

In this example, the path MTU is 1500.

Check and set the MTU on your Linux instance

Some instances are configured to use jumbo frames, and others are configured to use standard frame sizes. You may want to use jumbo frames for network traffic within your VPC or you may want to use standard frames for internet traffic. Whatever your use case, we recommend verifying that your instance will behave the way you expect it to. You can use the procedures in this section to check your network interface's MTU setting and modify it if needed.

To check the MTU setting on a Linux instance

You can check the current MTU value using the following `ip` command. Note that in the example output, `mtu 9001` indicates that this instance uses jumbo frames.

```
[ec2-user ~]$ ip link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP mode DEFAULT
    group default qlen 1000
        link/ether 02:90:c0:b7:9e:d1 brd ff:ff:ff:ff:ff:ff
```

To set the MTU value on a Linux instance

1. You can set the MTU value using the `ip` command. The following command sets the desired MTU value to 1500, but you could use 9001 instead.

```
[ec2-user ~]$ sudo ip link set dev eth0 mtu 1500
```

2. (Optional) To persist your network MTU setting after a reboot, modify the following configuration files, based on your operating system type.

- For Amazon Linux 2, add the following line to the `/etc/sysconfig/network-scripts/ifcfg-eth0` file:

```
MTU=1500
```

Add the following line to the `/etc/dhcp/dhclient.conf` file:

```
request subnet-mask, broadcast-address, time-offset, routers, domain-name, domain-search, domain-name-servers, host-name, nis-domain, nis-servers, ntp-servers;
```

- For Amazon Linux, add the following lines to your `/etc/dhcp/dhclient-eth0.conf` file.

```
interface "eth0" {
    supersede interface-mtu 1500;
}
```

- For other Linux distributions, consult their specific documentation.

3. (Optional) Reboot your instance and verify that the MTU setting is correct.

Troubleshooting

If you experience connectivity issues between your EC2 instance and an Amazon Redshift cluster when using jumbo frames, see [Queries Appear to Hang](#) in the *Amazon Redshift Cluster Management Guide*

Virtual private clouds

Amazon Virtual Private Cloud (Amazon VPC) enables you to define a virtual network in your own logically isolated area within the AWS cloud, known as a *virtual private cloud (VPC)*. You can launch your Amazon EC2 resources, such as instances, into the subnets of your VPC. Your VPC closely resembles a traditional network that you might operate in your own data center, with the benefits of using scalable infrastructure from AWS. You can configure your VPC; you can select its IP address range, create subnets, and configure route tables, network gateways, and security settings. You can connect instances in your VPC to the internet or to your own data center.

When you create your AWS account, we create a *default VPC* for you in each Region. A default VPC is a VPC that is already configured and ready for you to use. You can launch instances into your default VPC immediately. Alternatively, you can create your own *nondefault VPC* and configure it as you need.

If you created your AWS account before 2013-12-04, you might have support for the EC2-Classic platform in some regions. If you created your AWS account after 2013-12-04, it does not support EC2-Classic, so you must launch your resources in a VPC. For more information, see [EC2-Classic \(p. 884\)](#).

Amazon VPC documentation

For more information about Amazon VPC, see the following documentation.

Guide	Description
Amazon VPC User Guide	Describes key concepts and provides instructions for using the features of Amazon VPC.
Amazon VPC Peering Guide	Describes VPC peering connections and provides instructions for using them.
Amazon VPC Transit Gateways	Describes transit gateways and provides instructions for configuring and using them.
AWS Site-to-Site VPN User Guide	Describes Site-to-Site VPN connections and provides instructions for configuring and using them.

EC2-Classic

With EC2-Classic, your instances run in a single, flat network that you share with other customers. With Amazon VPC, your instances run in a virtual private cloud (VPC) that's logically isolated to your AWS account.

The EC2-Classic platform was introduced in the original release of Amazon EC2. If you created your AWS account after 2013-12-04, it does not support EC2-Classic, so you must launch your Amazon EC2 instances in a VPC.

If your account does not support EC2-Classic, we create a default VPC for you. By default, when you launch an instance, we launch it into your default VPC. Alternatively, you can create a nondefault VPC and specify it when you launch an instance.

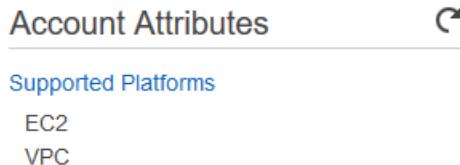
Detecting supported platforms

The Amazon EC2 console indicates which platforms you can launch instances into for the selected region, and whether you have a default VPC in that Region.

Verify that the Region you'll use is selected in the navigation bar. On the Amazon EC2 console dashboard, look for **Supported Platforms** under **Account Attributes**.

Accounts that support EC2-Classic

The dashboard displays the following under **Account Attributes** to indicate that the account supports both the EC2-Classic platform and VPCs in this Region, but the Region does not have a default VPC.

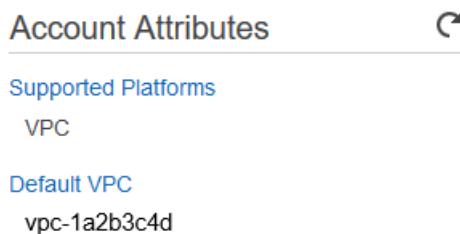


The output of the `describe-account-attributes` command includes both the `EC2` and `VPC` values for the `supported-platforms` attribute.

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
{
    "AccountAttributes": [
        {
            "AttributeName": "supported-platforms",
            "AttributeValues": [
                {
                    "AttributeValue": "EC2"
                },
                {
                    "AttributeValue": "VPC"
                }
            ]
        }
    ]
}
```

Accounts that require a VPC

The dashboard displays the following under **Account Attributes** to indicate that the account requires a VPC to launch instances in this Region, does not support the EC2-Classic platform in this Region, and the Region has a default VPC with the identifier `vpc-1a2b3c4d`.



The output of the `describe-account-attributes` command for the specified Region includes only the `VPC` value for the `supported-platforms` attribute.

```
aws ec2 describe-account-attributes --attribute-names supported-platforms --region us-east-2
{
    "AccountAttributes": [
        {
            "AttributeValues": [

```

```
        {
            "AttributeValue": "VPC"
        }
    ]
}
} ]
```

Instance types available in EC2-Classic

Most of the newer instance types require a VPC. The following are the only instance types supported in EC2-Classic:

- General purpose: M1, M3, and T1
- Compute optimized: C1, C3, and CC2
- Memory optimized: CR1, M2, and R3
- Storage optimized: D2, HS1, and I2
- Accelerated computing: G2

If your account supports EC2-Classic but you have not created a nondefault VPC, you can do one of the following to launch instances that require a VPC:

- Create a nondefault VPC and launch your VPC-only instance into it by specifying a subnet ID or a network interface ID in the request. Note that you must create a nondefault VPC if you do not have a default VPC and you are using the AWS CLI, Amazon EC2 API, or AWS SDK to launch a VPC-only instance.
- Launch your VPC-only instance using the Amazon EC2 console. The Amazon EC2 console creates a nondefault VPC in your account and launches the instance into the subnet in the first Availability Zone. The console creates the VPC with the following attributes:
 - One subnet in each Availability Zone, with the public IPv4 addressing attribute set to `true` so that instances receive a public IPv4 address. For more information, see [IP Addressing in Your VPC](#) in the *Amazon VPC User Guide*.
 - An Internet gateway, and a main route table that routes traffic in the VPC to the Internet gateway. This enables the instances you launch in the VPC to communicate over the Internet. For more information, see [Internet Gateways](#) in the *Amazon VPC User Guide*.
 - A default security group for the VPC and a default network ACL that is associated with each subnet. For more information, see [Security Groups for Your VPC](#) in the *Amazon VPC User Guide*.

If you have other resources in EC2-Classic, you can take steps to migrate them to a VPC. For more information, see [Migrating from EC2-Classic to a VPC \(p. 904\)](#).

Differences between instances in EC2-Classic and a VPC

The following table summarizes the differences between instances launched in EC2-Classic, instances launched in a default VPC, and instances launched in a nondefault VPC.

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Public IPv4 address (from	Your instance receives a public IPv4 address from	Your instance launched in a default subnet receives	Your instance doesn't receive a public IPv4

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Amazon's public IP address pool)	the EC2-Classic public IPv4 address pool.	a public IPv4 address by default, unless you specify otherwise during launch, or you modify the subnet's public IPv4 address attribute.	address by default, unless you specify otherwise during launch, or you modify the subnet's public IPv4 address attribute.
Private IPv4 address	Your instance receives a private IPv4 address from the EC2-Classic range each time it's started.	Your instance receives a static private IPv4 address from the address range of your default VPC.	Your instance receives a static private IPv4 address from the address range of your VPC.
Multiple private IPv4 addresses	We select a single private IP address for your instance; multiple IP addresses are not supported.	You can assign multiple private IPv4 addresses to your instance.	You can assign multiple private IPv4 addresses to your instance.
Elastic IP address (IPv4)	An Elastic IP is disassociated from your instance when you stop it.	An Elastic IP remains associated with your instance when you stop it.	An Elastic IP remains associated with your instance when you stop it.
Associating an Elastic IP address	You associate an Elastic IP address with an instance.	An Elastic IP address is a property of a network interface. You associate an Elastic IP address with an instance by updating the network interface attached to the instance.	An Elastic IP address is a property of a network interface. You associate an Elastic IP address with an instance by updating the network interface attached to the instance.
Reassociating an Elastic IP address	If the Elastic IP address is already associated with another instance, the address is automatically associated with the new instance.	If the Elastic IP address is already associated with another instance, the address is automatically associated with the new instance.	If the Elastic IP address is already associated with another instance, it succeeds only if you allowed reassociation.
Tagging Elastic IP addresses	You cannot apply tags to an Elastic IP address.	You can apply tags to an Elastic IP address.	You can apply tags to an Elastic IP address.
DNS hostnames	DNS hostnames are enabled by default.	DNS hostnames are enabled by default.	DNS hostnames are disabled by default.
Security group	A security group can reference security groups that belong to other AWS accounts.	A security group can reference security groups for your VPC, or for a peer VPC in a VPC peering connection.	A security group can reference security groups for your VPC only.

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Security group association	You can't change the security groups of your running instance. You can either modify the rules of the assigned security groups, or replace the instance with a new one (create an AMI from the instance, launch a new instance from this AMI with the security groups that you need, disassociate any Elastic IP address from the original instance and associate it with the new instance, and then terminate the original instance).	You can assign up to 5 security groups to an instance. You can assign security groups to your instance when you launch it and while it's running.	You can assign up to 5 security groups to an instance. You can assign security groups to your instance when you launch it and while it's running.
Security group rules	You can add rules for inbound traffic only.	You can add rules for inbound and outbound traffic.	You can add rules for inbound and outbound traffic.
Tenancy	Your instance runs on shared hardware.	You can run your instance on shared hardware or single-tenant hardware.	You can run your instance on shared hardware or single-tenant hardware.
Accessing the Internet	Your instance can access the Internet. Your instance automatically receives a public IP address, and can access the Internet directly through the AWS network edge.	By default, your instance can access the Internet. Your instance receives a public IP address by default. An Internet gateway is attached to your default VPC, and your default subnet has a route to the Internet gateway.	By default, your instance cannot access the Internet. Your instance doesn't receive a public IP address by default. Your VPC may have an Internet gateway, depending on how it was created.
IPv6 addressing	IPv6 addressing is not supported. You cannot assign IPv6 addresses to your instances.	You can optionally associate an IPv6 CIDR block with your VPC, and assign IPv6 addresses to instances in your VPC.	You can optionally associate an IPv6 CIDR block with your VPC, and assign IPv6 addresses to instances in your VPC.

Security groups for EC2-Classic

If you're using EC2-Classic, you must use security groups created specifically for EC2-Classic. When you launch an instance in EC2-Classic, you must specify a security group in the same Region as the instance. You can't specify a security group that you created for a VPC when you launch an instance in EC2-Classic.

After you launch an instance in EC2-Classic, you can't change its security groups. However, you can add rules to or remove rules from a security group, and those changes are automatically applied to all instances that are associated with the security group after a short period.

Your AWS account automatically has a default security group per Region for EC2-Classic. If you try to delete the default security group, you'll get the following error: Client.InvalidGroup.Reserved: The security group 'default' is reserved.

You can create custom security groups. The security group name must be unique within your account for the Region. To create a security group for use in EC2-Classic, choose **No VPC** for the VPC.

You can add inbound rules to your default and custom security groups. You can't change the outbound rules for an EC2-Classic security group. When you create a security group rule, you can use a different security group for EC2-Classic in the same Region as the source or destination. To specify a security group for another AWS account, add the AWS account ID as a prefix; for example, 111122223333/sg-edcd9784.

In EC2-Classic, you can have up to 500 security groups in each Region for each account. You can add up to 100 rules to a security group. You can have up to 800 security group rules per instance. This is calculated as the multiple of rules per security group and security groups per instance. If you reference other security groups in your security group rules, we recommend that you use security group names that are 22 characters or less in length.

IP addressing and DNS

Amazon provides a DNS server that resolves Amazon-provided IPv4 DNS hostnames to IPv4 addresses. In EC2-Classic, the Amazon DNS server is located at 172.16.0.23.

If you create a custom firewall configuration in EC2-Classic, you must create a rule in your firewall that allows inbound traffic from port 53 (DNS)—with a destination port from the ephemeral range—from the address of the Amazon DNS server; otherwise, internal DNS resolution from your instances fails. If your firewall doesn't automatically allow DNS query responses, then you need to allow traffic from the IP address of the Amazon DNS server. To get the IP address of the Amazon DNS server, use the following command from within your instance:

```
grep nameserver /etc/resolv.conf
```

Elastic IP addresses

If your account supports EC2-Classic, there's one pool of Elastic IP addresses for use with the EC2-Classic platform and another for use with your VPCs. You can't associate an Elastic IP address that you allocated for use with a VPC with an instance in EC2-Classic, and vice-versa. However, you can migrate an Elastic IP address you've allocated for use in the EC2-Classic platform for use with a VPC. You cannot migrate an Elastic IP address to another Region.

To allocate an Elastic IP address for use in EC2-Classic using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate new address**.
4. Select **Classic**, and then choose **Allocate**. Close the confirmation screen.

Migrating an Elastic IP Address from EC2-Classic

If your account supports EC2-Classic, you can migrate Elastic IP addresses that you've allocated for use with EC2-Classic platform to be used with a VPC, within the same Region. This can assist you to migrate your resources from EC2-Classic to a VPC; for example, you can launch new web servers in your VPC, and

then use the same Elastic IP addresses that you used for your web servers in EC2-Classic for your new VPC web servers.

After you've migrated an Elastic IP address to a VPC, you cannot use it with EC2-Classic. However, if required, you can restore it to EC2-Classic. You cannot migrate an Elastic IP address that was originally allocated for use with a VPC to EC2-Classic.

To migrate an Elastic IP address, it must not be associated with an instance. For more information about disassociating an Elastic IP address from an instance, see [Disassociating an Elastic IP address \(p. 786\)](#).

You can migrate as many EC2-Classic Elastic IP addresses as you can have in your account. However, when you migrate an Elastic IP address, it counts against your Elastic IP address limit for VPCs. You cannot migrate an Elastic IP address if it will result in your exceeding your limit. Similarly, when you restore an Elastic IP address to EC2-Classic, it counts against your Elastic IP address limit for EC2-Classic. For more information, see [Elastic IP address limit \(p. 788\)](#).

You cannot migrate an Elastic IP address that has been allocated to your account for less than 24 hours.

You can migrate an Elastic IP address from EC2-Classic using the Amazon EC2 console or the Amazon VPC console. This option is only available if your account supports EC2-Classic.

To move an Elastic IP address using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address, and choose **Actions, Move to VPC scope**.
4. In the confirmation dialog box, choose **Move Elastic IP**.

You can restore an Elastic IP address to EC2-Classic using the Amazon EC2 console or the Amazon VPC console.

To restore an Elastic IP address to EC2-Classic using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address, choose **Actions, Restore to EC2 scope**.
4. In the confirmation dialog box, choose **Restore**.

After you've performed the command to move or restore your Elastic IP address, the process of migrating the Elastic IP address can take a few minutes. Use the [describe-moving-addresses](#) command to check whether your Elastic IP address is still moving, or has completed moving.

After you've moved your Elastic IP address, you can view its allocation ID on the **Elastic IPs** page in the **Allocation ID** field.

If the Elastic IP address is in a moving state for longer than 5 minutes, contact [Premium Support](#).

To move an Elastic IP address using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [move-address-to-vpc](#) (AWS CLI)
- [Move-EC2AddressToVpc](#) (AWS Tools for Windows PowerShell)

To restore an Elastic IP address to EC2-Classic using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [restore-address-to-classic \(AWS CLI\)](#)
- [Restore-EC2AddressToClassic \(AWS Tools for Windows PowerShell\)](#)

To describe the status of your moving addresses using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-moving-addresses \(AWS CLI\)](#)
- [Get-EC2Address \(AWS Tools for Windows PowerShell\)](#)

Sharing and accessing resources between EC2-Classic and a VPC

Some resources and features in your AWS account can be shared or accessed between EC2-Classic and a VPC, for example, through ClassicLink. For more information, see [ClassicLink \(p. 892\)](#).

If your account supports EC2-Classic, you might have set up resources for use in EC2-Classic. If you want to migrate from EC2-Classic to a VPC, you must recreate those resources in your VPC. For more information about migrating from EC2-Classic to a VPC, see [Migrating from EC2-Classic to a VPC \(p. 904\)](#).

The following resources can be shared or accessed between EC2-Classic and a VPC.

Resource	Notes
AMI	
Bundle task	
EBS volume	
Elastic IP address (IPv4)	You can migrate an Elastic IP address from EC2-Classic to a VPC. You can't migrate an Elastic IP address that was originally allocated for use in a VPC to EC2-Classic. For more information, see Migrating an Elastic IP Address from EC2-Classic (p. 889) .
Instance	An EC2-Classic instance can communicate with instances in a VPC using public IPv4 addresses, or you can use ClassicLink to enable communication over private IPv4 addresses. You can't migrate an instance from EC2-Classic to a VPC. However, you can migrate your application from an instance in EC2-Classic to an instance in a VPC. For more information, see Migrating from EC2-Classic to a VPC (p. 904) .

Resource	Notes
Key pair	
Load balancer	If you're using ClassicLink, you can register a linked EC2-Classic instance with a load balancer in a VPC, provided that the VPC has a subnet in the same Availability Zone as the instance. You can't migrate a load balancer from EC2-Classic to a VPC. You can't register an instance in a VPC with a load balancer in EC2-Classic.
Placement group	
Reserved Instance	You can change the network platform for your Reserved Instances from EC2-Classic to a VPC. For more information, see Modifying Reserved Instances (p. 335) .
Security group	A linked EC2-Classic instance can use a VPC security groups through ClassicLink to control traffic to and from the VPC. VPC instances can't use EC2-Classic security groups. You can't migrate a security group from EC2-Classic to a VPC. You can copy rules from a security group for EC2-Classic to a security group for a VPC. For more information, see Creating a security group (p. 1004) .
Snapshot	

The following resources can't be shared or moved between EC2-Classic and a VPC:

- Spot Instances

ClassicLink

ClassicLink allows you to link EC2-Classic instances to a VPC in your account, within the same Region. If you associate the VPC security groups with a EC2-Classic instance, this enables communication between your EC2-Classic instance and instances in your VPC using private IPv4 addresses. ClassicLink removes the need to make use of public IPv4 addresses or Elastic IP addresses to enable communication between instances in these platforms.

ClassicLink is available to all users with accounts that support the EC2-Classic platform, and can be used with any EC2-Classic instance. For more information about migrating your resources to a VPC, see [Migrating from EC2-Classic to a VPC \(p. 904\)](#).

There is no additional charge for using ClassicLink. Standard charges for data transfer and instance usage apply.

Contents

- [ClassicLink basics \(p. 893\)](#)
- [ClassicLink limitations \(p. 895\)](#)
- [Working with ClassicLink \(p. 896\)](#)

- [Example IAM policies for ClassicLink \(p. 899\)](#)
- [Example: ClassicLink security group configuration for a three-tier web application \(p. 901\)](#)

ClassicLink basics

There are two steps to linking an EC2-Classic instance to a VPC using ClassicLink. First, you must enable the VPC for ClassicLink. By default, all VPCs in your account are not enabled for ClassicLink, to maintain their isolation. After you've enabled the VPC for ClassicLink, you can then link any running EC2-Classic instance in the same Region in your account to that VPC. Linking your instance includes selecting security groups from the VPC to associate with your EC2-Classic instance. After you've linked the instance, it can communicate with instances in your VPC using their private IP addresses, provided the VPC security groups allow it. Your EC2-Classic instance does not lose its private IP address when linked to the VPC.

Linking your instance to a VPC is sometimes referred to as *attaching* your instance.

A linked EC2-Classic instance can communicate with instances in a VPC, but it does not form part of the VPC. If you list your instances and filter by VPC, for example, through the `DescribeInstances` API request, or by using the **Instances** screen in the Amazon EC2 console, the results do not return any EC2-Classic instances that are linked to the VPC. For more information about viewing your linked EC2-Classic instances, see [Viewing your ClassicLink-enabled VPCs and linked instances \(p. 898\)](#).

By default, if you use a public DNS hostname to address an instance in a VPC from a linked EC2-Classic instance, the hostname resolves to the instance's public IP address. The same occurs if you use a public DNS hostname to address a linked EC2-Classic instance from an instance in the VPC. If you want the public DNS hostname to resolve to the private IP address, you can enable ClassicLink DNS support for the VPC. For more information, see [Enabling ClassicLink DNS support \(p. 898\)](#).

If you no longer require a ClassicLink connection between your instance and the VPC, you can unlink the EC2-Classic instance from the VPC. This disassociates the VPC security groups from the EC2-Classic instance. A linked EC2-Classic instance is automatically unlinked from a VPC when it's stopped. After you've unlinked all linked EC2-Classic instances from the VPC, you can disable ClassicLink for the VPC.

Using other AWS services in your VPC with ClassicLink

Linked EC2-Classic instances can access the following AWS services in the VPC: Amazon Redshift, Amazon ElastiCache, Elastic Load Balancing, and Amazon RDS. However, instances in the VPC cannot access the AWS services provisioned by the EC2-Classic platform using ClassicLink.

If you use Elastic Load Balancing, you can register your linked EC2-Classic instances with the load balancer. You must create your load balancer in the ClassicLink-enabled VPC and enable the Availability Zone in which the instance runs. If you terminate the linked EC2-Classic instance, the load balancer deregisters the instance.

If you use Amazon EC2 Auto Scaling, you can create an Amazon EC2 Auto Scaling group with instances that are automatically linked to a specified ClassicLink-enabled VPC at launch. For more information, see [Linking EC2-Classic Instances to a VPC](#) in the *Amazon EC2 Auto Scaling User Guide*.

If you use Amazon RDS instances or Amazon Redshift clusters in your VPC, and they are publicly accessible (accessible from the Internet), the endpoint you use to address those resources from a linked EC2-Classic instance by default resolves to a public IP address. If those resources are not publicly accessible, the endpoint resolves to a private IP address. To address a publicly accessible RDS instance or Redshift cluster over private IP using ClassicLink, you must use their private IP address or private DNS hostname, or you must enable ClassicLink DNS support for the VPC.

If you use a private DNS hostname or a private IP address to address an RDS instance, the linked EC2-Classic instance cannot use the failover support available for Multi-AZ deployments.

You can use the Amazon EC2 console to find the private IP addresses of your Amazon Redshift, Amazon ElastiCache, or Amazon RDS resources.

To locate the private IP addresses of AWS resources in your VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Check the descriptions of the network interfaces in the **Description** column. A network interface that's used by Amazon Redshift, Amazon ElastiCache, or Amazon RDS will have the name of the service in the description. For example, a network interface that's attached to an Amazon RDS instance will have the following description: `RDSNetworkInterface`.
4. Select the required network interface.
5. In the details pane, get the private IP address from the **Primary private IPv4 IP** field.

Controlling the use of ClassicLink

By default, IAM users do not have permission to work with ClassicLink. You can create an IAM policy that grants users permissions to enable or disable a VPC for ClassicLink, link or unlink an instance to a ClassicLink-enabled VPC, and to view ClassicLink-enabled VPCs and linked EC2-Classic instances. For more information about IAM policies for Amazon EC2, see [IAM policies for Amazon EC2 \(p. 921\)](#).

For more information about policies for working with ClassicLink, see the following example: [Example IAM policies for ClassicLink \(p. 899\)](#).

Security groups in ClassicLink

Linking your EC2-Classic instance to a VPC does not affect your EC2-Classic security groups. They continue to control all traffic to and from the instance. This excludes traffic to and from instances in the VPC, which is controlled by the VPC security groups that you associated with the EC2-Classic instance. EC2-Classic instances that are linked to the same VPC cannot communicate with each other through the VPC; regardless of whether they are associated with the same VPC security group. Communication between EC2-Classic instances is controlled by the EC2-Classic security groups associated with those instances. For an example of a security group configuration, see [Example: ClassicLink security group configuration for a three-tier web application \(p. 901\)](#).

After you've linked your instance to a VPC, you cannot change which VPC security groups are associated with the instance. To associate different security groups with your instance, you must first unlink the instance, and then link it to the VPC again, choosing the required security groups.

Routing for ClassicLink

When you enable a VPC for ClassicLink, a static route is added to all of the VPC route tables with a destination of `10.0.0.0/8` and a target of `local`. This allows communication between instances in the VPC and any EC2-Classic instances that are then linked to the VPC. If you add a custom route table to a ClassicLink-enabled VPC, a static route is automatically added with a destination of `10.0.0.0/8` and a target of `local`. When you disable ClassicLink for a VPC, this route is automatically deleted in all of the VPC route tables.

VPCs that are in the `10.0.0.0/16` and `10.1.0.0/16` IP address ranges can be enabled for ClassicLink only if they do not have any existing static routes in route tables in the `10.0.0.0/8` IP address range, excluding the local routes that were automatically added when the VPC was created. Similarly, if you've enabled a VPC for ClassicLink, you may not be able to add any more specific routes to your route tables within the `10.0.0.0/8` IP address range.

Important

If your VPC CIDR block is a publicly routable IP address range, consider the security implications before you link an EC2-Classic instance to your VPC. For example, if your linked EC2-Classic

instance receives an incoming Denial of Service (DoS) request flood attack from a source IP address that falls within the VPC's IP address range, the response traffic is sent into your VPC. We strongly recommend that you create your VPC using a private IP address range as specified in [RFC 1918](#).

For more information about route tables and routing in your VPC, see [Route Tables in the Amazon VPC User Guide](#).

Enabling a VPC peering connection for ClassicLink

If you have a VPC peering connection between two VPCs, and there are one or more EC2-Classic instances that are linked to one or both of the VPCs via ClassicLink, you can extend the VPC peering connection to enable communication between the EC2-Classic instances and the instances in the VPC on the other side of the VPC peering connection. This enables the EC2-Classic instances and the instances in the VPC to communicate using private IP addresses. To do this, you can enable a local VPC to communicate with a linked EC2-Classic instance in a peer VPC, or you can enable a local linked EC2-Classic instance to communicate with instances in a peer VPC.

If you enable a local VPC to communicate with a linked EC2-Classic instance in a peer VPC, a static route is automatically added to your route tables with a destination of 10.0.0.0/8 and a target of local.

For more information and examples, see [Configurations With ClassicLink in the Amazon VPC Peering Guide](#).

ClassicLink limitations

To use the ClassicLink feature, you need to be aware of the following limitations:

- You can link an EC2-Classic instance to only one VPC at a time.
- If you stop your linked EC2-Classic instance, it's automatically unlinked from the VPC and the VPC security groups are no longer associated with the instance. You can link your instance to the VPC again after you've restarted it.
- You cannot link an EC2-Classic instance to a VPC that's in a different Region or a different AWS account.
- You cannot use ClassicLink to link a VPC instance to a different VPC, or to a EC2-Classic resource. To establish a private connection between VPCs, you can use a VPC peering connection. For more information, see the [Amazon VPC Peering Guide](#).
- You cannot associate a VPC Elastic IP address with a linked EC2-Classic instance.
- You cannot enable EC2-Classic instances for IPv6 communication. You can associate an IPv6 CIDR block with your VPC and assign IPv6 address to resources in your VPC, however, communication between a ClassicLinked instance and resources in the VPC is over IPv4 only.
- VPCs with routes that conflict with the EC2-Classic private IP address range of 10/8 cannot be enabled for ClassicLink. This does not include VPCs with 10.0.0.0/16 and 10.1.0.0/16 IP address ranges that already have local routes in their route tables. For more information, see [Routing for ClassicLink \(p. 894\)](#).
- VPCs configured for dedicated hardware tenancy cannot be enabled for ClassicLink. Contact AWS support to request that your dedicated tenancy VPC be allowed to be enabled for ClassicLink.

Important

EC2-Classic instances are run on shared hardware. If you've set the tenancy of your VPC to dedicated because of regulatory or security requirements, then linking an EC2-Classic instance to your VPC might not conform to those requirements, as this allows a shared tenancy resource to address your isolated resources directly using private IP addresses. If you need to enable your dedicated VPC for ClassicLink, provide a detailed reason in your request to AWS support.

- If you link your EC2-Classic instance to a VPC in the 172.16.0.0/16 range, and you have a DNS server running on the 172.16.0.23/32 IP address within the VPC, then your linked EC2-Classic instance can't access the VPC DNS server. To work around this issue, run your DNS server on a different IP address within the VPC.
- ClassicLink doesn't support transitive relationships out of the VPC. Your linked EC2-Classic instance doesn't have access to any VPN connection, VPC gateway endpoint, NAT gateway, or Internet gateway associated with the VPC. Similarly, resources on the other side of a VPN connection or an Internet gateway don't have access to a linked EC2-Classic instance.

Working with ClassicLink

You can use the Amazon EC2 and Amazon VPC consoles to work with the ClassicLink feature. You can enable or disable a VPC for ClassicLink, and link and unlink EC2-Classic instances to a VPC.

Note

The ClassicLink features are only visible in the consoles for accounts and Regions that support EC2-Classic.

Tasks

- [Enabling a VPC for ClassicLink \(p. 896\)](#)
- [Creating a VPC with ClassicLink enabled \(p. 896\)](#)
- [Linking an instance to a VPC \(p. 897\)](#)
- [Linking an instance to a VPC at launch \(p. 897\)](#)
- [Viewing your ClassicLink-enabled VPCs and linked instances \(p. 898\)](#)
- [Enabling ClassicLink DNS support \(p. 898\)](#)
- [Disabling ClassicLink DNS support \(p. 898\)](#)
- [Unlinking an instance from a VPC \(p. 898\)](#)
- [Disabling ClassicLink for a VPC \(p. 899\)](#)

Enabling a VPC for ClassicLink

To link an EC2-Classic instance to a VPC, you must first enable the VPC for ClassicLink. You cannot enable a VPC for ClassicLink if the VPC has routing that conflicts with the EC2-Classic private IP address range. For more information, see [Routing for ClassicLink \(p. 894\)](#).

To enable a VPC for ClassicLink

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select the VPC.
4. Choose **Actions, Enable ClassicLink**.
5. When prompted for confirmation, choose **Enable ClassicLink**.
6. (Optional) If you want the public DNS hostname to resolve to the private IP address, enable ClassicLink DNS support for the VPC before you link any instances. For more information, see [Enabling ClassicLink DNS support \(p. 898\)](#).

Creating a VPC with ClassicLink enabled

You can create a new VPC and immediately enable it for ClassicLink by using the VPC wizard in the Amazon VPC console.

To create a VPC with ClassicLink enabled

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. From the Amazon VPC dashboard, choose **Launch VPC Wizard**.
3. Select one of the VPC configuration options and choose **Select**.
4. On the next page of the wizard, choose **Yes** for **Enable ClassicLink**. Complete the rest of the steps in the wizard to create your VPC. For more information about using the VPC wizard, see [Scenarios for Amazon VPC](#) in the *Amazon VPC User Guide*.
5. (Optional) If you want the public DNS hostname to resolve to the private IP address, enable ClassicLink DNS support for the VPC before you link any instances. For more information, see [Enabling ClassicLink DNS support \(p. 898\)](#).

Linking an instance to a VPC

After you've enabled a VPC for ClassicLink, you can link an EC2-Classic instance to it. The instance must be in the `running` state.

If you want the public DNS hostname to resolve to the private IP address, enable ClassicLink DNS support for the VPC before you link the instance. For more information, see [Enabling ClassicLink DNS support \(p. 898\)](#).

To link an instance to a VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select one or more running EC2-Classic instances.
4. Choose **Actions, ClassicLink, Link to VPC**.
5. Choose the VPC. The console displays only VPCs that are enabled for ClassicLink.
6. Select one or more security groups to associate with your instances. The console displays security groups only for VPCs enabled for ClassicLink.
7. Choose **Link**.

Linking an instance to a VPC at launch

You can use the launch wizard in the Amazon EC2 console to launch an EC2-Classic instance and immediately link it to a ClassicLink-enabled VPC.

To link an instance to a VPC at launch

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the Amazon EC2 dashboard, choose **Launch Instance**.
3. Select an AMI, and then choose an instance type that is supported on EC2-Classic. For more information, see [Instance types available in EC2-Classic \(p. 886\)](#).
4. On the **Configure Instance Details** page, do the following:
 - a. For **Network**, choose **Launch into EC2-Classic**. If this option is disabled, then the instance type is not supported on EC2-Classic.
 - b. Expand **Link to VPC (ClassicLink)** and choose a VPC from **Link to VPC**. The console displays only VPCs with ClassicLink enabled.
5. Complete the rest of the steps in the wizard to launch your instance. For more information, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#).

Viewing your ClassicLink-enabled VPCs and linked instances

You can view all of your ClassicLink-enabled VPCs in the Amazon VPC console, and your linked EC2-Classic instances in the Amazon EC2 console.

To view your ClassicLink-enabled VPCs

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select the VPC.
4. If the value of **ClassicLink** is **Enabled**, then the VPC is enabled for ClassicLink.

Enabling ClassicLink DNS support

You can enable ClassicLink DNS support for your VPC so that DNS hostnames that are addressed between linked EC2-Classic instances and instances in the VPC resolve to private IP addresses and not public IP addresses. For this feature to work, your VPC must be enabled for DNS hostnames and DNS resolution.

Note

If you enable ClassicLink DNS support for your VPC, your linked EC2-Classic instance can access any private hosted zone associated with the VPC. For more information, see [Working with Private Hosted Zones](#) in the *Amazon Route 53 Developer Guide*.

To enable ClassicLink DNS support

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select the VPC.
4. Choose **Actions, Edit ClassicLink DNS Support**.
5. For **ClassicLink DNS support**, select **Enable**.
6. Choose **Save changes**.

Disabling ClassicLink DNS support

You can disable ClassicLink DNS support for your VPC so that DNS hostnames that are addressed between linked EC2-Classic instances and instances in the VPC resolve to public IP addresses and not private IP addresses.

To disable ClassicLink DNS support

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select the VPC.
4. Choose **Actions, Edit ClassicLink DNS Support**.
5. For **ClassicLink DNS Support**, clear **Enable**.
6. Choose **Save changes**.

Unlinking an instance from a VPC

If you no longer require a ClassicLink connection between your EC2-Classic instance and your VPC, you can unlink the instance from the VPC. Unlinking the instance disassociates the VPC security groups from the instance.

A stopped instance is automatically unlinked from a VPC.

To unlink an instance from a VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select one or more of your instances.
4. Choose **Actions, ClassicLink, Unlink from VPC**.
5. When prompted for confirmation, choose **Unlink**.

Disabling ClassicLink for a VPC

If you no longer require a connection between EC2-Classic instances and your VPC, you can disable ClassicLink on the VPC. You must first unlink all linked EC2-Classic instances that are linked to the VPC.

To disable ClassicLink for a VPC

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**.
3. Select your VPC.
4. Choose **Actions, Disable ClassicLink**.
5. When prompted for confirmation, choose **Disable ClassicLink**.

Example IAM policies for ClassicLink

You can enable a VPC for ClassicLink and then link an EC2-Classic instance to the VPC. You can also view your ClassicLink-enabled VPCs, and all of your EC2-Classic instances that are linked to a VPC. You can create policies with resource-level permission for the `ec2:EnableVpcClassicLink`, `ec2:DisableVpcClassicLink`, `ec2:AttachClassicLinkVpc`, and `ec2:DetachClassicLinkVpc` actions to control how users are able to use those actions. Resource-level permissions are not supported for `ec2:Describe*` actions.

Examples

- [Full permissions to work with ClassicLink \(p. 899\)](#)
- [Enable and disable a VPC for ClassicLink \(p. 900\)](#)
- [Link instances \(p. 900\)](#)
- [Unlink instances \(p. 901\)](#)

Full permissions to work with ClassicLink

The following policy grants users permissions to view ClassicLink-enabled VPCs and linked EC2-Classic instances, to enable and disable a VPC for ClassicLink, and to link and unlink instances from a ClassicLink-enabled VPC.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeClassicLinkInstances", "ec2:DescribeVpcClassicLink",  
                "ec2:EnableVpcClassicLink", "ec2:DisableVpcClassicLink",  
                "ec2:AttachClassicLinkVpc", "ec2:DetachClassicLinkVpc"  
            ]  
        }  
    ]  
}
```

```
        ],
        "Resource": "*"
    }
}
```

Enable and disable a VPC for ClassicLink

The following policy allows user to enable and disable VPCs for ClassicLink that have the specific tag 'purpose=classiclink'. Users cannot enable or disable any other VPCs for ClassicLink.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:*VpcClassicLink",
            "Resource": "arn:aws:ec2:region:account:vpc/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/purpose": "classiclink"
                }
            }
        }
    ]
}
```

Link instances

The following policy grants users permissions to link instances to a VPC only if the instance is an m3.large instance type. The second statement allows users to use the VPC and security group resources, which are required to link an instance to a VPC.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:AttachClassicLinkVpc",
            "Resource": "arn:aws:ec2:region:account:instance/*",
            "Condition": {
                "StringEquals": {
                    "ec2:InstanceType": "m3.large"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:AttachClassicLinkVpc",
            "Resource": [
                "arn:aws:ec2:region:account:vpc/*",
                "arn:aws:ec2:region:account:security-group/*"
            ]
        }
    ]
}
```

The following policy grants users permissions to link instances to a specific VPC (vpc-1a2b3c4d) only, and to associate only specific security groups from the VPC to the instance (sg-1122aabb and sg-aabb2233). Users cannot link an instance to any other VPC, and they cannot specify any other of the VPC security groups to associate with the instance in the request.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:AttachClassicLinkVpc",  
            "Resource": [  
                "arn:aws:ec2:region:account:vpc/vpc-1a2b3c4d",  
                "arn:aws:ec2:region:account:instance/*",  
                "arn:aws:ec2:region:account:security-group/sg-1122aabb",  
                "arn:aws:ec2:region:account:security-group/sg-aabb2233"  
            ]  
        }  
    ]  
}
```

Unlink instances

The following grants users permission to unlink any linked EC2-Classic instance from a VPC, but only if the instance has the tag "unlink=true". The second statement grants users permissions to use the VPC resource, which is required to unlink an instance from a VPC.

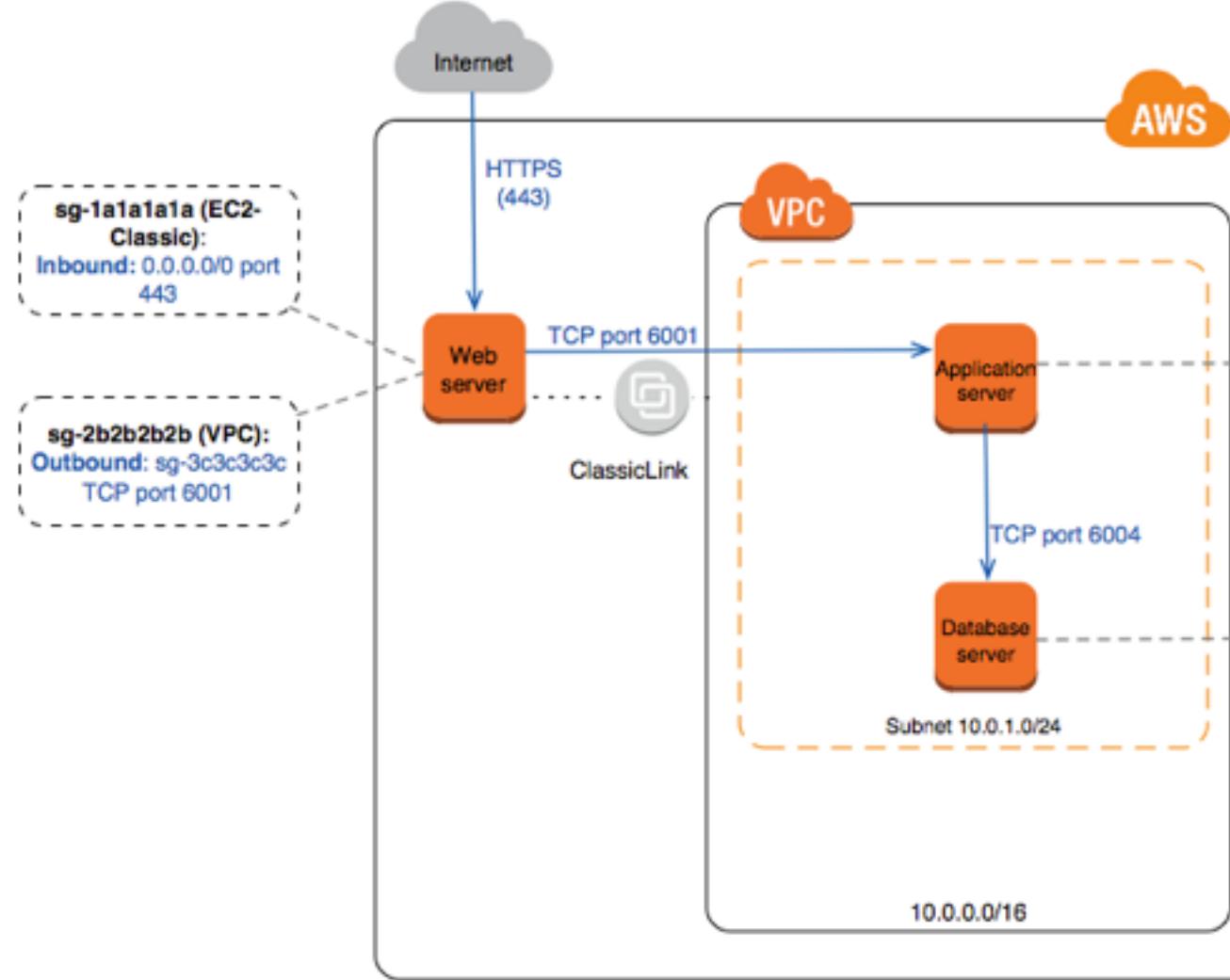
```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:DetachClassicLinkVpc",  
            "Resource": [  
                "arn:aws:ec2:region:account:instance/*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/unlink": "true"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:DetachClassicLinkVpc",  
            "Resource": [  
                "arn:aws:ec2:region:account:vpc/*"  
            ]  
        }  
    ]  
}
```

Example: ClassicLink security group configuration for a three-tier web application

In this example, you have an application with three instances: a public-facing web server, an application server, and a database server. Your web server accepts HTTPS traffic from the Internet, and then communicates with your application server over TCP port 6001. Your application server then communicates with your database server over TCP port 6004. You're in the process of migrating your entire application to a VPC in your account. You've already migrated your application server and your database server to your VPC. Your web server is still in EC2-Classic and linked to your VPC via ClassicLink.

You want a security group configuration that allows traffic to flow only between these instances. You have four security groups: two for your web server (sg-1a1a1a1a and sg-2b2b2b2b), one for your application server (sg-3c3c3c3c), and one for your database server (sg-4d4d4d4d).

The following diagram displays the architecture of your instances, and their security group configuration.



Security groups for your web server (**sg-1a1a1a1a** and **sg-2b2b2b2b**)

You have one security group in EC2-Classic, and the other in your VPC. You associated the VPC security group with your web server instance when you linked the instance to your VPC via ClassicLink. The VPC security group enables you to control the outbound traffic from your web server to your application server.

The following are the security group rules for the EC2-Classic security group (**sg-1a1a1a1a**).

Inbound			
Source	Type	Port Range	Comments
0.0.0.0/0	HTTPS	443	Allows Internet traffic to reach your web server.

The following are the security group rules for the VPC security group (**sg-2b2b2b2b**).

Outbound			
Destination	Type	Port Range	Comments
sg-3c3c3c3c	TCP	6001	Allows outbound traffic from your web server to your application server in your VPC (or to any other instance associated with sg-3c3c3c3c).

Security group for your application server (sg-3c3c3c3c)

The following are the security group rules for the VPC security group that's associated with your application server.

Inbound			
Source	Type	Port Range	Comments
sg-2b2b2b2b	TCP	6001	Allows the specified type of traffic from your web server (or any other instance associated with sg-2b2b2b2b) to reach your application server.

Outbound			
Destination	Type	Port Range	Comments
sg-4d4d4d4d	TCP	6004	Allows outbound traffic from the application server to the database server (or to any other instance associated with sg-4d4d4d4d).

Security group for your database server (sg-4d4d4d4d)

The following are the security group rules for the VPC security group that's associated with your database server.

Inbound			
Source	Type	Port Range	Comments
sg-3c3c3c3c	TCP	6004	Allows the specified type of traffic from your application server (or any other instance associated with sg-3c3c3c3c) to reach your database server.

Migrating from EC2-Classic to a VPC

If you created your AWS account before December 4, 2013, you might have support for EC2-Classic in some AWS Regions. Some Amazon EC2 resources and features, such as enhanced networking and newer instance types, require a virtual private cloud (VPC). Some resources can be shared between EC2-Classic and a VPC, while some can't. For more information, see [Sharing and accessing resources between EC2-Classic and a VPC \(p. 891\)](#). We recommend that you migrate to a VPC to take advantage of VPC-only features.

To migrate from EC2-Classic to a VPC, you must migrate or recreate your EC2-Classic resources in a VPC. You can migrate and recreate your resources in full, or you can perform an incremental migration over time using ClassicLink.

Contents

- [Options for getting a default VPC \(p. 904\)](#)
- [Migrate your resources to a VPC \(p. 905\)](#)
- [Use ClassicLink for an incremental migration \(p. 909\)](#)
- [Example: Migrate a simple web application \(p. 910\)](#)

Options for getting a default VPC

A *default VPC* is a VPC that is configured and ready for you to use, and is only available in Regions that are VPC-only. For Regions that support EC2-Classic, you can create a nondefault VPC to set up your resources. However, you might want to use a default VPC if you prefer not to set up a VPC yourself, or if you do not have specific requirements for your VPC configuration. For more information about default VPCs, see [Default VPC and Default Subnets](#) in the *Amazon VPC User Guide*.

The following are options for using a default VPC when you have an AWS account that supports EC2-Classic.

Options

- [Switch to a VPC-only Region \(p. 904\)](#)
- [Create a new AWS account \(p. 904\)](#)
- [Convert your existing AWS account to VPC-only \(p. 904\)](#)

Switch to a VPC-only Region

Use this option if you want to use your existing account to set up your resources in a default VPC and you do not need to use a specific Region. To find a Region that has a default VPC, see [Detecting supported platforms \(p. 884\)](#).

Create a new AWS account

New AWS accounts support VPC only. Use this option if you want an account that has a default VPC in every Region.

Convert your existing AWS account to VPC-only

Use this option if you want a default VPC in every Region in your existing account. Before you can convert your account, you must delete all of your EC2-Classic resources. You can also migrate some resources to a VPC. For more information, see [Migrate your resources to a VPC \(p. 905\)](#).

To convert your EC2-Classic account

1. Delete or migrate (if applicable) the resources that you have created for use in EC2-Classic. These include the following:
 - Amazon EC2 instances
 - EC2-Classic security groups (excluding the default security group, which you cannot delete yourself)
 - EC2-Classic Elastic IP addresses
 - Classic Load Balancers
 - Amazon RDS resources
 - Amazon ElastiCache resources
 - Amazon Redshift resources
 - AWS Elastic Beanstalk resources
 - AWS Data Pipeline resources
 - Amazon EMR resources
 - AWS OpsWorks resources
2. Go to the AWS Support Center at console.aws.amazon.com/support.
3. Choose **Create case**.
4. Choose **Account and billing support**.
5. For **Type**, choose **Account**. For **Category**, choose **Convert EC2 Classic to VPC**.
6. Fill in the other details as required, and choose **Submit**. We will review your request and contact you to guide you through the next steps.

Migrate your resources to a VPC

You can migrate or move some of your resources to a VPC. Some resources can only be migrated from EC2-Classic to a VPC that's in the same Region and in the same AWS account. If the resource cannot be migrated, you must create a new resource for use in your VPC.

Prerequisites

Before you begin, you must have a VPC. If you don't have a default VPC, you can create a nondefault VPC using one of these methods:

- In the Amazon VPC console, use the VPC wizard to create a new VPC. For more information, see [Amazon VPC Console Wizard Configurations](#). Use this option if you want to set up a VPC quickly, using one of the available configuration options.
- In the Amazon VPC console, set up the components of a VPC according to your requirements. For more information, see [VPCs and Subnets](#). Use this option if you have specific requirements for your VPC, such as a particular number of subnets.

Topics

- [Security groups \(p. 906\)](#)
- [Elastic IP addresses \(p. 906\)](#)
- [AMIs and instances \(p. 906\)](#)
- [Amazon RDS DB instances \(p. 909\)](#)

Security groups

If you want your instances in your VPC to have the same security group rules as your EC2-Classic instances, you can use the Amazon EC2 console to copy your existing EC2-Classic security group rules to a new VPC security group.

You can only copy security group rules to a new security group in the same AWS account in the same Region. If you are using a different Region or a different AWS account, you must create a new security group and manually add the rules yourself. For more information, see [Amazon EC2 security groups for Linux instances \(p. 999\)](#).

To copy your security group rules to a new security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group that's associated with your EC2-Classic instance, then choose **Actions**, and select **Copy to new**.

Note

To identify an EC2-Classic security group, check the **VPC ID** column. For each EC2-Classic security group, the value in the column is blank or a – symbol.

4. In the **Create Security Group** dialog box, specify a name and description for your new security group. Select your VPC from the **VPC** list.
5. The **Inbound** tab is populated with the rules from your EC2-Classic security group. You can modify the rules as required. In the **Outbound** tab, a rule that allows all outbound traffic has automatically been created for you. For more information about modifying security group rules, see [Amazon EC2 security groups for Linux instances \(p. 999\)](#).

Note

If you've defined a rule in your EC2-Classic security group that references another security group, you cannot use the same rule in your VPC security group. Modify the rule to reference a security group in the same VPC.

6. Choose **Create**.

Elastic IP addresses

You can migrate an Elastic IP address that is allocated for use in EC2-Classic for use with a VPC. You cannot migrate an Elastic IP address to another Region or AWS account. For more information, see [Migrating an Elastic IP Address from EC2-Classic \(p. 889\)](#).

To identify an Elastic IP address that is allocated for use in EC2-Classic

In the Amazon EC2 console, choose **Elastic IPs** in the navigation pane. In the **Scope** column, the value is **standard**.

Alternatively, use the following `describe-addresses` command.

```
aws ec2 describe-addresses --filters Name=domain,Values=standard
```

AMIs and instances

An AMI is a template for launching your Amazon EC2 instance. You can create your own AMI based on an existing EC2-Classic instance, then use that AMI to launch instances into your VPC.

Contents

- [Identify EC2-Classic instances \(p. 907\)](#)

- [Create an AMI \(p. 907\)](#)
- [\(Optional\) Share or copy your AMI \(p. 908\)](#)
- [\(Optional\) Store your data on Amazon EBS volumes \(p. 908\)](#)
- [Launch an instance into your VPC \(p. 908\)](#)

Identify EC2-Classic instances

If you have instances running in both EC2-Classic and a VPC, you can identify your EC2-Classic instances.

Amazon EC2 console

Choose **Instances** in the navigation pane. In the **VPC ID** column, the value for each EC2-Classic instance is blank or a – symbol. If the **VPC ID** column is not present, choose the gear icon and make the column visible.

AWS CLI

Use the following [describe-instances](#) AWS CLI command. The --query parameter displays only instances where the value for `VpcId` is null.

```
aws ec2 describe-instances --query 'Reservations[*].Instances[?VpcId==`null`]'
```

Create an AMI

After you've identified your EC2-Classic instance, you can create an AMI from it.

To create a Windows AMI

For more information, see [Creating a custom Windows AMI](#).

To create a Linux AMI

The method that you use to create your Linux AMI depends on the root device type of your instance, and the operating system platform on which your instance runs. To find out the root device type of your instance, go to the **Instances** page, select your instance, and look at the information in the **Root device type** field in the **Description** tab. If the value is `ebs`, then your instance is EBS-backed. If the value is `instance-store`, then your instance is instance store-backed. You can also use the [describe-instances](#) AWS CLI command to find out the root device type.

The following table provides options for you to create your Linux AMI based on the root device type of your instance, and the software platform.

Important

Some instance types support both PV and HVM virtualization, while others support only one or the other. If you plan to use your AMI to launch a different instance type than your current instance type, verify that the instance type supports the type of virtualization that your AMI offers. If your AMI supports PV virtualization, and you want to use an instance type that supports HVM virtualization, you might have to reinstall your software on a base HVM AMI. For more information about PV and HVM virtualization, see [Linux AMI virtualization types](#).

Instance root device type	Action
EBS	Create an EBS-backed AMI from your instance. For more information, see Creating an Amazon EBS-backed Linux AMI .

Instance root device type	Action
Instance store	Create an instance store-backed AMI from your instance using the AMI tools. For more information, see Creating an instance store-backed Linux AMI .
Instance store	Convert your instance store-backed instance to an EBS-backed instance. For more information, see Converting your instance store-backed AMI to an Amazon EBS-backed AMI .

(Optional) Share or copy your AMI

To use your AMI to launch an instance in a new AWS account, you must first share the AMI with your new account. For more information, see [Sharing an AMI with specific AWS accounts \(p. 113\)](#).

To use your AMI to launch an instance in a VPC in a different Region, you must first copy the AMI to that Region. For more information, see [Copying an AMI \(p. 163\)](#).

(Optional) Store your data on Amazon EBS volumes

You can create an Amazon EBS volume and use it to back up and store the data on your instance—like you would use a physical hard drive. Amazon EBS volumes can be attached and detached from any instance in the same Availability Zone. You can detach a volume from your instance in EC2-Classic, and attach it to a new instance that you launch into your VPC in the same Availability Zone.

For more information about Amazon EBS volumes, see the following topics:

- [Amazon EBS volumes \(p. 1021\)](#)
- [Creating an Amazon EBS volume \(p. 1040\)](#)
- [Attaching an Amazon EBS volume to an instance \(p. 1042\)](#)

To back up the data on your Amazon EBS volume, you can take periodic snapshots of your volume. For more information, see [Creating Amazon EBS snapshots \(p. 1063\)](#). If you need to, you can create an Amazon EBS volume from your snapshot. For more information, see [Creating a volume from a snapshot \(p. 1041\)](#).

Launch an instance into your VPC

After you've created an AMI, you can use the Amazon EC2 launch wizard to launch an instance into your VPC. The instance will have the same data and configurations as your existing EC2-Classic instance.

Note

You can use this opportunity to [upgrade to a current generation instance type](#). However, verify that the instance type supports the type of virtualization that your AMI offers (PV or HVM). For more information about PV and HVM virtualization, see [Linux AMI virtualization types \(p. 102\)](#).

To launch an instance into your VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch instance**.
3. On the **Choose an Amazon Machine Image** page, select the **My AMIs** category, and select the AMI you created. Alternatively, if you shared an AMI from another account, in the **Ownership** filter list, choose **Shared with me**. Select the AMI that you shared from your EC2-Classic account.
4. On the **Choose an Instance Type** page, select the type of instance, and choose **Next: Configure Instance Details**.

5. On the **Configure Instance Details** page, select your VPC from the **Network** list. Select the required subnet from the **Subnet** list. Configure any other details that you require, then go through the next pages of the wizard until you reach the **Configure Security Group** page.
6. Select **Select an existing group**, and select the security group that you created for your VPC. Choose **Review and Launch**.
7. Review your instance details, then choose **Launch** to specify a key pair and launch your instance.

For more information about the parameters that you can configure in each step of the wizard, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#).

Amazon RDS DB instances

You can move your EC2-Classic DB instance to a VPC in the same Region, in the same account. For more information, see [Updating the VPC for a DB Instance](#) in the *Amazon RDS User Guide*.

Use ClassicLink for an incremental migration

The ClassicLink feature makes it easier to manage an incremental migration to a VPC. ClassicLink enables you to link an EC2-Classic instance to a VPC in your account in the same Region, allowing your new VPC resources to communicate with the EC2-Classic instance using private IPv4 addresses. You can then migrate functionality one component at a time until your application is running fully in your VPC.

Use this option if you cannot afford downtime during the migration, for example, if you have a multi-tier application with processes that cannot be interrupted.

For more information about ClassicLink, see [ClassicLink \(p. 892\)](#).

Tasks

- [Step 1: Prepare your migration sequence \(p. 909\)](#)
- [Step 2: Enable your VPC for ClassicLink \(p. 909\)](#)
- [Step 3: Link your EC2-Classic instances to your VPC \(p. 910\)](#)
- [Step 4: Complete the VPC migration \(p. 910\)](#)

Step 1: Prepare your migration sequence

To use ClassicLink effectively, you must first identify the components of your application that must be migrated to the VPC, and then confirm the order in which to migrate that functionality.

For example, you have an application that relies on a presentation web server, a backend database server, and authentication logic for transactions. You may decide to start the migration process with the authentication logic, then the database server, and finally, the web server.

Then, you can start migrating or recreating your resources. For more information, see [Migrate your resources to a VPC \(p. 905\)](#).

Step 2: Enable your VPC for ClassicLink

After you've configured your new VPC instances and made the functionality of your application available in the VPC, you can use ClassicLink to enable private IP communication between your new VPC instances and your EC2-Classic instances. First, you must enable your VPC for ClassicLink.

To enable a VPC for ClassicLink

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.

2. In the navigation pane, choose **Your VPCs**.
3. Select a VPC.
4. Choose **Actions, Enable ClassicLink**.
5. When prompted for confirmation, choose **Enable ClassicLink**.

Step 3: Link your EC2-Classic instances to your VPC

After you've enabled ClassicLink in your VPC, you can link your EC2-Classic instances to the VPC. The instance must be in the `running` state.

To link an instance to a VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select one or more running EC2-Classic instances.
4. Choose **Actions, ClassicLink, Link to VPC**.
5. Choose a VPC. The console displays only VPCs that are enabled for ClassicLink.
6. Select one or more security groups to associate with your instances. The console displays security groups only for VPCs enabled for ClassicLink.
7. Choose **Link**.

Step 4: Complete the VPC migration

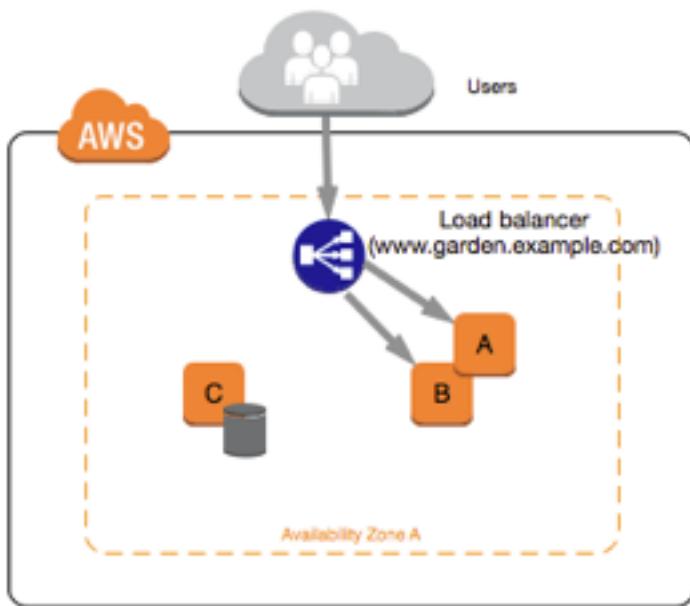
Depending on the size of your application and the functionality that must be migrated, repeat the preceding steps until you've moved all of the components of your application from EC2-Classic into your VPC.

After you've enabled internal communication between the EC2-Classic and VPC instances, you must update your application to point to your migrated service in your VPC, instead of your service in the EC2-Classic platform. The exact steps for this depend on your application's design. Generally, this includes updating your destination IP addresses to point to the IP addresses of your VPC instances instead of your EC2-Classic instances.

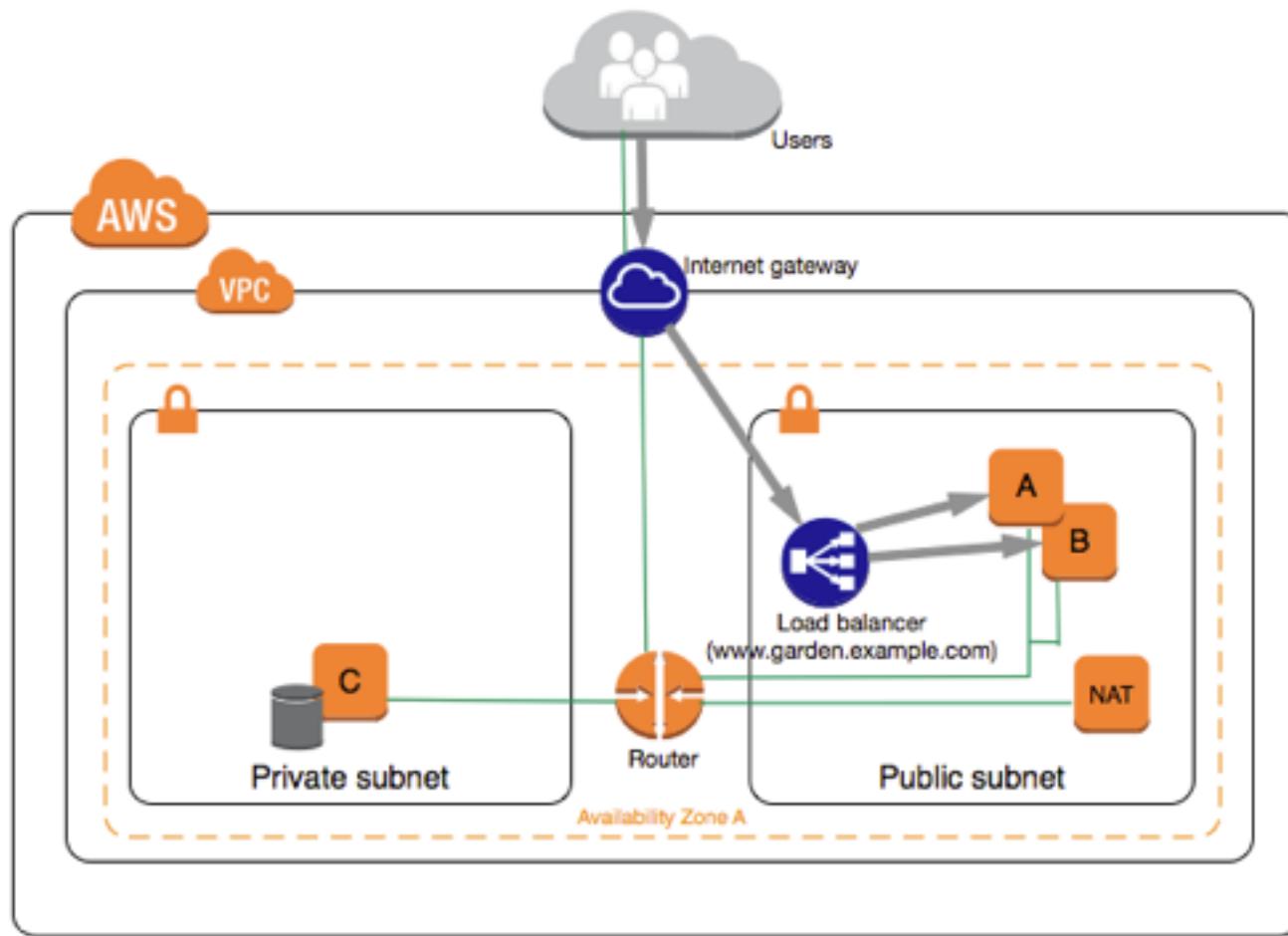
After you've completed this step and you've tested that the application is functioning from your VPC, you can terminate your EC2-Classic instances, and disable ClassicLink for your VPC. You can also clean up any EC2-Classic resources that you no longer need to avoid incurring charges for them. For example, you can release Elastic IP addresses and delete the volumes that were associated with your EC2-Classic instances.

Example: Migrate a simple web application

In this example, you use AWS to host your gardening website. To manage your website, you have three running instances in EC2-Classic. Instances A and B host your public-facing web application, and you use Elastic Load Balancing to load balance the traffic between these instances. You've assigned Elastic IP addresses to instances A and B so that you have static IP addresses for configuration and administration tasks on those instances. Instance C holds your MySQL database for your website. You've registered the domain name `www.garden.example.com`, and you've used Route 53 to create a hosted zone with an alias record set that's associated with the DNS name of your load balancer.



The first part of migrating to a VPC is deciding what kind of VPC architecture suits your needs. In this case, you've decided on the following: one public subnet for your web servers, and one private subnet for your database server. As your website grows, you can add more web servers and database servers to your subnets. By default, instances in the private subnet cannot access the internet; however, you can enable internet access through a Network Address Translation (NAT) device in the public subnet. You might want to set up a NAT device to support periodic updates and patches from the internet for your database server. You'll migrate your Elastic IP addresses to a VPC, and create a load balancer in your public subnet to load balance the traffic between your web servers.



To migrate your web application to a VPC, you can follow these steps:

- **Create a VPC:** In this case, you can use the VPC wizard in the Amazon VPC console to create your VPC and subnets. The second wizard configuration creates a VPC with one private and one public subnet, and launches and configures a NAT device in your public subnet for you. For more information, see [VPC with public and private subnets \(NAT\)](#) in the *Amazon VPC User Guide*.
- **Configure your security groups:** In your EC2-Classic environment, you have one security group for your web servers, and another security group for your database server. You can use the Amazon EC2 console to copy the rules from each security group into new security groups for your VPC. For more information, see [Security groups \(p. 906\)](#).

Tip

Create the security groups that are referenced by other security groups first.

- **Create AMIs and launch new instances:** Create an AMI from one of your web servers, and a second AMI from your database server. Then, launch replacement web servers into your public subnet, and launch your replacement database server into your private subnet. For more information, see [Create an AMI \(p. 907\)](#).
- **Configure your NAT device:** If you are using a NAT instance, you must create a security group for it that allows HTTP and HTTPS traffic from your private subnet. For more information, see [NAT instances](#). If you are using a NAT gateway, traffic from your private subnet is automatically allowed.
- **Configure your database:** When you created an AMI from your database server in EC2-Classic, all of the configuration information that was stored in that instance was copied to the AMI. You might have to connect to your new database server and update the configuration details. For example, if you

configured your database to grant full read, write, and modification permissions to your web servers in EC2-Classic, you need to update the configuration files to grant the same permissions to your new VPC web servers instead.

- **Configure your web servers:** Your web servers will have the same configuration settings as your instances in EC2-Classic. For example, if you configured your web servers to use the database in EC2-Classic, update your web servers' configuration settings to point to your new database instance.

Note

By default, instances launched into a nondefault subnet are not assigned a public IP address, unless you specify otherwise at launch. Your new database server might not have a public IP address. In this case, you can update your web servers' configuration file to use your new database server's private DNS name. Instances in the same VPC can communicate with each other via private IP address.

- **Migrate your Elastic IP addresses:** Disassociate your Elastic IP addresses from your web servers in EC2-Classic, and then migrate them to a VPC. After you've migrated them, you can associate them with your new web servers in your VPC. For more information, see [Migrating an Elastic IP Address from EC2-Classic \(p. 889\)](#).
- **Create a new load balancer:** To continue using Elastic Load Balancing to load balance the traffic to your instances, make sure you understand the various ways to configure your load balancer in VPC. For more information, see the [Elastic Load Balancing User Guide](#).
- **Update your DNS records:** After you've set up your load balancer in your public subnet, verify that your `www.garden.example.com` domain points to your new load balancer. To do this, update your DNS records and your alias record set in Route 53. For more information about using Route 53, see [Getting Started with Route 53](#).
- **Shut down your EC2-Classic resources:** After you've verified that your web application is working from within the VPC architecture, you can shut down your EC2-Classic resources to stop incurring charges for them.

Security in Amazon EC2

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The [shared responsibility model](#) describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the [AWS Compliance Programs](#). To learn about the compliance programs that apply to Amazon EC2, see [AWS Services in Scope by Compliance Program](#).
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Amazon EC2. It shows you how to configure Amazon EC2 to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon EC2 resources.

Contents

- [Infrastructure security in Amazon EC2 \(p. 914\)](#)
- [Amazon EC2 and interface VPC endpoints \(p. 916\)](#)
- [Resilience in Amazon EC2 \(p. 917\)](#)
- [Data protection in Amazon EC2 \(p. 918\)](#)
- [Identity and access management for Amazon EC2 \(p. 919\)](#)
- [Amazon EC2 key pairs and Linux instances \(p. 985\)](#)
- [Amazon EC2 security groups for Linux instances \(p. 999\)](#)
- [Update management in Amazon EC2 \(p. 1017\)](#)
- [Compliance validation for Amazon EC2 \(p. 1017\)](#)

Infrastructure security in Amazon EC2

As a managed service, Amazon EC2 is protected by the AWS global network security procedures that are described in the [Amazon Web Services: Overview of Security Processes](#) whitepaper.

You use AWS published API calls to access Amazon EC2 through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the [AWS Security Token Service](#) (AWS STS) to generate temporary security credentials to sign requests.

Network isolation

A virtual private cloud (VPC) is a virtual network in your own logically isolated area in the AWS Cloud. Use separate VPCs to isolate infrastructure by workload or organizational entity.

A subnet is a range of IP addresses in a VPC. When you launch an instance, you launch it into a subnet in your VPC. Use subnets to isolate the tiers of your application (for example, web, application, and database) within a single VPC. Use private subnets for your instances if they should not be accessed directly from the internet.

To call the Amazon EC2 API from your VPC without sending traffic over the public internet, use AWS PrivateLink.

Isolation on physical hosts

Different EC2 instances on the same physical host are isolated from each other as though they are on separate physical hosts. The hypervisor isolates CPU and memory, and the instances are provided virtualized disks instead of access to the raw disk devices.

When you stop or terminate an instance, the memory allocated to it is scrubbed (set to zero) by the hypervisor before it is allocated to a new instance, and every block of storage is reset. This ensures that your data is not unintentionally exposed to another instance.

Network MAC addresses are dynamically assigned to instances by the AWS network infrastructure. IP addresses are either dynamically assigned to instances by the AWS network infrastructure, or assigned by an EC2 administrator through authenticated API requests. The AWS network allows instances to send traffic only from the MAC and IP addresses assigned to them. Otherwise, the traffic is dropped.

By default, an instance cannot receive traffic that is not specifically addressed to it. If you need to run network address translation (NAT), routing, or firewall services on your instance, you can disable source/destination checking for the network interface.

Controlling network traffic

Consider the following options for controlling network traffic to your EC2 instances:

- Restrict access to your instances using [security groups \(p. 999\)](#). For example, you can allow traffic only from the address ranges for your corporate network.
- Use private subnets for your instances if they should not be accessed directly from the internet. Use a bastion host or NAT gateway for internet access from an instance in a private subnet.
- Use AWS Virtual Private Network or AWS Direct Connect to establish private connections from your remote networks to your VPCs. For more information, see [Network-to-Amazon VPC Connectivity Options](#).
- Use [VPC Flow Logs](#) to monitor the traffic that reaches your instances.
- Use [AWS Security Hub](#) to check for unintended network accessibility from your instances.
- Use [EC2 Instance Connect \(p. 569\)](#) to connect to your instances using Secure Shell (SSH) without the need to share and manage SSH keys.
- Use [AWS Systems Manager Session Manager](#) to access your instances remotely instead of opening inbound SSH ports and managing SSH keys.
- Use [AWS Systems Manager Run Command](#) to automate common administrative tasks instead of opening inbound SSH ports and managing SSH keys.

In addition to restricting network access to each Amazon EC2 instance, Amazon VPC supports implementing additional network security controls like in-line gateways, proxy servers, and various network monitoring options.

For more information, see the [AWS Security Best Practices](#) whitepaper.

Amazon EC2 and interface VPC endpoints

You can improve the security posture of your VPC by configuring Amazon EC2 to use an interface VPC endpoint. Interface endpoints are powered by AWS PrivateLink, a technology that enables you to privately access Amazon EC2 APIs by restricting all network traffic between your VPC and Amazon EC2 to the Amazon network. With interface endpoints, you also don't need an internet gateway, a NAT device, or a virtual private gateway.

You are not required to configure AWS PrivateLink, but it's recommended. For more information about AWS PrivateLink and VPC endpoints, see [Interface VPC Endpoints \(AWS PrivateLink\)](#).

Topics

- [Create an interface VPC endpoint \(p. 916\)](#)
- [Create an interface VPC endpoint policy \(p. 916\)](#)

Create an interface VPC endpoint

Create an endpoint for Amazon EC2 using the following service name:

- **com.amazonaws.*region*.ec2** — Creates an endpoint for the Amazon EC2 API actions.

For more information, see [Creating an Interface Endpoint](#) in the *Amazon VPC User Guide*.

Create an interface VPC endpoint policy

You can attach a policy to your VPC endpoint to control access to the Amazon EC2 API. The policy specifies:

- The principal that can perform actions.
- The actions that can be performed.
- The resource on which the actions can be performed.

Important

When a non-default policy is applied to an interface VPC endpoint for Amazon EC2, certain failed API requests, such as those failing from `RequestLimitExceeded`, might not be logged to AWS CloudTrail or Amazon CloudWatch.

For more information, see [Controlling Access to Services with VPC Endpoints](#) in the *Amazon VPC User Guide*.

The following example shows a VPC endpoint policy that denies permission to create unencrypted volumes or to launch instances with unencrypted volumes. The example policy also grants permission to perform all other Amazon EC2 actions.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": "ec2:*",  
            "Effect": "Allow",  
            "Resource": "*",  
            "Principal": "*"  
        }  
    ]  
}
```

```
},
{
  "Action": [
    "ec2:CreateVolume"
  ],
  "Effect": "Deny",
  "Resource": "*",
  "Principal": "*",
  "Condition": {
    "Bool": {
      "ec2:Encrypted": "false"
    }
  }
},
{
  "Action": [
    "ec2:RunInstances"
  ],
  "Effect": "Deny",
  "Resource": "*",
  "Principal": "*",
  "Condition": {
    "Bool": {
      "ec2:Encrypted": "false"
    }
  }
}
]
```

Resilience in Amazon EC2

The AWS global infrastructure is built around AWS Regions and Availability Zones. Regions provide multiple physically separated and isolated Availability Zones, which are connected through low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

If you need to replicate your data or applications over greater geographic distances, use AWS Local Zones. An AWS Local Zone is an extension of an AWS Region in geographic proximity to your users. Local Zones have their own connections to the internet and support AWS Direct Connect. Like all AWS Regions, AWS Local Zones are completely isolated from other AWS Zones.

If you need to replicate your data or applications in an AWS Local Zone, AWS recommends that you use one of the following zones as the failover zone:

- Another Local Zone
- An Availability Zone in the Region that is not the parent zone. You can use the [describe-availability-zones](#) command to view the parent zone.

For more information about AWS Regions and Availability Zones, see [AWS Global Infrastructure](#).

In addition to the AWS global infrastructure, Amazon EC2 offers the following features to support your data resiliency:

- Copying AMIs across Regions
- Copying EBS snapshots across Regions
- Automating EBS snapshots using Amazon Data Lifecycle Manager

- Maintaining the health and availability of your fleet using Amazon EC2 Auto Scaling
- Distributing incoming traffic across multiple instances in a single Availability Zone or multiple Availability Zones using Elastic Load Balancing

Data protection in Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) conforms to the AWS [shared responsibility model](#), which includes regulations and guidelines for data protection. AWS is responsible for protecting the global infrastructure that runs all AWS services. AWS maintains control over data hosted on this infrastructure, including the security configuration controls for handling customer content and personal data. AWS customers and APN Partners, acting either as data controllers or data processors, are responsible for any personal data that they put in the AWS Cloud.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM), so that each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see [Federal Information Processing Standard \(FIPS\) 140-2](#).

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into free-form fields or metadata, such as function names and tags. Any data that you enter into metadata might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credential information in the URL to validate your request to that server.

For more information about data protection, see the [AWS Shared Responsibility Model and GDPR](#) blog post on the [AWS Security Blog](#).

Encryption at rest

Amazon EBS encryption is an encryption solution for your EBS volumes and snapshots. It uses AWS Key Management Service (AWS KMS) customer master keys (CMK). For more information, see [Amazon EBS encryption \(p. 1121\)](#).

The data on NVMe instance store volumes is encrypted using an XTS-AES-256 cipher implemented on a hardware module on the instance. The encryption keys are generated using the hardware module and are unique to each NVMe instance storage device. All encryption keys are destroyed when the instance is stopped or terminated and cannot be recovered. You cannot disable this encryption and you cannot provide your own encryption key.

Encryption in transit

SSH provides a secure communications channel for remote access to your Linux instances. Remote access to your instances using AWS Systems Manager Session Manager and Run Command is encrypted using TLS 1.2, and requests to create a connection are signed using SigV4.

Use an encryption protocol such as Transport Layer Security (TLS) to encrypt sensitive data in transit between clients and your instances.

AWS provides secure and private connectivity between EC2 instances of all types. In addition, some instance types use the offload capabilities of the underlying hardware to automatically encrypt in-transit traffic between instances, using AEAD algorithms with 256-bit encryption. There is no impact on network performance. The following requirements must be met to ensure the additional in-transit traffic encryption:

- The instances use the following instance types: C5a, C5ad, C5n, G4, I3en, M5dn, M5n, P3dn, R5dn, and R5n.
- The instances are in the same Region.
- The instances are in the same VPC or peered VPCs, and the traffic does not pass through a virtual network device, such as a load balancer or a transit gateway.

Identity and access management for Amazon EC2

Your security credentials identify you to services in AWS and grant you unlimited use of your AWS resources, such as your Amazon EC2 resources. You can use features of Amazon EC2 and AWS Identity and Access Management (IAM) to allow other users, services, and applications to use your Amazon EC2 resources without sharing your security credentials. You can use IAM to control how other users use resources in your AWS account, and you can use security groups to control access to your Amazon EC2 instances. You can choose to allow full use or limited use of your Amazon EC2 resources.

Contents

- [Network access to your instance \(p. 919\)](#)
- [Amazon EC2 permission attributes \(p. 919\)](#)
- [IAM and Amazon EC2 \(p. 920\)](#)
- [IAM policies for Amazon EC2 \(p. 921\)](#)
- [IAM roles for Amazon EC2 \(p. 974\)](#)
- [Authorizing inbound traffic for your Linux instances \(p. 983\)](#)

Network access to your instance

A security group acts as a firewall that controls the traffic allowed to reach one or more instances. When you launch an instance, you assign it one or more security groups. You add rules to each security group that control traffic for the instance. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances to which the security group is assigned.

For more information, see [Authorizing inbound traffic for your Linux instances \(p. 983\)](#).

Amazon EC2 permission attributes

Your organization might have multiple AWS accounts. Amazon EC2 enables you to specify additional AWS accounts that can use your Amazon Machine Images (AMIs) and Amazon EBS snapshots. These permissions work at the AWS account level only; you can't restrict permissions for specific users within the specified AWS account. All users in the AWS account that you've specified can use the AMI or snapshot.

Each AMI has a `LaunchPermission` attribute that controls which AWS accounts can access the AMI. For more information, see [Making an AMI public \(p. 112\)](#).

Each Amazon EBS snapshot has a `createVolumePermission` attribute that controls which AWS accounts can use the snapshot. For more information, see [Sharing an Amazon EBS snapshot \(p. 1073\)](#).

IAM and Amazon EC2

IAM enables you to do the following:

- Create users and groups under your AWS account
- Assign unique security credentials to each user under your AWS account
- Control each user's permissions to perform tasks using AWS resources
- Allow the users in another AWS account to share your AWS resources
- Create roles for your AWS account and define the users or services that can assume them
- Use existing identities for your enterprise to grant permissions to perform tasks using AWS resources

By using IAM with Amazon EC2, you can control whether users in your organization can perform a task using specific Amazon EC2 API actions and whether they can use specific AWS resources.

This topic helps you answer the following questions:

- How do I create groups and users in IAM?
- How do I create a policy?
- What IAM policies do I need to carry out tasks in Amazon EC2?
- How do I grant permissions to perform actions in Amazon EC2?
- How do I grant permissions to perform actions on specific resources in Amazon EC2?

Creating an IAM group and users

To create an IAM group

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Groups** and then choose **Create New Group**.
3. For **Group Name**, enter a name for your group, and then choose **Next Step**.
4. On the **Attach Policy** page, select an AWS managed policy and then choose **Next Step**. For example, for Amazon EC2, one of the following AWS managed policies might meet your needs:
 - PowerUserAccess
 - ReadOnlyAccess
 - AmazonEC2FullAccess
 - AmazonEC2ReadOnlyAccess
5. Choose **Create Group**.

Your new group is listed under **Group Name**.

To create an IAM user, add the user to your group, and create a password for the user

1. In the navigation pane, choose **Users**, **Add user**.
2. For **User name**, enter a user name.
3. For **Access type**, select both **Programmatic access** and **AWS Management Console access**.

4. For **Console password**, choose one of the following:
 - **Autogenerated password.** Each user gets a randomly generated password that meets the current password policy in effect (if any). You can view or download the passwords when you get to the **Final** page.
 - **Custom password.** Each user is assigned the password that you enter in the box.
 5. Choose **Next: Permissions**.
 6. On the **Set permissions** page, choose **Add user to group**. Select the check box next to the group that you created earlier and choose **Next: Review**.
 7. Choose **Create user**.
 8. To view the users' access keys (access key IDs and secret access keys), choose **Show** next to each password and secret access key to see. To save the access keys, choose **Download .csv** and then save the file to a safe location.
- Important**
You cannot retrieve the secret access key after you complete this step; if you misplace it you must create a new one.
9. Choose **Close**.
 10. Give each user his or her credentials (access keys and password); this enables them to use services based on the permissions you specified for the IAM group.

Related topics

For more information about IAM, see the following:

- [IAM policies for Amazon EC2 \(p. 921\)](#)
- [IAM roles for Amazon EC2 \(p. 974\)](#)
- [AWS Identity and Access Management \(IAM\)](#)
- [IAM User Guide](#)

IAM policies for Amazon EC2

By default, IAM users don't have permission to create or modify Amazon EC2 resources, or perform tasks using the Amazon EC2 API. (This means that they also can't do so using the Amazon EC2 console or CLI.) To allow IAM users to create or modify resources and perform tasks, you must create IAM policies that grant IAM users permission to use the specific resources and API actions they'll need, and then attach those policies to the IAM users or groups that require those permissions.

When you attach a policy to a user or group of users, it allows or denies the users permission to perform the specified tasks on the specified resources. For more general information about IAM policies, see [Permissions and Policies](#) in the *IAM User Guide*. For more information about managing and creating custom IAM policies, see [Managing IAM Policies](#).

Getting Started

An IAM policy must grant or deny permissions to use one or more Amazon EC2 actions. It must also specify the resources that can be used with the action, which can be all resources, or in some cases, specific resources. The policy can also include conditions that you apply to the resource.

Amazon EC2 partially supports resource-level permissions. This means that for some EC2 API actions, you cannot specify which resource a user is allowed to work with for that action. Instead, you have to allow users to work with all resources for that action.

Task	Topic
Understand the basic structure of a policy	Policy syntax (p. 922)
Define actions in your policy	Actions for Amazon EC2 (p. 923)
Define specific resources in your policy	Amazon Resource Names (ARNs) for Amazon EC2 (p. 924)
Apply conditions to the use of the resources	Condition keys for Amazon EC2 (p. 925)
Work with the available resource-level permissions for Amazon EC2	Actions, Resources, and Condition Keys for Amazon EC2 (IAM User Guide)
Test your policy	Checking that users have the required permissions (p. 926)
Example policies for a CLI or SDK	Example policies for working with the AWS CLI or an AWS SDK (p. 929)
Example policies for the Amazon EC2 console	Example policies for working in the Amazon EC2 console (p. 965)

Policy structure

The following topics explain the structure of an IAM policy.

Contents

- [Policy syntax \(p. 922\)](#)
- [Actions for Amazon EC2 \(p. 923\)](#)
- [Supported resource-level permissions for Amazon EC2 API actions \(p. 923\)](#)
- [Amazon Resource Names \(ARNs\) for Amazon EC2 \(p. 924\)](#)
- [Condition keys for Amazon EC2 \(p. 925\)](#)
- [Checking that users have the required permissions \(p. 926\)](#)

Policy syntax

An IAM policy is a JSON document that consists of one or more statements. Each statement is structured as follows.

```
{
  "Statement": [
    {
      "Effect": "effect",
      "Action": "action",
      "Resource": "arn",
      "Condition": {
        "condition": {
          "key": "value"
        }
      }
    }
  ]
}
```

There are various elements that make up a statement:

- **Effect:** The *effect* can be `Allow` or `Deny`. By default, IAM users don't have permission to use resources and API actions, so all requests are denied. An explicit allow overrides the default. An explicit deny overrides any allows.
- **Action:** The *action* is the specific API action for which you are granting or denying permission. To learn about specifying *action*, see [Actions for Amazon EC2 \(p. 923\)](#).
- **Resource:** The resource that's affected by the action. Some Amazon EC2 API actions allow you to include specific resources in your policy that can be created or modified by the action. You specify a resource using an Amazon Resource Name (ARN) or using the wildcard (*) to indicate that the statement applies to all resources. For more information, see [Supported resource-level permissions for Amazon EC2 API actions \(p. 923\)](#).
- **Condition:** Conditions are optional. They can be used to control when your policy is in effect. For more information about specifying conditions for Amazon EC2, see [Condition keys for Amazon EC2 \(p. 925\)](#).

For more information about example IAM policy statements for Amazon EC2, see [Example policies for working with the AWS CLI or an AWS SDK \(p. 929\)](#).

Actions for Amazon EC2

In an IAM policy statement, you can specify any API action from any service that supports IAM. For Amazon EC2, use the following prefix with the name of the API action: `ec2::`. For example: `ec2:RunInstances` and `ec2:CreateImage`.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": ["ec2:action1", "ec2:action2"]
```

You can also specify multiple actions using wildcards. For example, you can specify all actions whose name begins with the word "Describe" as follows:

```
"Action": "ec2:Describe*"
```

To specify all Amazon EC2 API actions, use the `*` wildcard as follows:

```
"Action": "ec2:*"
```

For a list of Amazon EC2 actions, see [Actions in the Amazon EC2 API Reference](#).

Supported resource-level permissions for Amazon EC2 API actions

Resource-level permissions refers to the ability to specify which resources users are allowed to perform actions on. Amazon EC2 has partial support for resource-level permissions. This means that for certain Amazon EC2 actions, you can control when users are allowed to use those actions based on conditions that have to be fulfilled, or specific resources that users are allowed to use. For example, you can grant users permissions to launch instances, but only of a specific type, and only using a specific AMI.

To specify a resource in an IAM policy statement, use its Amazon Resource Name (ARN). For more information about specifying the ARN value, see [Amazon Resource Names \(ARNs\) for Amazon EC2 \(p. 924\)](#). If an API action does not support individual ARNs, you must use a wildcard (*) to specify that all resources can be affected by the action.

To see tables that identify which Amazon EC2 API actions support resource-level permissions, and the ARNs and condition keys that you can use in a policy, see [Actions, Resources, and Condition Keys for Amazon EC2](#) in the *IAM User Guide*. Condition keys for Amazon EC2 are also further explained in a later section.

Keep in mind that you can apply tag-based resource-level permissions in the IAM policies you use for Amazon EC2 API actions. This gives you better control over which resources a user can create, modify, or use. For more information, see [Granting permission to tag resources during creation \(p. 926\)](#).

Amazon Resource Names (ARNs) for Amazon EC2

Each IAM policy statement applies to the resources that you specify using their ARNs.

An ARN has the following general syntax:

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

service

The service (for example, ec2).

region

The Region for the resource (for example, us-east-1).

account

The AWS account ID, with no hyphens (for example, 123456789012).

resourceType

The type of resource (for example, instance).

resourcePath

A path that identifies the resource. You can use the * wildcard in your paths.

For example, you can indicate a specific instance (i-1234567890abcdef0) in your statement using its ARN as follows.

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
```

You can specify all instances that belong to a specific account by using the * wildcard as follows.

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

You can also specify all Amazon EC2 resources that belong to a specific account by using the * wildcard as follows.

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:*
```

To specify all resources, or if a specific API action does not support ARNs, use the * wildcard in the Resource element as follows.

```
"Resource": "*"
```

Many Amazon EC2 API actions involve multiple resources. For example, `AttachVolume` attaches an Amazon EBS volume to an instance, so an IAM user must have permissions to use the volume and the instance. To specify multiple resources in a single statement, separate their ARNs with commas, as follows.

```
"Resource": ["arn1", "arn2"]
```

For a list of ARNs for Amazon EC2 resources, see [Resource Types Defined by Amazon EC2](#) in the *IAM User Guide*.

Condition keys for Amazon EC2

In a policy statement, you can optionally specify conditions that control when it is in effect. Each condition contains one or more key-value pairs. Condition keys are not case-sensitive. We've defined AWS-wide condition keys, plus additional service-specific condition keys.

For a list of service-specific condition keys for Amazon EC2, see [Condition Keys for Amazon EC2](#) in the *IAM User Guide*. Amazon EC2 also implements the AWS-wide condition keys. For more information, see [Information Available in All Requests](#) in the *IAM User Guide*.

To use a condition key in your IAM policy, use the `Condition` statement. For example, the following policy grants users permission to add and remove inbound and outbound rules for any security group. It uses the `ec2:Vpc` condition key to specify that these actions can only be performed on security groups in a specific VPC.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:AuthorizeSecurityGroupIngress",  
                "ec2:AuthorizeSecurityGroupEgress",  
                "ec2:RevokeSecurityGroupIngress",  
                "ec2:RevokeSecurityGroupEgress"],  
            "Resource": "arn:aws:ec2:region:account:security-group/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:Vpc": "arn:aws:ec2:region:account:vpc/vpc-11223344556677889"  
                }  
            }  
        }  
    ]  
}
```

If you specify multiple conditions, or multiple keys in a single condition, we evaluate them using a logical AND operation. If you specify a single condition with multiple values for one key, we evaluate the condition using a logical OR operation. For permissions to be granted, all conditions must be met.

You can also use placeholders when you specify conditions. For example, you can grant an IAM user permission to use resources with a tag that specifies his or her IAM user name. For more information, see [Policy Variables](#) in the *IAM User Guide*.

Important

Many condition keys are specific to a resource, and some API actions use multiple resources. If you write a policy with a condition key, use the `Resource` element of the statement to specify the resource to which the condition key applies. If not, the policy may prevent users from performing the action at all, because the condition check fails for the resources to which the condition key does not apply. If you do not want to specify a resource, or if you've written the `Action` element of your policy to include multiple API actions, then you must use the `...IfExists` condition type to ensure that the condition key is ignored for resources that do not use it. For more information, see [...IfExists Conditions](#) in the *IAM User Guide*.

All Amazon EC2 actions support the `aws:RequestedRegion` and `ec2:Region` condition keys. For more information, see [Example: Restricting access to a specific Region \(p. 930\)](#).

The `ec2:SourceInstanceARN` key can be used for conditions that specify the ARN of the instance from which a request is made. This condition key is available AWS-wide and is not service-specific. For policy examples, see [Allows an EC2 Instance to Attach or Detach Volumes](#) and [Example: Allowing a specific](#)

instance to view resources in other AWS services (p. 961). The `ec2:SourceInstanceARN` key cannot be used as a variable to populate the ARN for the `Resource` element in a statement.

For example policy statements for Amazon EC2, see [Example policies for working with the AWS CLI or an AWS SDK \(p. 929\)](#).

Checking that users have the required permissions

After you've created an IAM policy, we recommend that you check whether it grants users the permissions to use the particular API actions and resources they need before you put the policy into production.

First, create an IAM user for testing purposes, and then attach the IAM policy that you created to the test user. Then, make a request as the test user.

If the Amazon EC2 action that you are testing creates or modifies a resource, you should make the request using the `DryRun` parameter (or run the AWS CLI command with the `--dry-run` option). In this case, the call completes the authorization check, but does not complete the operation. For example, you can check whether the user can terminate a particular instance without actually terminating it. If the test user has the required permissions, the request returns `DryRunOperation`; otherwise, it returns `UnauthorizedOperation`.

If the policy doesn't grant the user the permissions that you expected, or is overly permissive, you can adjust the policy as needed and retest until you get the desired results.

Important

It can take several minutes for policy changes to propagate before they take effect. Therefore, we recommend that you allow five minutes to pass before you test your policy updates.

If an authorization check fails, the request returns an encoded message with diagnostic information. You can decode the message using the `DecodeAuthorizationMessage` action. For more information, see [DecodeAuthorizationMessage](#) in the *AWS Security Token Service API Reference*, and [decode-authorization-message](#) in the *AWS CLI Command Reference*.

Granting permission to tag resources during creation

Some resource-creating Amazon EC2 API actions enable you to specify tags when you create the resource. For more information, see [Tagging your resources \(p. 1231\)](#).

To enable users to tag resources on creation, they must have permissions to use the action that creates the resource, such as `ec2:RunInstances` or `ec2>CreateVolume`. If tags are specified in the resource-creating action, Amazon performs additional authorization on the `ec2:CreateTags` action to verify if users have permissions to create tags. Therefore, users must also have explicit permissions to use the `ec2:CreateTags` action.

In the IAM policy definition for the `ec2:CreateTags` action, use the `Condition` element with the `ec2:CreateAction` condition key to give tagging permissions to the action that creates the resource.

The following example demonstrates a policy that allows users to launch instances and apply any tags to instances and volumes during launch. Users are not permitted to tag any existing resources (they cannot call the `ec2:CreateTags` action directly).

```
{  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances"  
            ],  
            "Resource": "*"
```

```
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:region:account:/*/*",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction" : "RunInstances"
        }
    }
}
]
```

Similarly, the following policy allows users to create volumes and apply any tags to the volumes during volume creation. Users are not permitted to tag any existing resources (they cannot call the `ec2:CreateTags` action directly).

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2>CreateVolume"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:region:account:/*/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction" : "CreateVolume"
                }
            }
        }
    ]
}
```

The `ec2:CreateTags` action is only evaluated if tags are applied during the resource-creating action. Therefore, a user that has permissions to create a resource (assuming there are no tagging conditions) does not require permissions to use the `ec2:CreateTags` action if no tags are specified in the request. However, if the user attempts to create a resource with tags, the request fails if the user does not have permissions to use the `ec2:CreateTags` action.

The `ec2:CreateTags` action is also evaluated if tags are provided in a launch template. For an example policy, see [Tags in a launch template \(p. 949\)](#).

Controlling access to specific tags

You can use additional conditions in the `Condition` element of your IAM policies to control the tag keys and values that can be applied to resources.

The following condition keys can be used with the examples in the preceding section:

- `aws:RequestTag`: To indicate that a particular tag key or tag key and value must be present in a request. Other tags can also be specified in the request.

- Use with the `StringEquals` condition operator to enforce a specific tag key and value combination, for example, to enforce the tag `cost-center=cc123`:

```
"StringEquals": { "aws:RequestTag/cost-center": "cc123" }
```

- Use with the `StringLike` condition operator to enforce a specific tag key in the request; for example, to enforce the tag key `purpose`:

```
"StringLike": { "aws:RequestTag/purpose": "*" }
```

- `aws:TagKeys`: To enforce the tag keys that are used in the request.
 - Use with the `ForAllValues` modifier to enforce specific tag keys if they are provided in the request (if tags are specified in the request, only specific tag keys are allowed; no other tags are allowed). For example, the tag keys `environment` or `cost-center` are allowed:

```
"ForAllValues:StringEquals": { "aws:TagKeys": [ "environment", "cost-center" ] }
```

- Use with the `ForAnyValue` modifier to enforce the presence of at least one of the specified tag keys in the request. For example, at least one of the tag keys `environment` or `webserver` must be present in the request:

```
"ForAnyValue:StringEquals": { "aws:TagKeys": [ "environment", "webserver" ] }
```

These condition keys can be applied to resource-creating actions that support tagging, as well as the `ec2:CreateTags` and `ec2:DeleteTags` actions. To learn whether an Amazon EC2 API action supports tagging, see [Actions, Resources, and Condition Keys for Amazon EC2](#) in the *IAM User Guide*.

To force users to specify tags when they create a resource, you must use the `aws:RequestTag` condition key or the `aws:TagKeys` condition key with the `ForAnyValue` modifier on the resource-creating action. The `ec2:CreateTags` action is not evaluated if a user does not specify tags for the resource-creating action.

For conditions, the condition key is not case-sensitive and the condition value is case-sensitive. Therefore, to enforce the case-sensitivity of a tag key, use the `aws:TagKeys` condition key, where the tag key is specified as a value in the condition.

For example IAM policies, see [Example policies for working with the AWS CLI or an AWS SDK \(p. 929\)](#). For more information about multi-value conditions, see [Creating a Condition That Tests Multiple Key Values](#) in the *IAM User Guide*.

Controlling access to EC2 resources using resource tags

When you create an IAM policy that grants IAM users permission to use EC2 resources, you can include tag information in the Condition element of the policy to control access based on tags. This gives you better control over which EC2 resources a user can modify, use, or delete.

For example, you can create a policy that allows users to terminate an instance but denies the action if the instance has the tag `environment=production`. To do this, you use the `ec2:ResourceTag` condition key to allow or deny access to the resource based on the tags that are attached to the resource.

```
"StringEquals": { "ec2:ResourceTag/environment": "production" }
```

To learn whether an Amazon EC2 API action supports controlling access using the `ec2:ResourceTag` condition key, see [Actions, Resources, and Condition Keys for Amazon EC2](#) in the *IAM User Guide*. Note that the `Describe` actions do not support resource-level permissions, and therefore you must specify them in a separate statement without conditions.

For example IAM policies, see [Example policies for working with the AWS CLI or an AWS SDK \(p. 929\)](#).

Note

If you allow or deny users access to resources based on tags, you must consider explicitly denying users the ability to add those tags to or remove them from the same resources. Otherwise, it's possible for a user to circumvent your restrictions and gain access to a resource by modifying its tags.

Example policies for working with the AWS CLI or an AWS SDK

The following examples show policy statements that you could use to control the permissions that IAM users have to Amazon EC2. These policies are designed for requests that are made with the AWS CLI or an AWS SDK. For example policies for working in the Amazon EC2 console, see [Example policies for working in the Amazon EC2 console \(p. 965\)](#). For examples of IAM policies specific to Amazon VPC, see [Identity and Access Management for Amazon VPC](#).

Examples

- [Example: Read-only access \(p. 929\)](#)
- [Example: Restricting access to a specific Region \(p. 930\)](#)
- [Working with instances \(p. 930\)](#)
- [Working with volumes \(p. 932\)](#)
- [Working with snapshots \(p. 934\)](#)
- [Launching instances \(RunInstances\) \(p. 941\)](#)
- [Working with Spot Instances \(p. 952\)](#)
- [Example: Working with Reserved Instances \(p. 957\)](#)
- [Example: Tagging resources \(p. 958\)](#)
- [Example: Working with IAM roles \(p. 960\)](#)
- [Example: Working with route tables \(p. 961\)](#)
- [Example: Allowing a specific instance to view resources in other AWS services \(p. 961\)](#)
- [Example: Working with launch templates \(p. 962\)](#)
- [Working with instance metadata \(p. 963\)](#)

Example: Read-only access

The following policy grants users permissions to use all Amazon EC2 API actions whose names begin with `Describe`. The `Resource` element uses a wildcard to indicate that users can specify all resources with these API actions. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see [Actions, Resources, and Condition Keys for Amazon EC2](#) in the *IAM User Guide*.

Users don't have permission to perform any actions on the resources (unless another statement grants them permission to do so) because they're denied permission to use API actions by default.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:Describe*",  
            "Resource": "*"  
        }  
    ]  
}
```

Example: Restricting access to a specific Region

The following policy denies users permission to use all Amazon EC2 API actions unless the Region is Europe (Frankfurt). It uses the global condition key `aws:RequestedRegion`, which is supported by all Amazon EC2 API actions.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Action": "ec2:*",  
            "Resource": "*",  
            "Condition": {  
                "StringNotEquals": {  
                    "aws:RequestedRegion": "eu-central-1"  
                }  
            }  
        }  
    ]  
}
```

Alternatively, you can use the condition key `ec2:Region`, which is specific to Amazon EC2 and is supported by all Amazon EC2 API actions.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Action": "ec2:*",  
            "Resource": "*",  
            "Condition": {  
                "StringNotEquals": {  
                    "ec2:Region": "eu-central-1"  
                }  
            }  
        }  
    ]  
}
```

Working with instances

Examples

- [Example: Describe, launch, stop, start, and terminate all instances \(p. 930\)](#)
- [Example: Describe all instances, and stop, start, and terminate only particular instances \(p. 931\)](#)

Example: Describe, launch, stop, start, and terminate all instances

The following policy grants users permissions to use the API actions specified in the `Action` element. The `Resource` element uses a `*` wildcard to indicate that users can specify all resources with these API actions. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see [Actions, Resources, and Condition Keys for Amazon EC2](#) in the *IAM User Guide*.

The users don't have permission to use any other API actions (unless another statement grants them permission to do so) because users are denied permission to use API actions by default.

```
{
```

```

"Version": "2012-10-17",
"Statement": [
{
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeImages",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeAvailabilityZones",
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "ec2:StopInstances",
        "ec2:StartInstances"
    ],
    "Resource": "*"
}
]
}

```

[Example: Describe all instances, and stop, start, and terminate only particular instances](#)

The following policy allows users to describe all instances, to start and stop only instances i-1234567890abcdef0 and i-0598c7d356eba48d7, and to terminate only instances in the US East (N. Virginia) Region (us-east-1) with the resource tag "purpose=test".

The first statement uses a * wildcard for the Resource element to indicate that users can specify all resources with the action; in this case, they can list all instances. The * wildcard is also necessary in cases where the API action does not support resource-level permissions (in this case, ec2:DescribeInstances). For more information about which ARNs you can use with which Amazon EC2 API actions, see [Actions, Resources, and Condition Keys for Amazon EC2](#) in the *IAM User Guide*.

The second statement uses resource-level permissions for the StopInstances and StartInstances actions. The specific instances are indicated by their ARNs in the Resource element.

The third statement allows users to terminate all instances in the US East (N. Virginia) Region (us-east-1) that belong to the specified AWS account, but only where the instance has the tag "purpose=test". The Condition element qualifies when the policy statement is in effect.

```

{
    "Version": "2012-10-17",
    "Statement": [
{
    "Effect": "Allow",
    "Action": "ec2:DescribeInstances",
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:StopInstances",
        "ec2:StartInstances"
    ],
    "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0",
        "arn:aws:ec2:us-east-1:123456789012:instance/i-0598c7d356eba48d7"
    ]
},
{
    "Effect": "Allow",
    "Action": "ec2:TerminateInstances",
    "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
    "Condition": {
        "StringEquals": {
            "aws:TagKeys": "purpose"
        }
    }
}
]
}

```

```
        "Condition": {
            "StringEquals": {
                "ec2:ResourceTag/purpose": "test"
            }
        }
    ]
}
```

Working with volumes

Examples

- [Example: Attaching and detaching volumes \(p. 932\)](#)
- [Example: Creating a volume \(p. 933\)](#)
- [Example: Creating a volume with tags \(p. 933\)](#)

Example: Attaching and detaching volumes

When an API action requires a caller to specify multiple resources, you must create a policy statement that allows users to access all required resources. If you need to use a Condition element with one or more of these resources, you must create multiple statements as shown in this example.

The following policy allows users to attach volumes with the tag "volume_user=*iam-user-name*" to instances with the tag "department=dev", and to detach those volumes from those instances. If you attach this policy to an IAM group, the `aws:username` policy variable gives each IAM user in the group permission to attach or detach volumes from the instances with a tag named `volume_user` that has his or her IAM user name as a value.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/department": "dev"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/volume_user": "${aws:username}"
                }
            }
        }
    ]
}
```

Example: Creating a volume

The following policy allows users to use the [CreateVolume](#) API action. The user is allowed to create a volume only if the volume is encrypted and only if the volume size is less than 20 GiB.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateVolume"  
            ],  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",  
            "Condition":{  
                "NumericLessThan": {  
                    "ec2:VolumeSize" : "20"  
                },  
                "Bool":{  
                    "ec2:Encrypted" : "true"  
                }  
            }  
        }  
    ]  
}
```

Example: Creating a volume with tags

The following policy includes the `aws:RequestTag` condition key that requires users to tag any volumes they create with the tags `costcenter=115` and `stack=prod`. The `aws:TagKeys` condition key uses the `ForAllValues` modifier to indicate that only the keys `costcenter` and `stack` are allowed in the request (no other tags can be specified). If users don't pass these specific tags, or if they don't specify tags at all, the request fails.

For resource-creating actions that apply tags, users must also have permissions to use the `CreateTags` action. The second statement uses the `ec2:CreateAction` condition key to allow users to create tags only in the context of `CreateVolume`. Users cannot tag existing volumes or any other resources. For more information, see [Granting permission to tag resources during creation \(p. 926\)](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowCreateTaggedVolumes",  
            "Effect": "Allow",  
            "Action": "ec2:CreateVolume",  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/costcenter": "115",  
                    "aws:RequestTag/stack": "prod"  
                },  
                "ForAllValues:StringEquals": {  
                    "aws:TagKeys": ["costcenter","stack"]  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/costcenter": "115",  
                    "aws:RequestTag/stack": "prod"  
                },  
                "ForAllValues:StringEquals": {  
                    "aws:TagKeys": ["costcenter","stack"]  
                }  
            }  
        }  
    ]  
}
```

```
    "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
    "Condition": [
        "StringEquals": {
            "ec2:CreateAction" : "CreateVolume"
        }
    ]
}
```

The following policy allows users to create a volume without having to specify tags. The `CreateTags` action is only evaluated if tags are specified in the `CreateVolume` request. If users do specify tags, the tag must be `purpose=test`. No other tags are allowed in the request.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateVolume",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:1234567890:volume/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/purpose": "test",
                    "ec2:CreateAction" : "CreateVolume"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "purpose"
                }
            }
        }
    ]
}
```

Working with snapshots

The following are example policies for both `CreateSnapshot` (point-in-time snapshot of an EBS volume) and `CreateSnapshots` (multi-volume snapshots).

Examples

- [Example: Creating a snapshot \(p. 934\)](#)
- [Example: Creating snapshots \(p. 935\)](#)
- [Example: Creating a snapshot with tags \(p. 935\)](#)
- [Example: Creating snapshots with tags \(p. 936\)](#)
- [Example: Modifying permission settings for snapshots \(p. 941\)](#)

Example: Creating a snapshot

The following policy allows customers to use the `CreateSnapshot` API action. The customer can create snapshots only if the volume is encrypted and only if the volume size is less than 20 GiB.

```
{
```

```
"Version":"2012-10-17",
"Statement": [
    {
        "Effect":"Allow",
        "Action":"ec2:CreateSnapshot",
        "Resource":"arn:aws:ec2:us-east-1::snapshot/*"
    },
    {
        "Effect":"Allow",
        "Action":"ec2:CreateSnapshot",
        "Resource":"arn:aws:ec2:us-east-1:123456789012:volume/*",
        "Condition":{
            "NumericLessThan":{
                "ec2:VolumeSize":"20"
            },
            "Bool":{
                "ec2:Encrypted":"true"
            }
        }
    }
]
```

Example: Creating snapshots

The following policy allows customers to use the [CreateSnapshots](#) API action. The customer can create snapshots only if all of the volumes on the instance are type GP2.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect":"Allow",
            "Action":"ec2:CreateSnapshots",
            "Resource":[
                "arn:aws:ec2:us-east-1::snapshot/*",
                "arn:aws:ec2:*::instance/*"
            ]
        },
        {
            "Effect":"Allow",
            "Action":"ec2:CreateSnapshots",
            "Resource":"arn:aws:ec2:us-east-1::*:volume/*",
            "Condition":{
                "StringLikeIfExists":{
                    "ec2:VolumeType":"gp2"
                }
            }
        }
    ]
}
```

Example: Creating a snapshot with tags

The following policy includes the `aws:RequestTag` condition key that requires the customer to apply the tags `costcenter=115` and `stack=prod` to any new snapshot. The `aws:TagKeys` condition key uses the `ForAllValues` modifier to indicate that only the keys `costcenter` and `stack` can be specified in the request. The request fails if either of these conditions is not met.

For resource-creating actions that apply tags, customers must also have permissions to use the `CreateTags` action. The third statement uses the `ec2:CreateAction` condition key to allow customers to create tags only in the context of `CreateSnapshot`. Customers cannot tag existing

volumes or any other resources. For more information, see [Granting permission to tag resources during creation \(p. 926\)](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:CreateSnapshot",  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*"  
        },  
        {  
            "Sid": "AllowCreateTaggedSnapshots",  
            "Effect": "Allow",  
            "Action": "ec2:CreateSnapshot",  
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/costcenter": "115",  
                    "aws:RequestTag/stack": "prod"  
                },  
                "ForAllValues:StringEquals": {  
                    "aws:TagKeys": [  
                        "costcenter",  
                        "stack"  
                    ]  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:CreateTags",  
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:CreateAction": "CreateSnapshot"  
                }  
            }  
        }  
    ]  
}
```

Example: Creating snapshots with tags

The following policy includes the `aws:RequestTag` condition key that requires the customer to apply the tags `costcenter=115` and `stack=prod` to any new snapshot. The `aws:TagKeys` condition key uses the `ForAllValues` modifier to indicate that only the keys `costcenter` and `stack` can be specified in the request. The request fails if either of these conditions is not met.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:CreateSchedules",  
            "Resource": [  
                "arn:aws:ec2:us-east-1::snapshot/*",  
                "arn:aws:ec2:/*::instance/*",  
                "arn:aws:ec2:/*::volume/*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:CreateSnapshot",  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/costcenter": "115",  
                    "aws:RequestTag/stack": "prod"  
                },  
                "ForAllValues:StringEquals": {  
                    "aws:TagKeys": [  
                        "costcenter",  
                        "stack"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

```

    "Sid": "AllowCreateTaggedSnapshots",
    "Effect": "Allow",
    "Action": "ec2:CreateSnapshots",
    "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/costcenter": "115",
            "aws:RequestTag/stack": "prod"
        },
        "ForAllValues:StringEquals": {
            "aws:TagKeys": [
                "costcenter",
                "stack"
            ]
        }
    },
    {
        "Effect": "Allow",
        "Action": "ec2:CreateTags",
        "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
        "Condition": {
            "StringEquals": {
                "ec2:CreateAction": "CreateSnapshots"
            }
        }
    }
}
]
}

```

The following policy allows customers to create a snapshot without having to specify tags. The `CreateTags` action is evaluated only if tags are specified in the `CreateSnapshot` or `CreateSnapshots` request. If a tag is specified, the tag must be `purpose=test`. No other tags are allowed in the request.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshot",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/purpose": "test",
                    "ec2:CreateAction": "CreateSnapshot"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "purpose"
                }
            }
        }
    ]
}

```

```
{
    "Version": "2012-10-17",
    "Statement": [

```

```
{
    "Effect": "Allow",
    "Action": "ec2:CreateSnapshots",
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/purpose": "test",
            "ec2:CreateAction": "CreateSnapshots"
        },
        "ForAllValues:StringEquals": {
            "aws:TagKeys": "purpose"
        }
    }
}
]
```

The following policy allows snapshots to be created only if the source volume is tagged with `User:username` for the customer, and the snapshot itself is tagged with `Environment:Dev` and `User:username`. The customer can add additional tags to the snapshot.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshot",
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/User": "${aws:username}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshot",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Environment": "Dev",
                    "aws:RequestTag/User": "${aws:username}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*"
        }
    ]
}
```

The following policy for `CreateSnapshots` allows snapshots to be created only if the source volume is tagged with `User:username` for the customer, and the snapshot itself is tagged with `Environment:Dev` and `User:username`.

```
{
```

```

"Version":"2012-10-17",
"Statement": [
    {
        "Effect":"Allow",
        "Action":"ec2:CreateSnapshots",
        "Resource":"arn:aws:ec2:us-east-1::*:instance/*",
    },
    {
        "Effect":"Allow",
        "Action":"ec2:CreateSnapshots",
        "Resource":"arn:aws:ec2:us-east-1:123456789012:volume/*",
        "Condition":{
            "StringEquals":{
                "ec2:ResourceTag/User":"${aws:username}"
            }
        }
    },
    {
        "Effect":"Allow",
        "Action":"ec2:CreateSnapshots",
        "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
        "Condition":{
            "StringEquals":{
                "aws:RequestTag/Environment":"Dev",
                "aws:RequestTag/User":"${aws:username}"
            }
        }
    },
    {
        "Effect":"Allow",
        "Action":"ec2:CreateTags",
        "Resource":"arn:aws:ec2:us-east-1::snapshot/*"
    }
]
}

```

The following policy allows deletion of a snapshot only if the snapshot is tagged with User:*username* for the customer.

```

{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect":"Allow",
            "Action":"ec2>DeleteSnapshot",
            "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
            "Condition":{
                "StringEquals":{
                    "ec2:ResourceTag/User":"${aws:username}"
                }
            }
        }
    ]
}

```

The following policy allows a customer to create a snapshot but denies the action if the snapshot being created has a tag key value=stack.

```

{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect":"Allow",

```

```

    "Action": [
        "ec2:CreateSnapshot",
        "ec2:CreateTags"
    ],
    "Resource": "*"
},
{
    "Effect": "Deny",
    "Action": "ec2:CreateSnapshot",
    "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:TagKeys": "stack"
        }
    }
}
]
}

```

The following policy allows a customer to create snapshots but denies the action if the snapshots being created have a tag key value=stack.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshots",
                "ec2:CreateTags"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": "ec2:CreateSnapshots",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": "stack"
                }
            }
        }
    ]
}

```

The following policy allows you to combine multiple actions into a single policy. You can only create a snapshot (in the context of CreateSnapshots) when the snapshot is created in Region us-east-1. You can only create snapshots (in the context of CreateSnapshot) when the snapshots are being created in the Region us-east-1 and when the instance type is t2*.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshots",
                "ec2:CreateSnapshot",
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::instance/*",

```

```
        "arn:aws:ec2:*::snapshot/*",
        "arn:aws:ec2:*::volume/*"
    ],
    "Condition": {
        "StringEqualsIgnoreCase": {
            "ec2:Region": "us-east-1"
        },
        "StringLikeIfExists": {
            "ec2:InstanceType": ["t2.*"]
        }
    }
}
]
```

Example: Modifying permission settings for snapshots

The following policy allows modification of a snapshot only if the snapshot is tagged with `User:username`, where `username` is the customer's AWS account user name. The request fails if this condition is not met.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2: ModifySnapshotAttribute",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/user-name": "${aws:username}"
                }
            }
        }
    ]
}
```

Launching instances (RunInstances)

The [RunInstances](#) API action launches one or more On-Demand Instances or one or more Spot Instances. `RunInstances` requires an AMI and creates an instance. Users can specify a key pair and security group in the request. Launching into a VPC requires a subnet, and creates a network interface. Launching from an Amazon EBS-backed AMI creates a volume. Therefore, the user must have permissions to use these Amazon EC2 resources. You can create a policy statement that requires users to specify an optional parameter on `RunInstances`, or restricts users to particular values for a parameter.

For more information about the resource-level permissions that are required to launch an instance, see [Actions, Resources, and Condition Keys for Amazon EC2](#) in the *IAM User Guide*.

By default, users don't have permissions to describe, start, stop, or terminate the resulting instances. One way to grant the users permission to manage the resulting instances is to create a specific tag for each instance, and then create a statement that enables them to manage instances with that tag. For more information, see [Working with instances \(p. 930\)](#).

Resources

- [AMIs \(p. 942\)](#)
- [Instance types \(p. 943\)](#)
- [Subnets \(p. 944\)](#)
- [EBS volumes \(p. 945\)](#)
- [Tags \(p. 945\)](#)

- [Tags in a launch template \(p. 949\)](#)
- [Elastic GPUs \(p. 950\)](#)
- [Launch templates \(p. 951\)](#)

AMIs

The following policy allows users to launch instances using only the specified AMIs, `ami-9e1670f7` and `ami-45cf5c3c`. The users can't launch an instance using other AMIs (unless another statement grants the users permission to do so).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:region::image/ami-9e1670f7",  
                "arn:aws:ec2:region::image/ami-45cf5c3c",  
                "arn:aws:ec2:region:account:instance/*",  
                "arn:aws:ec2:region:account:volume/*",  
                "arn:aws:ec2:region:account:key-pair/*",  
                "arn:aws:ec2:region:account:security-group/*",  
                "arn:aws:ec2:region:account:subnet/*",  
                "arn:aws:ec2:region:account:network-interface/*"  
            ]  
        }  
    ]  
}
```

Alternatively, the following policy allows users to launch instances from all AMIs owned by Amazon. The `Condition` element of the first statement tests whether `ec2:Owner` is `amazon`. The users can't launch an instance using other AMIs (unless another statement grants the users permission to do so).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:region::image/ami-*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "ec2:Owner": "amazon"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:region:account:instance/*",  
                "arn:aws:ec2:region:account:subnet/*",  
                "arn:aws:ec2:region:account:volume/*",  
                "arn:aws:ec2:region:account:network-interface/*",  
                "arn:aws:ec2:region:account:key-pair/*",  
                "arn:aws:ec2:region:account:security-group/*"  
            ]  
        }  
    ]  
}
```

```
}
```

Instance types

The following policy allows users to launch instances using only the `t2.micro` or `t2.small` instance type, which you might do to control costs. The users can't launch larger instances because the `Condition` element of the first statement tests whether `ec2:InstanceType` is either `t2.micro` or `t2.small`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region:account:instance/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:InstanceType": ["t2.micro", "t2.small"]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region::image/ami-*",
                "arn:aws:ec2:region:account:subnet/*",
                "arn:aws:ec2:region:account:network-interface/*",
                "arn:aws:ec2:region:account:volume/*",
                "arn:aws:ec2:region:account:key-pair/*",
                "arn:aws:ec2:region:account:security-group/*"
            ]
        }
    ]
}
```

Alternatively, you can create a policy that denies users permissions to launch any instances except `t2.micro` and `t2.small` instance types.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region:account:instance/*"
            ],
            "Condition": {
                "StringNotEquals": {
                    "ec2:InstanceType": ["t2.micro", "t2.small"]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region::image/ami-*",
                "arn:aws:ec2:region:account:network-interface/*",

```

```
        "arn:aws:ec2:region:account:instance/*",
        "arn:aws:ec2:region:account:subnet/*",
        "arn:aws:ec2:region:account:volume/*",
        "arn:aws:ec2:region:account:key-pair/*",
        "arn:aws:ec2:region:account:security-group/*"
    ]
}
]
```

Subnets

The following policy allows users to launch instances using only the specified subnet, subnet-12345678. The group can't launch instances into any another subnet (unless another statement grants the users permission to do so).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region:account:subnet/subnet-12345678",
                "arn:aws:ec2:region:account:network-interface/*",
                "arn:aws:ec2:region:account:instance/*",
                "arn:aws:ec2:region:account:volume/*",
                "arn:aws:ec2:region::image/ami-*",
                "arn:aws:ec2:region:account:key-pair/*",
                "arn:aws:ec2:region:account:security-group/*"
            ]
        }
    ]
}
```

Alternatively, you could create a policy that denies users permissions to launch an instance into any other subnet. The statement does this by denying permission to create a network interface, except where subnet subnet-12345678 is specified. This denial overrides any other policies that are created to allow launching instances into other subnets.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region:account:network-interface/*"
            ],
            "Condition": {
                "ArnNotEquals": {
                    "ec2:Subnet": "arn:aws:ec2:region:account:subnet/subnet-12345678"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region::image/ami-*",
                "arn:aws:ec2:region:account:network-interface/*",
                "arn:aws:ec2:region:account:instance/*",
                "arn:aws:ec2:region:account:subnet/*",
                "arn:aws:ec2:region:account:volume/*",
                "arn:aws:ec2:region:account:key-pair/*",
                "arn:aws:ec2:region:account:security-group/*"
            ]
        }
    ]
}
```

```
        "arn:aws:ec2:region:account:volume/*",
        "arn:aws:ec2:region:account:key-pair/*",
        "arn:aws:ec2:region:account:security-group/*"
    ]
}
}
```

EBS volumes

The following policy allows users to launch instances only if the EBS volumes for the instance are encrypted. The user must launch an instance from an AMI that was created with encrypted snapshots, to ensure that the root volume is encrypted. Any additional volume that the user attaches to the instance during launch must also be encrypted.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:***:volume/*"
            ],
            "Condition": {
                "Bool": {
                    "ec2:Encrypted": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:***:image/ami-*",
                "arn:aws:ec2:***:network-interface/*",
                "arn:aws:ec2:***:instance/*",
                "arn:aws:ec2:***:subnet/*",
                "arn:aws:ec2:***:key-pair/*",
                "arn:aws:ec2:***:security-group/*"
            ]
        }
    ]
}
```

Tags

Tag instances on creation

The following policy allows users to launch instances and tag the instances during creation. For resource-creating actions that apply tags, users must have permissions to use the `CreateTags` action. The second statement uses the `ec2:CreateAction` condition key to allow users to create tags only in the context of `RunInstances`, and only for instances. Users cannot tag existing resources, and users cannot tag volumes using the `RunInstances` request.

For more information, see [Granting permission to tag resources during creation \(p. 926\)](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [

```

```

        "ec2:RunInstances"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction" : "RunInstances"
        }
    }
}
]
}

```

Tag instances and volumes on creation with specific tags

The following policy includes the `aws:RequestTag` condition key that requires users to tag any instances and volumes that are created by `RunInstances` with the tags `environment=production` and `purpose=webserver`. The `aws:TagKeys` condition key uses the `ForAllValues` modifier to indicate that only the keys `environment` and `purpose` are allowed in the request (no other tags can be specified). If no tags are specified in the request, the request fails.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:region::image/*",
                "arn:aws:ec2:region:account:subnet/*",
                "arn:aws:ec2:region:account:network-interface/*",
                "arn:aws:ec2:region:account:security-group/*",
                "arn:aws:ec2:region:account:key-pair/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:region:account:volume/*",
                "arn:aws:ec2:region:account:instance/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production" ,
                    "aws:RequestTag/purpose": "webserver"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": ["environment", "purpose"]
                }
            }
        },
        {
            "Effect": "Allow",

```

```
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:region:account:/*/*",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction" : "RunInstances"
        }
    }
}
```

Tag instances and volumes on creation with at least one specific tag

The following policy uses the `ForAnyValue` modifier on the `aws:TagKeys` condition to indicate that at least one tag must be specified in the request, and it must contain the key `environment` or `webserver`. The tag must be applied to both instances and volumes. Any tag values can be specified in the request.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:region::image/*",
                "arn:aws:ec2:region:account:subnet/*",
                "arn:aws:ec2:region:account:network-interface/*",
                "arn:aws:ec2:region:account:security-group/*",
                "arn:aws:ec2:region:account:key-pair/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:region:account:volume/*",
                "arn:aws:ec2:region:account:instance/*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": ["environment", "webserver"]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:region:account:/*/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction" : "RunInstances"
                }
            }
        }
    ]
}
```

If instances are tagged on creation, they must be tagged with a specific tag

In the following policy, users do not have to specify tags in the request, but if they do, the tag must be `purpose=test`. No other tags are allowed. Users can apply the tags to any taggable resource in the `RunInstances` request.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags"  
            ],  
            "Resource": "arn:aws:ec2:region:account:*/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/purpose": "test",  
                    "ec2:CreateAction" : "RunInstances"  
                },  
                "ForAllValues:StringEquals": {  
                    "aws:TagKeys": "purpose"  
                }  
            }  
        }  
    ]  
}
```

To disallow anyone called tag on create for RunInstances

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowRun",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances"  
            ],  
            "Resource": [  
                "arn:aws:ec2:us-east-1::image/*",  
                "arn:aws:ec2:us-east-1::subnet/*",  
                "arn:aws:ec2:us-east-1::network-interface/*",  
                "arn:aws:ec2:us-east-1::security-group/*",  
                "arn:aws:ec2:us-east-1::key-pair/*",  
                "arn:aws:ec2:us-east-1::volume/*",  
                "arn:aws:ec2:us-east-1::instance/*",  
                "arn:aws:ec2:us-east-1::spot-instances-request/*"  
            ]  
        },  
        {  
            "Sid": "VisualEditor0",  
            "Effect": "Deny",  
            "Action": "ec2:CreateTags",  
            "Resource": "*"  
        }  
    ]  
}
```

```
}
```

Only allow specific tags for spot-instances-request. Surprise inconsistency number 2 comes into play here. Under normal circumstances, specifying no tags will result in Unauthenticated. In the case of spot-instances-request, this policy will not be evaluated if there are no spot-instances-request tags, so a non-tag Spot on Run request will succeed.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1::subnet/*",
                "arn:aws:ec2:us-east-1::network-interface/*",
                "arn:aws:ec2:us-east-1::security-group/*",
                "arn:aws:ec2:us-east-1::key-pair/*",
                "arn:aws:ec2:us-east-1::volume/*",
                "arn:aws:ec2:us-east-1::instance/*",
            ]
        },
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1::spot-instances-request/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        }
    ]
}
```

Tags in a launch template

In the following example, users can launch instances, but only if they use a specific launch template (lt-09477bcd97b0d310e). The ec2: IsLaunchTemplateResource condition key prevents users from overriding any of the resources specified in the launch template. The second part of the statement allows users to tag instances on creation—this part of the statement is necessary if tags are specified for the instance in the launch template.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/
lt-09477bcd97b0d310e"
                },
                "Bool": {
                    "ec2:IsLaunchTemplateResource": "true"
                }
            }
        }
    ]
}
```

```
        }
    },
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:region:account:instance/*",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction" : "RunInstances"
        }
    }
}
]
```

Elastic GPUs

In the following policy, users can launch an instance and specify an elastic GPU to attach to the instance. Users can launch instances in any Region, but they can only attach an elastic GPU during a launch in the us-east-2 Region.

The `ec2:ElasticGpuType` condition key uses the `ForAnyValue` modifier to indicate that only the elastic GPU types `eg1.medium` and `eg1.large` are allowed in the request.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:*:account:elastic-gpu/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:Region": "us-east-2"
                },
                "ForAnyValue:StringLike": {
                    "ec2:ElasticGpuType": [
                        "eg1.medium",
                        "eg1.large"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2::image/ami-*",
                "arn:aws:ec2::*:account:network-interface/*",
                "arn:aws:ec2::*:account:instance/*",
                "arn:aws:ec2::*:account:subnet/*",
                "arn:aws:ec2::*:account:volume/*",
                "arn:aws:ec2::*:account:key-pair/*",
                "arn:aws:ec2::*:account:security-group/*"
            ]
        }
    ]
}
```

```
}
```

Launch templates

In the following example, users can launch instances, but only if they use a specific launch template (lt-09477bcd97b0d310e). Users can override any parameters in the launch template by specifying the parameters in the RunInstances action.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/
lt-09477bcd97b0d310e"
                }
            }
        }
    ]
}
```

In this example, users can launch instances only if they use a launch template. The policy uses the `ec2:IsLaunchTemplateResource` condition key to prevent users from overriding any pre-existing ARNs in the launch template.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/*"
                },
                "Bool": {
                    "ec2:IsLaunchTemplateResource": "true"
                }
            }
        }
    ]
}
```

The following example policy allows user to launch instances, but only if they use a launch template. Users cannot override the subnet and network interface parameters in the request; these parameters can only be specified in the launch template. The first part of the statement uses the `NotResource` element to allow all other resources except subnets and network interfaces. The second part of the statement allows the subnet and network interface resources, but only if they are sourced from the launch template.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",

```

```

    "Action": "ec2:RunInstances",
    "NotResource": [ "arn:aws:ec2:region:account:subnet/*",
                    "arn:aws:ec2:region:account:network-interface/*" ],
    "Condition": {
        "ArnLike": {
            "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/*"
        }
    },
    {
        "Effect": "Allow",
        "Action": "ec2:RunInstances",
        "Resource": [ "arn:aws:ec2:region:account:subnet/*",
                      "arn:aws:ec2:region:account:network-interface/*" ],
        "Condition": {
            "ArnLike": {
                "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/*"
            },
            "Bool": {
                "ec2:IsLaunchTemplateResource": "true"
            }
        }
    }
}
]
}

```

The following example allows users to launch instances only if they use a launch template, and only if the launch template has the tag Purpose=Webservers. Users cannot override any of the launch template parameters in the RunInstances action.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "NotResource": "arn:aws:ec2:region:account:launch-template/*",
            "Condition": {
                "ArnLike": {
                    "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/*"
                },
                "Bool": {
                    "ec2:IsLaunchTemplateResource": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:region:account:launch-template/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Purpose": "Webservers"
                }
            }
        }
    ]
}

```

Working with Spot Instances

You can use the RunInstances action to create Spot Instance requests, and tag the Spot Instance requests on create. The resource to specify for RunInstances is spot-instances-request.

The `spot-instances-request` resource is evaluated in the IAM policy as follows:

- If you don't tag a Spot Instance request on create, Amazon EC2 does not evaluate the `spot-instances-request` resource in the `RunInstances` statement.
- If you tag a Spot Instance request on create, Amazon EC2 evaluates the `spot-instances-request` resource in the `RunInstances` statement.

Therefore, for the `spot-instances-request` resource, the following rules apply to the IAM policy:

- If you use `RunInstances` to create a Spot Instance request and you don't intend to tag the Spot Instance request on create, you don't need to explicitly allow the `spot-instances-request` resource; the call will succeed.
- If you use `RunInstances` to create a Spot Instance request and intend to tag the Spot Instance request on create, you must include the `spot-instances-request` resource in the `RunInstances` `allow` statement, otherwise the call will fail.
- If you use `RunInstances` to create a Spot Instance request and intend to tag the Spot Instance request on create, you must specify the `spot-instances-request` resource or `*` wildcard in the `CreateTags` `allow` statement, otherwise the call will fail.

You can request Spot Instances using `RunInstances` or `RequestSpotInstances`. The following example IAM policies apply only when requesting Spot Instances using `RunInstances`.

Example: Request Spot Instances using RunInstances

The following policy allows users to request Spot Instances by using the `RunInstances` action. The `spot-instances-request` resource, which is created by `RunInstances`, requests Spot Instances.

Note

To use `RunInstances` to create Spot Instance requests, you can omit `spot-instances-request` from the Resource list if you do not intend to tag the Spot Instance requests on create. This is because Amazon EC2 does not evaluate the `spot-instances-request` resource in the `RunInstances` statement if the Spot Instance request is not tagged on create.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowRun",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances"  
            ],  
            "Resource": [  
                "arn:aws:ec2:us-east-1::image/*",  
                "arn:aws:ec2:us-east-1::subnet/*",  
                "arn:aws:ec2:us-east-1::network-interface/*",  
                "arn:aws:ec2:us-east-1::security-group/*",  
                "arn:aws:ec2:us-east-1::key-pair/*",  
                "arn:aws:ec2:us-east-1::volume/*",  
                "arn:aws:ec2:us-east-1::instance/*",  
                "arn:aws:ec2:us-east-1::spot-instances-request/*"  
            ]  
        }  
    ]  
}
```

Warning

NOT SUPPORTED – Example: Deny users permission to request Spot Instances using RunInstances

The following policy is not supported for the `spot-instances-request` resource. The following policy is meant to give users the permission to launch On-Demand Instances, but deny users the permission to request Spot Instances. The `spot-instances-request` resource, which is created by `RunInstances`, is the resource that requests Spot Instances. The second statement is meant to deny the `RunInstances` action for the `spot-instances-request` resource. However, this condition is not supported because Amazon EC2 does not evaluate the `spot-instances-request` resource in the `RunInstances` statement if the Spot Instance request is not tagged on create.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowRun",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances"  
            ],  
            "Resource": [  
                "arn:aws:ec2:us-east-1::image/*",  
                "arn:aws:ec2:us-east-1::subnet/*",  
                "arn:aws:ec2:us-east-1::network-interface/*",  
                "arn:aws:ec2:us-east-1::security-group/*",  
                "arn:aws:ec2:us-east-1::key-pair/*",  
                "arn:aws:ec2:us-east-1::volume/*",  
                "arn:aws:ec2:us-east-1::instance/*"  
            ]  
        },  
        {  
            "Sid": "DenySpotInstancesRequests - NOT SUPPORTED - DO NOT USE!",  
            "Effect": "Deny",  
            "Action": "ec2:RunInstances",  
            "Resource": "arn:aws:ec2:us-east-1::spot-instances-request/*"  
        }  
    ]  
}
```

Example: Tag Spot Instance requests on create

The following policy allows users to tag all resources that are created during instance launch. The first statement allows `RunInstances` to create the listed resources. The `spot-instances-request` resource, which is created by `RunInstances`, is the resource that requests Spot Instances. The second statement provides a * wildcard to allow all resources to be tagged when they are created at instance launch.

Note

If you tag a Spot Instance request on create, Amazon EC2 evaluates the `spot-instances-request` resource in the `RunInstances` statement. Therefore, you must explicitly allow the `spot-instances-request` resource for the `RunInstances` action, otherwise the call will fail.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowRun",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances"  
            ],  
            "Resource": [  
                "arn:aws:ec2:us-east-1::image/*",  
                "arn:aws:ec2:us-east-1::subnet/*",  
                "arn:aws:ec2:us-east-1::network-interface/*",  
                "arn:aws:ec2:us-east-1::volume/*",  
                "arn:aws:ec2:us-east-1::instance/*"  
            ]  
        }  
    ]  
}
```

```
        "arn:aws:ec2:us-east-1::security-group/*",
        "arn:aws:ec2:us-east-1::key-pair/*",
        "arn:aws:ec2:us-east-1::volume/*",
        "arn:aws:ec2:us-east-1::instance/*",
        "arn:aws:ec2:us-east-1::spot-instances-request/*"
    ],
},
{
    "Sid": "TagResources",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "*"
}
]
```

Example: Deny tag on create for Spot Instance requests

The following policy denies users the permission to tag the resources that are created during instance launch.

The first statement allows RunInstances to create the listed resources. The spot-instances-request resource, which is created by RunInstances, is the resource that requests Spot Instances. The second statement provides a * wildcard to deny all resources being tagged when they are created at instance launch. If spot-instances-request or any other resource is tagged on create, the RunInstances call will fail.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1::subnet/*",
                "arn:aws:ec2:us-east-1::network-interface/*",
                "arn:aws:ec2:us-east-1::security-group/*",
                "arn:aws:ec2:us-east-1::key-pair/*",
                "arn:aws:ec2:us-east-1::volume/*",
                "arn:aws:ec2:us-east-1::instance/*",
                "arn:aws:ec2:us-east-1::spot-instances-request/*"
            ]
        },
        {
            "Sid": "DenyTagResources",
            "Effect": "Deny",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

Warning

NOT SUPPORTED – Example: Allow creating a Spot Instance request only if it is assigned a specific tag

The following policy is not supported for the spot-instances-request resource.

The following policy is meant to grant RunInstances the permission to create a Spot Instance request only if the request is tagged with a specific tag.

The first statement allows RunInstances to create the listed resources.

The second statement is meant to grant users the permission to create a Spot Instance request only if the request has the tag `environment=production`. If this condition is applied to other resources created by RunInstances, specifying no tags results in an Unauthenticated error. However, if no tags are specified for the Spot Instance request, Amazon EC2 does not evaluate the `spot-instances-request` resource in the RunInstances statement, which results in non-tagged Spot Instance requests being created by RunInstances.

Note that specifying another tag other than `environment=production` results in an Unauthenticated error, because if a user tags a Spot Instance request, Amazon EC2 evaluates the `spot-instances-request` resource in the RunInstances statement.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowRun",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances"  
            ],  
            "Resource": [  
                "arn:aws:ec2:us-east-1::image/*",  
                "arn:aws:ec2:us-east-1::subnet/*",  
                "arn:aws:ec2:us-east-1::network-interface/*",  
                "arn:aws:ec2:us-east-1::security-group/*",  
                "arn:aws:ec2:us-east-1::key-pair/*",  
                "arn:aws:ec2:us-east-1::volume/*",  
                "arn:aws:ec2:us-east-1::instance/*"  
            ]  
        },  
        {  
            "Sid": "RequestSpotInstancesOnlyIfTagIs_environment=production - NOT  
SUPPORTED - DO NOT USE!",  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": "arn:aws:ec2:us-east-1::spot-instances-request/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/environment": "production"  
                }  
            }  
        },  
        {  
            "Sid": "TagResources",  
            "Effect": "Allow",  
            "Action": "ec2:CreateTags",  
            "Resource": "*"  
        }  
    ]  
}
```

Example: Deny creating a Spot Instance request if it is assigned a specific tag

The following policy denies RunInstances the permission to create a Spot Instance request if the request is tagged with `environment=production`.

The first statement allows RunInstances to create the listed resources.

The second statement denies users the permission to create a Spot Instance request if the request has the tag `environment=production`. Specifying `environment=production` as a tag results in an Unauthenticated error. Specifying other tags or specifying no tags will result in the creation of a Spot Instance request.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1::image/*",
                "arn:aws:ec2:us-east-1::subnet/*",
                "arn:aws:ec2:us-east-1::network-interface/*",
                "arn:aws:ec2:us-east-1::security-group/*",
                "arn:aws:ec2:us-east-1::key-pair/*",
                "arn:aws:ec2:us-east-1::volume/*",
                "arn:aws:ec2:us-east-1::instance/*",
                "arn:aws:ec2:us-east-1::spot-instances-request/*"
            ]
        },
        {
            "Sid": "DenySpotInstancesRequests",
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1::spot-instances-request/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        },
        {
            "Sid": "TagResources",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

Example: Working with Reserved Instances

The following policy gives users permission to view, modify, and purchase Reserved Instances in your account.

It is not possible to set resource-level permissions for individual Reserved Instances. This policy means that users have access to all the Reserved Instances in the account.

The `Resource` element uses a `*` wildcard to indicate that users can specify all resources with the action; in this case, they can list and modify all Reserved Instances in the account. They can also purchase Reserved Instances using the account credentials. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeReservedInstances",
                "ec2:ModifyReservedInstances",
                "ec2:PurchaseReservedInstancesOffering",

```

```
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeReservedInstancesOfferings"
    ],
    "Resource": "*"
}
]
```

To allow users to view and modify the Reserved Instances in your account, but not purchase new Reserved Instances.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeReservedInstances",
                "ec2:ModifyReservedInstances",
                "ec2:DescribeAvailabilityZones"
            ],
            "Resource": "*"
        }
    ]
}
```

Example: Tagging resources

The following policy allows users to use the `CreateTags` action to apply tags to an instance only if the tag contains the key `environment` and the value `production`. The `ForAllValues` modifier is used with the `aws:TagKeys` condition key to indicate that only the key `environment` is allowed in the request (no other tags are allowed). The user cannot tag any other resource types.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:region:account:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "environment"
                    ]
                }
            }
        }
    ]
}
```

The following policy allows users to tag any taggable resource that already has a tag with a key of `owner` and a value of the IAM username. In addition, users must specify a tag with a key of `anycompany:environment-type` and a value of either `test` or `prod` in the request. Users can specify additional tags in the request.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags"  
            ],  
            "Resource": "arn:aws:ec2:region:account:/*/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/anycompany:environment-type": ["test", "prod"],  
                    "ec2:ResourceTag/owner": "${aws:username}"  
                }  
            }  
        }  
    ]  
}
```

You can create an IAM policy that allows users to delete specific tags for a resource. For example, the following policy allows users to delete tags for a volume if the tag keys specified in the request are `environment` or `cost-center`. Any value can be specified for the tag but the tag key must match either of the specified keys.

Note

If you delete a resource, all tags associated with the resource are also deleted. Users do not need permissions to use the `ec2:DeleteTags` action to delete a resource that has tags; they only need permissions to perform the deleting action.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:DeleteTags",  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",  
            "Condition": {  
                "ForAllValues:StringEquals": {  
                    "aws:TagKeys": ["environment", "cost-center"]  
                }  
            }  
        }  
    ]  
}
```

This policy allows users to delete only the `environment=prod` tag on any resource, and only if the resource is already tagged with a key of `owner` and a value of the IAM username. Users cannot delete any other tags for a resource.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DeleteTags"  
            ],  
            "Resource": "arn:aws:ec2:region:account:/*/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/environment": "prod",  
                    "aws:RequestTag/owner": "${aws:username}"  
                }  
            }  
        }  
    ]  
}
```

```
        "ec2:ResourceTag/owner": "${aws:username}"
    },
    "ForAllValues:StringEquals": {
        "aws:TagKeys": [ "environment" ]
    }
}
]
```

Example: Working with IAM roles

The following policy allows users to attach, replace, and detach an IAM role to instances that have the tag `department=test`. Replacing or detaching an IAM role requires an association ID, therefore the policy also grants users permission to use the `ec2:DescribeIamInstanceProfileAssociations` action.

IAM users must have permission to use the `iam:PassRole` action in order to pass the role to the instance.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AssociateIamInstanceProfile",
                "ec2:ReplaceIamInstanceProfileAssociation",
                "ec2:DisassociateIamInstanceProfile"
            ],
            "Resource": "arn:aws:ec2:region:account:instance/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/department": "test"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeIamInstanceProfileAssociations",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*"
        }
    ]
}
```

The following policy allows users to attach or replace an IAM role for any instance. Users can only attach or replace IAM roles with names that begin with `TestRole-`. For the `iam:PassRole` action, ensure that you specify the name of the IAM role and not the instance profile (if the names are different). For more information, see [Instance profiles \(p. 974\)](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [

```

```
        "ec2:AssociateIamInstanceProfile",
        "ec2:ReplaceIamInstanceProfileAssociation"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": "ec2:DescribeIamInstanceProfileAssociations",
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::account:role/TestRole-*"
}
]
```

Example: Working with route tables

The following policy allows users to add, remove, and replace routes for route tables that are associated with VPC vpc-ec43eb89 only. To specify a VPC for the ec2:Vpc condition key, you must specify the full ARN of the VPC.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteRoute",
                "ec2:CreateRoute",
                "ec2:ReplaceRoute"
            ],
            "Resource": [
                "arn:aws:ec2:region:account:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:Vpc": "arn:aws:ec2:region:account:vpc/vpc-ec43eb89"
                }
            }
        }
    ]
}
```

Example: Allowing a specific instance to view resources in other AWS services

The following is an example of a policy that you might attach to an IAM role. The policy allows an instance to view resources in various AWS services. It uses the ec2:SourceInstanceARN condition key to specify that the instance from which the request is made must be instance i-093452212644b0dd6. If the same IAM role is associated with another instance, the other instance cannot perform any of these actions.

The ec2:SourceInstanceARN key is an AWS-wide condition key, therefore it can be used for other service actions, not just Amazon EC2.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {

```

```
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeVolumes",
            "s3>ListAllMyBuckets",
            "dynamodb>ListTables",
            "rds:DescribeDBInstances"
        ],
        "Resource": [
            "*"
        ],
        "Condition": {
            "ArnEquals": {
                "ec2:SourceInstanceARN": "arn:aws:ec2:region:account:instance/i-093452212644b0dd6"
            }
        }
    ]
}
```

Example: Working with launch templates

The following policy allows users to create a launch template version and modify a launch template, but only for a specific launch template (`lt-09477bcd97b0d3abc`). Users cannot work with other launch templates.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2>CreateLaunchTemplateVersion",
                "ec2>ModifyLaunchTemplate"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:ec2:region:account:launch-template/lt-09477bcd97b0d3abc"
        }
    ]
}
```

The following policy allows users to delete any launch template and launch template version, provided that the launch template has the tag `Purpose=Testing`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2>DeleteLaunchTemplate",
                "ec2>DeleteLaunchTemplateVersions"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:ec2:region:account:launch-template/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Purpose": "Testing"
                }
            }
        }
    ]
}
```

Working with instance metadata

The following policies ensure that users can only retrieve [instance metadata \(p. 658\)](#) using Instance Metadata Service Version 2 (IMDSv2). You can combine the following four policies into one policy with four statements. When combined as one policy, you can use the policy as a service control policy (SCP). It can work equally well as a *deny* policy that you apply to an existing IAM policy (taking away and limiting existing permission), or as an SCP that is applied globally across an account, an organizational unit (OU), or an entire organization.

Note

The following RunInstances metadata options policies must be used in conjunction with a policy that gives the principal permissions to launch an instance with RunInstances. If the principal does not also have RunInstances permissions, it will not be able to launch an instance. For more information, see the policies in [Working with instances \(p. 930\)](#) and [Launching instances \(RunInstances\) \(p. 941\)](#).

Important

If you use Auto Scaling groups and you need to require the use of IMDSv2 on all new instances, your Auto Scaling groups must use *launch templates*.

When an Auto Scaling group uses a launch template, the `ec2:RunInstances` permissions of the IAM principal are checked when a new Auto Scaling group is created. They are also checked when an existing Auto Scaling group is updated to use a new launch template or a new version of a launch template.

Restrictions on the use of IMDSv1 on IAM principals for `RunInstances` are only checked when an Auto Scaling group that is using a launch template, is created or updated. For an Auto Scaling group that is configured to use the `Latest` or `Default` launch template, the permissions are not checked when a new version of the launch template is created. For permissions to be checked, you must configure the Auto Scaling group to use a *specific version* of the launch template.

To enforce the use of IMDSv2 on instances launched by Auto Scaling groups, the following additional steps are required:

1. Disable the use of launch configurations for all accounts in your organization by using either service control policies (SCPs) or IAM permissions boundaries for new principals that are created. For existing IAM principals with Auto Scaling group permissions, update their associated policies with this condition key. To disable the use of launch configurations, create or modify the relevant SCP, permissions boundary, or IAM policy with the "`autoscaling:LaunchConfigurationName`" condition key with the value specified as `null`.
2. For new launch templates, configure the instance metadata options in the launch template. For existing launch templates, create a new version of the launch template and configure the instance metadata options in the new version.
3. In the policy that gives any principal the permission to use a launch template, restrict association of `$latest` and `$default` by specifying `"autoscaling:LaunchTemplateVersionSpecified": "true"`. By restricting the use to a specific version of a launch template, you can ensure that new instances will be launched using the version in which the instance metadata options are configured. For more information, see [LaunchTemplateSpecification](#) in the *Amazon EC2 Auto Scaling API Reference*, specifically the `Version` parameter.
4. For an Auto Scaling group that uses a launch configuration, replace the launch configuration with a launch template. For more information, see [Replacing a Launch Configuration with a Launch Template](#) in the *Amazon EC2 Auto Scaling User Guide*.
5. For an Auto Scaling group that uses a launch template, make sure that it uses a new launch template with the instance metadata options configured, or uses a new version of the current launch template with the instance metadata options configured. For more information, see [update-auto-scaling-group](#) in the *AWS CLI Command Reference*.

Examples

- [Require the use of IMDSv2 \(p. 964\)](#)
- [Specify maximum hop limit \(p. 964\)](#)
- [Limit who can modify the instance metadata options \(p. 964\)](#)
- [Require role credentials to be retrieved from IMDSv2 \(p. 965\)](#)

Require the use of IMDSv2

The following policy specifies that you can't call the RunInstances API unless the instance is also opted in to require the use of IMDSv2 (indicated by "ec2:MetadataHttpTokens": "required"). If you do not specify that the instance requires IMDSv2, you get an `UnauthorizedOperation` error when you call the RunInstances API.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "RequireImdsV2",  
            "Effect": "Deny",  
            "Action": "ec2:RunInstances",  
            "Resource": "arn:aws:ec2:*::instance/*",  
            "Condition": {  
                "StringNotEquals": {  
                    "ec2:MetadataHttpTokens": "required"  
                }  
            }  
        }  
    ]  
}
```

Specify maximum hop limit

The following policy specifies that you can't call the RunInstances API unless you also specify a hop limit, and the hop limit can't be more than 3. If you fail to do that, you get an `UnauthorizedOperation` error when you call the RunInstances API.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "MaxImdsHopLimit",  
            "Effect": "Deny",  
            "Action": "ec2:RunInstances",  
            "Resource": "arn:aws:ec2:*::instance/*",  
            "Condition": {  
                "NumericGreaterThanOrEqual": {  
                    "ec2:MetadataHttpPutResponseHopLimit": "3"  
                }  
            }  
        }  
    ]  
}
```

Limit who can modify the instance metadata options

The following policy removes the ability for the general population of administrators to modify instance metadata options, and permits only users with the role `ec2-imds-admins` to make changes. If any principal other than the `ec2-imds-admins` role tries to call the `ModifyInstanceMetadataOptions` API, it will get an `UnauthorizedOperation` error. This statement could be used to control the use of the

ModifyInstanceMetadataOptions API; there are currently no fine-grained access controls (conditions) for the ModifyInstanceMetadataOptions API.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowOnlyImdsAdminsToModifySettings",  
            "Effect": "Deny",  
            "Action": "ec2:ModifyInstanceMetadataOptions",  
            "Resource": "*",  
            "Condition": {  
                "StringNotLike": {  
                    "aws:PrincipalARN": "arn:aws:iam::*:role/ec2-imds-admins"  
                }  
            }  
        }  
    ]  
}
```

Require role credentials to be retrieved from IMDSv2

The following policy specifies that if this policy is applied to a role, and the role is assumed by the EC2 service and the resulting credentials are used to sign a request, then the request must be signed by EC2 role credentials retrieved from IMDSv2. Otherwise, all of its API calls will get an UnauthorizedOperation error. This statement/policy can be applied generally because, if the request is not signed by EC2 role credentials, it has no effect.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "RequireAllEc2RolesToUseV2",  
            "Effect": "Deny",  
            "Action": "*",  
            "Resource": "*",  
            "Condition": {  
                "NumericLessThan": {  
                    "ec2:RoleDelivery": "2.0"  
                }  
            }  
        }  
    ]  
}
```

Example policies for working in the Amazon EC2 console

You can use IAM policies to grant users permissions to view and work with specific resources in the Amazon EC2 console. You can use the example policies in the previous section; however, they are designed for requests that are made with the AWS CLI or an AWS SDK. The console uses additional API actions for its features, so these policies may not work as expected. For example, a user that has permission to use only the `DescribeVolumes` API action will encounter errors when trying to view volumes in the console. This section demonstrates policies that enable users to work with specific parts of the console.

Tip

To help you work out which API actions are required to perform tasks in the console, you can use a service such as AWS CloudTrail. For more information, see the [AWS CloudTrail User Guide](#).

If your policy does not grant permission to create or modify a specific resource, the console displays an encoded message with diagnostic information. You can decode the message using

the [DecodeAuthorizationMessage](#) API action for AWS STS, or the [decode-authorization-message](#) command in the AWS CLI.

Examples

- [Example: Read-only access \(p. 966\)](#)
- [Example: Using the EC2 launch wizard \(p. 967\)](#)
- [Example: Working with volumes \(p. 970\)](#)
- [Example: Working with security groups \(p. 970\)](#)
- [Example: Working with Elastic IP addresses \(p. 972\)](#)
- [Example: Working with Reserved Instances \(p. 973\)](#)

For additional information about creating policies for the Amazon EC2 console, see the following AWS Security Blog post: [Granting Users Permission to Work in the Amazon EC2 Console](#).

Example: Read-only access

To allow users to view all resources in the Amazon EC2 console, you can use the same policy as the following example: [Example: Read-only access \(p. 929\)](#). Users cannot perform any actions on those resources or create new resources, unless another statement grants them permission to do so.

View instances, AMIs, and snapshots

Alternatively, you can provide read-only access to a subset of resources. To do this, replace the * wildcard in the `ec2:Describe` API action with specific `ec2:Describe` actions for each resource. The following policy allows users to view all instances, AMIs, and snapshots in the Amazon EC2 console. The `ec2:DescribeTags` action allows users to view public AMIs. The console requires the tagging information to display public AMIs; however, you can remove this action to allow users to view only private AMIs.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": [  
            "ec2:DescribeInstances",  
            "ec2:DescribeImages",  
            "ec2:DescribeTags",  
            "ec2:DescribeSnapshots"  
        ],  
        "Resource": "*"  
    }  
}]
```

Note

The Amazon EC2 `ec2:Describe*` API actions do not support resource-level permissions, so you cannot control which individual resources users can view in the console. Therefore, the * wildcard is necessary in the Resource element of the above statement. For more information about which ARNs you can use with which Amazon EC2 API actions, see [Actions, Resources, and Condition Keys for Amazon EC2](#) in the *IAM User Guide*.

View instances and CloudWatch metrics

The following policy allows users to view instances in the Amazon EC2 console, as well as CloudWatch alarms and metrics in the **Monitoring** tab of the **Instances** page. The Amazon EC2 console uses the CloudWatch API to display the alarms and metrics, so you must grant users permission to use the `cloudwatch:DescribeAlarms` and `cloudwatch:GetMetricStatistics` actions.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeInstances",  
                "cloudwatch:DescribeAlarms",  
                "cloudwatch:GetMetricStatistics"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Example: Using the EC2 launch wizard

The Amazon EC2 launch wizard is a series of screens with options to configure and launch an instance. Your policy must include permission to use the API actions that allow users to work with the wizard's options. If your policy does not include permission to use those actions, some items in the wizard cannot load properly, and users cannot complete a launch.

Basic launch wizard access

To complete a launch successfully, users must be given permission to use the `ec2:RunInstances` API action, and at least the following API actions:

- `ec2:DescribeImages`: To view and select an AMI.
- `ec2:DescribeVpcs`: To view the available network options.
- `ec2:DescribeSubnets`: To view all available subnets for the chosen VPC.
- `ec2:DescribeSecurityGroups` or `ec2>CreateSecurityGroup`: To view and select an existing security group, or to create a new one.
- `ec2:DescribeKeyPairs` or `ec2>CreateKeyPair`: To select an existing key pair, or to create a new one.
- `ec2:AuthorizeSecurityGroupIngress`: To add inbound rules.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeInstances",  
                "ec2:DescribeImages",  
                "ec2:DescribeKeyPairs",  
                "ec2:DescribeVpcs",  
                "ec2:DescribeSubnets",  
                "ec2:DescribeSecurityGroups",  
                "ec2:CreateSecurityGroup",  
                "ec2:AuthorizeSecurityGroupIngress",  
                "ec2>CreateKeyPair"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": "*"  
        }  
    ]  
}
```

}

You can add API actions to your policy to provide more options for users, for example:

- `ec2:DescribeAvailabilityZones`: To view and select a specific Availability Zone.
- `ec2:DescribeNetworkInterfaces`: To view and select existing network interfaces for the selected subnet.
- To add outbound rules to VPC security groups, users must be granted permission to use the `ec2:AuthorizeSecurityGroupEgress` API action. To modify or delete existing rules, users must be granted permission to use the relevant `ec2:RevokeSecurityGroup*` API action.
- `ec2:CreateTags`: To tag the resources that are created by `RunInstances`. For more information, see [Granting permission to tag resources during creation \(p. 926\)](#). If users do not have permission to use this action and they attempt to apply tags on the tagging page of the launch wizard, the launch fails.

Important

Be careful about granting users permission to use the `ec2:CreateTags` action, because doing so limits your ability to use the `ec2:ResourceTag` condition key to restrict their use of other resources. If you grant users permission to use the `ec2:CreateTags` action, they can change a resource's tag in order to bypass those restrictions. For more information, see [Controlling access to EC2 resources using resource tags \(p. 928\)](#).

- To use Systems Manager parameters when selecting an AMI, you must add `ssm:DescribeParameters` and `ssm:GetParameters` to your policy. `ssm:DescribeParameters` grants your IAM users the permission to view and select Systems Manager parameters. `ssm:GetParameters` grants your IAM users the permission to get the values of the Systems Manager parameters. You can also restrict access to specific Systems Manager parameters. For more information, see [Restrict access to specific Systems Manager parameters](#) later in this section.

Currently, the Amazon EC2 `Describe*` API actions do not support resource-level permissions, so you cannot restrict which individual resources users can view in the launch wizard. However, you can apply resource-level permissions on the `ec2:RunInstances` API action to restrict which resources users can use to launch an instance. The launch fails if users select options that they are not authorized to use.

Restrict access to a specific instance type, subnet, and Region

The following policy allows users to launch `t2.micro` instances using AMIs owned by Amazon, and only into a specific subnet (`subnet-1a2b3c4d`). Users can only launch in the `sa-east-1` Region. If users select a different Region, or select a different instance type, AMI, or subnet in the launch wizard, the launch fails.

The first statement grants users permission to view the options in the launch wizard or to create new ones, as explained in the example above. The second statement grants users permission to use the network interface, volume, key pair, security group, and subnet resources for the `ec2:RunInstances` action, which are required to launch an instance into a VPC. For more information about using the `ec2:RunInstances` action, see [Launching instances \(`RunInstances`\) \(p. 941\)](#). The third and fourth statements grant users permission to use the instance and AMI resources respectively, but only if the instance is a `t2.micro` instance, and only if the AMI is owned by Amazon.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeInstances",  
                "ec2:DescribeImages",  
                "ec2:DescribeKeyPairs",  
                "ec2:CreateKeyPair",  
                "ec2:DescribeVpcs",  
                "ec2:DescribeSubnets",  
                "ec2:RunInstances"  
            ],  
            "Resource": "arn:aws:ec2:sa-east-1:123456789012:instance/*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeNetworkInterfaces",  
                "ec2:DescribeVolumes",  
                "ec2:DescribeSecurityGroups",  
                "ec2:DescribeSubnets",  
                "ec2:DescribeInstances",  
                "ec2:RunInstances"  
            ],  
            "Resource": "arn:aws:ec2:sa-east-1:123456789012:subnet-1a2b3c4d"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeInstances",  
                "ec2:RunInstances"  
            ],  
            "Resource": "arn:aws:ec2:sa-east-1:123456789012:ami-01234567890abcdef"  
        }  
    ]  
}
```

```

        "ec2:DescribeSecurityGroups",
        "ec2>CreateSecurityGroup",
        "ec2:AuthorizeSecurityGroupIngress"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
        "arn:aws:ec2:sa-east-1:111122223333:network-interface/*",
        "arn:aws:ec2:sa-east-1:111122223333:volume/*",
        "arn:aws:ec2:sa-east-1:111122223333:key-pair/*",
        "arn:aws:ec2:sa-east-1:111122223333:security-group/*",
        "arn:aws:ec2:sa-east-1:111122223333:subnet/subnet-1a2b3c4d"
    ]
},
{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
        "arn:aws:ec2:sa-east-1:111122223333:instance/*"
    ],
    "Condition": {
        "StringEquals": {
            "ec2:InstanceType": "t2.micro"
        }
    }
},
{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
        "arn:aws:ec2:sa-east-1::image/ami-*"
    ],
    "Condition": {
        "StringEquals": {
            "ec2:Owner": "amazon"
        }
    }
}
]
}

```

Restrict access to specific Systems Manager parameters

The following policy grants access to use Systems Manager parameters with a specific name.

The first statement grants users the permission to view Systems Manager parameters when selecting an AMI in the launch wizard. The second statement grants users the permission to only use parameters that are named prod-*.

```

{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ssm:DescribeParameters"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [

```

```
    "ssm:GetParameters"
],
"Resource": "arn:aws:ssm:us-east-2:123456123:parameter/prod-*"
}
]
}
```

Example: Working with volumes

The following policy grants users permission to view and create volumes, and attach and detach volumes to specific instances.

Users can attach any volume to instances that have the tag "purpose=test", and also detach volumes from those instances. To attach a volume using the Amazon EC2 console, it is helpful for users to have permission to use the ec2:DescribeInstances action, as this allows them to select an instance from a pre-populated list in the **Attach Volume** dialog box. However, this also allows users to view all instances on the **Instances** page in the console, so you can omit this action.

In the first statement, the ec2:DescribeAvailabilityZones action is necessary to ensure that a user can select an Availability Zone when creating a volume.

Users cannot tag the volumes that they create (either during or after volume creation).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVolumes",
        "ec2:DescribeAvailabilityZones",
        "ec2:CreateVolume",
        "ec2:DescribeInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:region:111122223333:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/purpose": "test"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:region:111122223333:volume/*"
    }
  ]
}
```

Example: Working with security groups

View security groups and add and remove rules

The following policy grants users permission to view security groups in the Amazon EC2 console, to add and remove inbound and outbound rules, and to modify rule descriptions for existing security groups that have the tag `Department=Test`.

In the first statement, the `ec2:DescribeTags` action allows users to view tags in the console, which makes it easier for users to identify the security groups that they are allowed to modify.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeSecurityGroups",  
                "ec2:DescribeTags"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:AuthorizeSecurityGroupIngress",  
                "ec2:RevokeSecurityGroupIngress",  
                "ec2:AuthorizeSecurityGroupEgress",  
                "ec2:RevokeSecurityGroupEgress",  
                "ec2:UpdateSecurityGroupRuleDescriptionsIngress",  
                "ec2:UpdateSecurityGroupRuleDescriptionsEgress"  
            ],  
            "Resource": [  
                "arn:aws:ec2:region:111122223333:security-group/*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/Department": "Test"  
                }  
            }  
        }  
    ]  
}
```

Working with the Create Security Group dialog box

You can create a policy that allows users to work with the **Create Security Group** dialog box in the Amazon EC2 console. To use this dialog box, users must be granted permission to use at least the following API actions:

- `ec2:CreateSecurityGroup`: To create a new security group.
- `ec2:DescribeVpcs`: To view a list of existing VPCs in the **VPC** list.

With these permissions, users can create a new security group successfully, but they cannot add any rules to it. To work with rules in the **Create Security Group** dialog box, you can add the following API actions to your policy:

- `ec2:AuthorizeSecurityGroupIngress`: To add inbound rules.
- `ec2:AuthorizeSecurityGroupEgress`: To add outbound rules to VPC security groups.
- `ec2:RevokeSecurityGroupIngress`: To modify or delete existing inbound rules. This is useful to allow users to use the **Copy to new** feature in the console. This feature opens the **Create Security Group** dialog box and populates it with the same rules as the security group that was selected.
- `ec2:RevokeSecurityGroupEgress`: To modify or delete outbound rules for VPC security groups. This is useful to allow users to modify or delete the default outbound rule that allows all outbound traffic.

- `ec2:DeleteSecurityGroup`: To cater for when invalid rules cannot be saved. The console first creates the security group, and then adds the specified rules. If the rules are invalid, the action fails, and the console attempts to delete the security group. The user remains in the **Create Security Group** dialog box so that they can correct the invalid rule and try to create the security group again. This API action is not required, but if a user is not granted permission to use it and attempts to create a security group with invalid rules, the security group is created without any rules, and the user must add them afterward.
- `ec2:UpdateSecurityGroupRuleDescriptionsIngress`: To add or update descriptions of ingress (inbound) security group rules.
- `ec2:UpdateSecurityGroupRuleDescriptionsEgress`: To add or update descriptions of egress (outbound) security group rules.

Currently, the `ec2:CreateSecurityGroup` API action does not support resource-level permissions; however, you can apply resource-level permissions to the `ec2:AuthorizeSecurityGroupIngress` and `ec2:AuthorizeSecurityGroupEgress` actions to control how users can create rules.

The following policy grants users permission to use the **Create Security Group** dialog box, and to create inbound and outbound rules for security groups that are associated with a specific VPC (`vpc-1a2b3c4d`). Users can create security groups for EC2-Classic or another VPC, but they cannot add any rules to them. Similarly, users cannot add any rules to any existing security group that's not associated with VPC `vpc-1a2b3c4d`. Users are also granted permission to view all security groups in the console. This makes it easier for users to identify the security groups to which they can add inbound rules. This policy also grants users permission to delete security groups that are associated with VPC `vpc-1a2b3c4d`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeSecurityGroups",  
                "ec2:CreateSecurityGroup",  
                "ec2:DescribeVpcs"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DeleteSecurityGroup",  
                "ec2:AuthorizeSecurityGroupIngress",  
                "ec2:AuthorizeSecurityGroupEgress"  
            ],  
            "Resource": "arn:aws:ec2:region:111122223333:security-group/*",  
            "Condition": {  
                "ArnEquals": {  
                    "ec2:Vpc": "arn:aws:ec2:region:111122223333:vpc/vpc-1a2b3c4d"  
                }  
            }  
        }  
    ]  
}
```

Example: Working with Elastic IP addresses

To allow users to view Elastic IP addresses in the Amazon EC2 console, you must grant users permission to use the `ec2:DescribeAddresses` action.

To allow users to work with Elastic IP addresses, you can add the following actions to your policy.

- `ec2:AllocateAddress`: To allocate an Elastic IP address.

- `ec2:ReleaseAddress`: To release an Elastic IP address.
- `ec2:AssociateAddress`: To associate an Elastic IP address with an instance or a network interface.
- `ec2:DescribeNetworkInterfaces` and `ec2:DescribeInstances`: To work with the **Associate address** screen. The screen displays the available instances or network interfaces to which you can associate an Elastic IP address.
- `ec2:DisassociateAddress`: To disassociate an Elastic IP address from an instance or a network interface.

The following policy allows users to view, allocate, and associate Elastic IP addresses with instances. Users cannot associate Elastic IP addresses with network interfaces, disassociate Elastic IP addresses, or release them.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeAddresses",  
                "ec2:AllocateAddress",  
                "ec2:DescribeInstances",  
                "ec2:AssociateAddress"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Example: Working with Reserved Instances

The following policy can be attached to an IAM user. It gives the user access to view and modify Reserved Instances in your account, as well as purchase new Reserved Instances in the AWS Management Console.

This policy allows users to view all the Reserved Instances, as well as On-Demand Instances, in the account. It's not possible to set resource-level permissions for individual Reserved Instances.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeReservedInstances",  
                "ec2:ModifyReservedInstances",  
                "ec2:PurchaseReservedInstancesOffering",  
                "ec2:DescribeInstances",  
                "ec2:DescribeInstanceTypes",  
                "ec2:DescribeAvailabilityZones",  
                "ec2:DescribeReservedInstancesOfferings"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

The `ec2:DescribeAvailabilityZones` action is necessary to ensure that the Amazon EC2 console can display information about the Availability Zones in which you can purchase Reserved Instances. The `ec2:DescribeInstances` action is not required, but ensures that the user can view the instances in the account and purchase reservations to match the correct specifications.

You can adjust the API actions to limit user access, for example removing `ec2:DescribeInstances` and `ec2:DescribeAvailabilityZones` means the user has read-only access.

IAM roles for Amazon EC2

Applications must sign their API requests with AWS credentials. Therefore, if you are an application developer, you need a strategy for managing credentials for your applications that run on EC2 instances. For example, you can securely distribute your AWS credentials to the instances, enabling the applications on those instances to use your credentials to sign requests, while protecting your credentials from other users. However, it's challenging to securely distribute credentials to each instance, especially those that AWS creates on your behalf, such as Spot Instances or instances in Auto Scaling groups. You must also be able to update the credentials on each instance when you rotate your AWS credentials.

We designed IAM roles so that your applications can securely make API requests from your instances, without requiring you to manage the security credentials that the applications use. Instead of creating and distributing your AWS credentials, you can delegate permission to make API requests using IAM roles as follows:

1. Create an IAM role.
2. Define which accounts or AWS services can assume the role.
3. Define which API actions and resources the application can use after assuming the role.
4. Specify the role when you launch your instance, or attach the role to an existing instance.
5. Have the application retrieve a set of temporary credentials and use them.

For example, you can use IAM roles to grant permissions to applications running on your instances that need to use a bucket in Amazon S3. You can specify permissions for IAM roles by creating a policy in JSON format. These are similar to the policies that you create for IAM users. If you change a role, the change is propagated to all instances.

When creating IAM roles, associate least privilege IAM policies that restrict access to the specific API calls the application requires.

You cannot attach multiple IAM roles to a single instance, but you can attach a single IAM role to multiple instances. For more information about creating and using IAM roles, see [Roles](#) in the *IAM User Guide*.

You can apply resource-level permissions to your IAM policies to control the users' ability to attach, replace, or detach IAM roles for an instance. For more information, see [Supported resource-level permissions for Amazon EC2 API actions \(p. 923\)](#) and the following example: [Example: Working with IAM roles \(p. 960\)](#).

Contents

- [Instance profiles \(p. 974\)](#)
- [Retrieving security credentials from instance metadata \(p. 975\)](#)
- [Granting an IAM user permission to pass an IAM role to an instance \(p. 976\)](#)
- [Working with IAM roles \(p. 976\)](#)

Instance profiles

Amazon EC2 uses an *instance profile* as a container for an IAM role. When you create an IAM role using the IAM console, the console creates an instance profile automatically and gives it the same name as the role to which it corresponds. If you use the Amazon EC2 console to launch an instance with an IAM role or to attach an IAM role to an instance, you choose the role based on a list of instance profile names.

If you use the AWS CLI, API, or an AWS SDK to create a role, you create the role and instance profile as separate actions, with potentially different names. If you then use the AWS CLI, API, or an AWS SDK to launch an instance with an IAM role or to attach an IAM role to an instance, specify the instance profile name.

An instance profile can contain only one IAM role. This limit cannot be increased.

For more information, see [Instance Profiles](#) in the *IAM User Guide*.

Retrieving security credentials from instance metadata

An application on the instance retrieves the security credentials provided by the role from the instance metadata item `iam/security-credentials/role-name`. The application is granted the permissions for the actions and resources that you've defined for the role through the security credentials associated with the role. These security credentials are temporary and we rotate them automatically. We make new credentials available at least five minutes before the expiration of the old credentials.

Warning

If you use services that use instance metadata with IAM roles, ensure that you don't expose your credentials when the services make HTTP calls on your behalf. The types of services that could expose your credentials include HTTP proxies, HTML/CSS validator services, and XML processors that support XML inclusion.

The following command retrieves the security credentials for an IAM role named `s3access`.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/iam/security-credentials/s3access
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/s3access
```

The following is example output.

```
{
  "Code" : "Success",
  "LastUpdated" : "2012-04-26T16:39:16Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIAIOSFODNN7EXAMPLE",
  "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
  "Token" : "token",
  "Expiration" : "2017-05-17T15:09:54Z"
}
```

For applications, AWS CLI, and Tools for Windows PowerShell commands that run on the instance, you do not have to explicitly get the temporary security credentials—the AWS SDKs, AWS CLI, and Tools for Windows PowerShell automatically get the credentials from the EC2 instance metadata service and use them. To make a call outside of the instance using temporary security credentials (for example, to test IAM policies), you must provide the access key, secret key, and the session token. For more information, see [Using Temporary Security Credentials to Request Access to AWS Resources](#) in the *IAM User Guide*.

For more information about instance metadata, see [Instance metadata and user data \(p. 658\)](#). For information about the instance metadata IP address, see [Retrieving instance metadata \(p. 665\)](#).

Granting an IAM user permission to pass an IAM role to an instance

To enable an IAM user to launch an instance with an IAM role or to attach or replace an IAM role for an existing instance, you must grant the user permission to pass the role to the instance.

The following IAM policy grants users permission to launch instances (`ec2:RunInstances`) with an IAM role, or to attach or replace an IAM role for an existing instance (`ec2:AssociateIamInstanceProfile` and `ec2:ReplaceIamInstanceProfileAssociation`).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances",  
                "ec2:AssociateIamInstanceProfile",  
                "ec2:ReplaceIamInstanceProfileAssociation"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "iam:PassRole",  
            "Resource": "*"  
        }  
    ]  
}
```

This policy grants IAM users access to all your roles by specifying the resource as "*" in the policy. However, consider whether users who launch instances with your roles (ones that exist or that you create later on) might be granted permissions that they don't need or shouldn't have.

Working with IAM roles

You can create an IAM role and attach it to an instance during or after launch. You can also replace or detach an IAM role for an instance.

Contents

- [Creating an IAM role \(p. 976\)](#)
- [Launching an instance with an IAM role \(p. 978\)](#)
- [Attaching an IAM role to an instance \(p. 979\)](#)
- [Replacing an IAM role \(p. 980\)](#)
- [Detaching an IAM role \(p. 981\)](#)

Creating an IAM role

You must create an IAM role before you can launch an instance with that role or attach it to an instance.

To create an IAM role using the IAM console

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**, **Create role**.

3. On the **Select role type** page, choose **EC2** and the **EC2** use case. Choose **Next: Permissions**.
4. On the **Attach permissions policy** page, select an AWS managed policy that grants your instances access to the resources that they need.
5. On the **Review** page, enter a name for the role and choose **Create role**.

Alternatively, you can use the AWS CLI to create an IAM role. The following example creates an IAM role with a policy that allows the role to use an Amazon S3 bucket.

To create an IAM role and instance profile (AWS CLI)

1. Create the following trust policy and save it in a text file named `ec2-role-trust-policy.json`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": { "Service": "ec2.amazonaws.com"},  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

2. Create the `s3access` role and specify the trust policy that you created using the [create-role](#) command.

```
aws iam create-role --role-name s3access --assume-role-policy-document file://ec2-role-trust-policy.json  
{  
    "Role": {  
        "AssumeRolePolicyDocument": {  
            "Version": "2012-10-17",  
            "Statement": [  
                {  
                    "Action": "sts:AssumeRole",  
                    "Effect": "Allow",  
                    "Principal": {  
                        "Service": "ec2.amazonaws.com"  
                    }  
                }  
            ]  
        },  
        "RoleId": "AROAIIZKPBKS2LEXAMPLE",  
        "CreateDate": "2013-12-12T23:46:37.247Z",  
        "RoleName": "s3access",  
        "Path": "/",  
        "Arn": "arn:aws:iam::123456789012:role/s3access"  
    }  
}
```

3. Create an access policy and save it in a text file named `ec2-role-access-policy.json`. For example, this policy grants administrative permissions for Amazon S3 to applications running on the instance.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": ["s3:*"],  
            "Resource": ["*"]  
        }  
    ]  
}
```

```
        ]  
    }
```

4. Attach the access policy to the role using the [put-role-policy](#) command.

```
aws iam put-role-policy --role-name s3access --policy-name S3-Permissions --policy-document file://ec2-role-access-policy.json
```

5. Create an instance profile named `s3access-profile` using the [create-instance-profile](#) command.

```
aws iam create-instance-profile --instance-profile-name s3access-profile  
{  
    "InstanceProfile": {  
        "InstanceProfileId": "AIPAJTLBPJLEGREXAMPLE",  
        "Roles": [],  
        "CreateDate": "2013-12-12T23:53:34.093Z",  
        "InstanceProfileName": "s3access-profile",  
        "Path": "/",  
        "Arn": "arn:aws:iam::123456789012:instance-profile/s3access-profile"  
    }  
}
```

6. Add the `s3access` role to the `s3access-profile` instance profile.

```
aws iam add-role-to-instance-profile --instance-profile-name s3access-profile --role-name s3access
```

Alternatively, you can use the following AWS Tools for Windows PowerShell commands:

- [New-IAMRole](#)
- [Register-IAMRolePolicy](#)
- [New-IAMInstanceProfile](#)

Launching an instance with an IAM role

After you've created an IAM role, you can launch an instance, and associate that role with the instance during launch.

Important

After you create an IAM role, it may take several seconds for the permissions to propagate. If your first attempt to launch an instance with a role fails, wait a few seconds before trying again. For more information, see [Troubleshooting Working with Roles](#) in the *IAM User Guide*.

To launch an instance with an IAM role (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch instance**.
3. Select an AMI and instance type and then choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, for **IAM role**, select the IAM role that you created.

Note

The **IAM role** list displays the name of the instance profile that you created when you created your IAM role. If you created your IAM role using the console, the instance profile was created for you and given the same name as the role. If you created your IAM role using the AWS CLI, API, or an AWS SDK, you may have named your instance profile differently.

5. Configure any other details, then follow the instructions through the rest of the wizard, or choose **Review and Launch** to accept default settings and go directly to the **Review Instance Launch** page.
6. Review your settings, then choose **Launch** to choose a key pair and launch your instance.
7. If you are using the Amazon EC2 API actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. The AWS SDK does this for you.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

Alternatively, you can use the AWS CLI to associate a role with an instance during launch. You must specify the instance profile in the command.

To launch an instance with an IAM role (AWS CLI)

1. Use the [run-instances](#) command to launch an instance using the instance profile. The following example shows how to launch an instance with the instance profile.

```
aws ec2 run-instances \
--image-id ami-11aa22bb \
--iam-instance-profile Name="s3access-profile" \
--key-name my-key-pair \
--security-groups my-security-group \
--subnet-id subnet-1a2b3c4d
```

Alternatively, use the [New-EC2Instance](#) Tools for Windows PowerShell command.

2. If you are using the Amazon EC2 API actions in your application, retrieve the AWS security credentials made available on the instance and use them to sign the requests. The AWS SDK does this for you.

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

Attaching an IAM role to an instance

To attach an IAM role to an instance that has no role, the instance can be in the stopped or running state.

New console

To attach an IAM role to an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions**, **Instance settings**, **Modify IAM role**.

4. Select the IAM role to attach to your instance, and choose **Save**.

Old console

To attach an IAM role to an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions, Instance Settings, Attach/Replace IAM role**.
4. Select the IAM role to attach to your instance, and choose **Apply**.

To attach an IAM role to an instance (AWS CLI)

1. If required, describe your instances to get the ID of the instance to which to attach the role.

```
aws ec2 describe-instances
```

2. Use the **associate-iam-instance-profile** command to attach the IAM role to the instance by specifying the instance profile. You can use the Amazon Resource Name (ARN) of the instance profile, or you can use its name.

```
aws ec2 associate-iam-instance-profile \
    --instance-id i-1234567890abcdef0 \
    --iam-instance-profile Name="TestRole-1"\n\n{\n    "IamInstanceProfileAssociation": {\n        "InstanceId": "i-1234567890abcdef0",\n        "State": "associating",\n        "AssociationId": "iip-assoc-0dbd8529a48294120",\n        "IamInstanceProfile": {\n            "Id": "AIPAJLNLDX3AMYZNWYYAY",\n            "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-1"\n        }\n    }\n}
```

Alternatively, use the following Tools for Windows PowerShell commands:

- [Get-EC2Instance](#)
- [Register-EC2IamInstanceProfile](#)

Replacing an IAM role

To replace the IAM role on an instance that already has an attached IAM role, the instance must be in the running state. You can do this if you want to change the IAM role for an instance without detaching the existing one first. For example, you can do this to ensure that API actions performed by applications running on the instance are not interrupted.

New console

To replace an IAM role for an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.

3. Select the instance, choose **Actions**, **Instance settings**, **Modify IAM role**.
4. Select the IAM role to attach to your instance, and choose **Save**.

Old console

To replace an IAM role for an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions**, **Instance Settings**, **Attach/Replace IAM role**.
4. Select the IAM role to attach to your instance, and choose **Apply**.

To replace an IAM role for an instance (AWS CLI)

1. If required, describe your IAM instance profile associations to get the association ID for the IAM instance profile to replace.

```
aws ec2 describe-iam-instance-profile-associations
```

2. Use the [replace-iam-instance-profile-association](#) command to replace the IAM instance profile by specifying the association ID for the existing instance profile and the ARN or name of the instance profile that should replace it.

```
aws ec2 replace-iam-instance-profile-association \
--association-id iip-assoc-0044d817db6c0a4ba \
--iam-instance-profile Name="TestRole-2"  
  
{  
    "IamInstanceProfileAssociation": {  
        "InstanceId": "i-087711ddaf98f9489",  
        "State": "associating",  
        "AssociationId": "iip-assoc-09654be48e33b91e0",  
        "IamInstanceProfile": {  
            "Id": "AIPAJCJEDKX7QYHWYK7GS",  
            "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-2"  
        }  
    }  
}
```

Alternatively, use the following Tools for Windows PowerShell commands:

- [Get-EC2IamInstanceProfileAssociation](#)
- [Set-EC2IamInstanceProfileAssociation](#)

Detaching an IAM role

You can detach an IAM role from a running or stopped instance.

New console

To detach an IAM role from an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.

3. Select the instance, choose **Actions**, **Instance settings**, **Modify IAM role**.
4. For **IAM role**, choose **No IAM Role**. Choose **Save**.
5. In the confirmation dialog box, enter **Detach**, and then choose **Detach**.

Old console

To detach an IAM role from an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions**, **Instance Settings**, **Attach/Replace IAM role**.
4. For **IAM role**, choose **No Role**. Choose **Apply**.
5. In the confirmation dialog box, choose **Yes, Detach**.

To detach an IAM role from an instance (AWS CLI)

1. If required, use [describe-iam-instance-profile-associations](#) to describe your IAM instance profile associations and get the association ID for the IAM instance profile to detach.

```
aws ec2 describe-iam-instance-profile-associations

{
    "IamInstanceProfileAssociations": [
        {
            "InstanceId": "i-088ce778fbfeb4361",
            "State": "associated",
            "AssociationId": "iip-assoc-0044d817db6c0a4ba",
            "IamInstanceProfile": {
                "Id": "AIPAJEDNCAA64SSD265D6",
                "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-2"
            }
        }
    ]
}
```

2. Use the [disassociate-iam-instance-profile](#) command to detach the IAM instance profile using its association ID.

```
aws ec2 disassociate-iam-instance-profile --association-id iip-assoc-0044d817db6c0a4ba

{
    "IamInstanceProfileAssociation": {
        "InstanceId": "i-087711ddaf98f9489",
        "State": "disassociating",
        "AssociationId": "iip-assoc-0044d817db6c0a4ba",
        "IamInstanceProfile": {
            "Id": "AIPAJEDNCAA64SSD265D6",
            "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-2"
        }
    }
}
```

Alternatively, use the following Tools for Windows PowerShell commands:

- [Get-EC2IamInstanceProfileAssociation](#)
- [Unregister-EC2IamInstanceProfile](#)

Authorizing inbound traffic for your Linux instances

Security groups enable you to control traffic to your instance, including the kind of traffic that can reach your instance. For example, you can allow computers from only your home network to access your instance using SSH. If your instance is a web server, you can allow all IP addresses to access your instance using HTTP or HTTPS, so that external users can browse the content on your web server.

Your default security groups and newly created security groups include default rules that do not enable you to access your instance from the internet. For more information, see [Default security groups \(p. 1003\)](#) and [Custom security groups \(p. 1003\)](#). To enable network access to your instance, you must allow inbound traffic to your instance. To open a port for inbound traffic, add a rule to a security group that you associated with your instance when you launched it.

To connect to your instance, you must set up a rule to authorize SSH traffic from your computer's public IPv4 address. To allow SSH traffic from additional IP address ranges, add another rule for each range you need to authorize.

If you've enabled your VPC for IPv6 and launched your instance with an IPv6 address, you can connect to your instance using its IPv6 address instead of a public IPv4 address. Your local computer must have an IPv6 address and must be configured to use IPv6.

If you need to enable network access to a Windows instance, see [Authorizing inbound traffic for your Windows instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Before you start

Decide who requires access to your instance; for example, a single host or a specific network that you trust such as your local computer's public IPv4 address. The security group editor in the Amazon EC2 console can automatically detect the public IPv4 address of your local computer for you. Alternatively, you can use the search phrase "what is my IP address" in an internet browser, or use the following service: [Check IP](#). If you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

Warning

If you use `0.0.0.0/0`, you enable all IPv4 addresses to access your instance using SSH. If you use `::/0`, you enable all IPv6 address to access your instance. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, you authorize only a specific IP address or range of addresses to access your instance.

Decide whether you'll support SSH access to your instances using EC2 Instance Connect. If you will not use EC2 Instance Connect, consider uninstalling it or denying the following action in your IAM policies: `ec2-instance-connect:SendSSHPublicKey`. For more information, see [Uninstall EC2 Instance Connect \(p. 577\)](#) and [Configure IAM Permissions for EC2 Instance Connect \(p. 573\)](#).

Adding a rule for inbound SSH traffic to a Linux instance

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group that enable you to connect to your Linux instance from your IP address using SSH.

To add a rule to a security group for inbound SSH traffic over IPv4 (console)

1. In the navigation pane of the Amazon EC2 console, choose **Instances**. Select your instance and look at the **Description** tab; **Security groups** lists the security groups that are associated with the instance. Choose **view inbound rules** to display a list of the rules that are in effect for the instance.
2. In the navigation pane, choose **Security Groups**. Select one of the security groups associated with your instance.

3. In the details pane, on the **Inbound** tab, choose **Edit**. In the dialog, choose **Add Rule**, and then choose **SSH** from the **Type** list.
4. In the **Source** field, choose **My IP** to automatically populate the field with the public IPv4 address of your local computer. Alternatively, choose **Custom** and specify the public IPv4 address of your computer or network in CIDR notation. For example, if your IPv4 address is 203.0.113.25, specify 203.0.113.25/32 to list this single IPv4 address in CIDR notation. If your company allocates addresses from a range, specify the entire range, such as 203.0.113.0/24.

For information about finding your IP address, see [Before you start \(p. 983\)](#).
5. Choose **Save**.

If you launched an instance with an IPv6 address and want to connect to your instance using its IPv6 address, you must add rules that allow inbound IPv6 traffic over SSH.

To add a rule to a security group for inbound SSH traffic over IPv6 (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**. Select the security group for your instance.
3. Choose **Inbound, Edit, Add Rule**.
4. For **Type**, choose **SSH**.
5. In the **Source** field, specify the IPv6 address of your computer in CIDR notation. For example, if your IPv6 address is 2001:db8:1234:1a00:9691:9503:25ad:1761, specify 2001:db8:1234:1a00:9691:9503:25ad:1761/128 to list the single IP address in CIDR notation. If your company allocates addresses from a range, specify the entire range, such as 2001:db8:1234:1a00::/64.
6. Choose **Save**.

Note

Be sure to run the following commands on your local system, not on the instance itself. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

To add a rule to a security group using the command line

1. Find the security group that is associated with your instance using one of the following commands:
 - [describe-instance-attribute](#) (AWS CLI)

```
aws ec2 describe-instance-attribute --instance-id instance_id --attribute groupSet
```

- [Get-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

```
PS C:\> (Get-EC2InstanceAttribute -InstanceId instance_id -Attribute groupSet).Groups
```

Both commands return a security group ID, which you use in the next step.

2. Add the rule to the security group using one of the following commands:
 - [authorize-security-group-ingress](#) (AWS CLI)

```
aws ec2 authorize-security-group-ingress --group-id security_group_id --protocol tcp  
--port 22 --cidr cidr_ip_range
```

- [Grant-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

The `Grant-EC2SecurityGroupIngress` command needs an `IpPermission` parameter, which describes the protocol, port range, and IP address range to be used for the security group rule. The following command creates the `IpPermission` parameter:

```
PS C:\> $ip1 = @{ IpProtocol="tcp"; FromPort="22"; ToPort="22";  
IpRanges="cidr_ip_range" }
```

```
PS C:\> Grant-EC2SecurityGroupIngress -GroupId security_group_id -IpPermission  
@($ip1)
```

Assigning a security group to an instance

You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group.

After you launch an instance, you can change its security groups. For more information, see [Changing an instance's security groups](#) in the *Amazon VPC User Guide*.

Amazon EC2 key pairs and Linux instances

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance. Amazon EC2 stores the public key, and you store the private key. You use the private key, instead of a password, to securely access your instances. Anyone who possesses your private keys can connect to your instances, so it's important that you store your private keys in a secure place.

When you launch an instance, you are [prompted for a key pair \(p. 504\)](#). If you plan to connect to the instance using SSH, you must specify a key pair. You can choose an existing key pair or create a new one. When your instance boots for the first time, the content of the public key that you specified at launch is placed on your Linux instance in an entry within `~/.ssh/authorized_keys`. When you connect to your Linux instance using SSH, to log in you must specify the private key that corresponds to the public key content. For more information about connecting to your instance, see [Connect to your Linux instance \(p. 563\)](#). For more information about key pairs and Windows instances, see [Amazon EC2 key pairs and Windows instances](#) in the *Amazon EC2 User Guide for Windows Instances*

Because Amazon EC2 doesn't keep a copy of your private key, there is no way to recover a private key if you lose it. However, there can still be a way to connect to instances for which you've lost the private key. For more information, see [Connecting to your Linux instance if you lose your private key \(p. 994\)](#).

The keys that Amazon EC2 uses are 2048-bit SSH-2 RSA keys. You can have up to 5,000 key pairs per Region.

Contents

- [Creating or importing a key pair \(p. 986\)](#)
- [Tagging a key pair \(p. 989\)](#)
- [Retrieving the public key for your key pair \(p. 991\)](#)
- [Retrieving the public key for your key pair through instance metadata \(p. 991\)](#)
- [Locating the public key on an instance \(p. 992\)](#)
- [Identifying the key pair that was specified at launch \(p. 992\)](#)
- [\(Optional\) Verifying your key pair's fingerprint \(p. 993\)](#)

- [Adding or replacing a key pair for your instance \(p. 993\)](#)
- [Connecting to your Linux instance if you lose your private key \(p. 994\)](#)
- [Deleting your key pair \(p. 998\)](#)

Creating or importing a key pair

You can use Amazon EC2 to create a new key pair, or you can import an existing key pair.

Options

- [Option 1: Create a key pair using Amazon EC2 \(p. 986\)](#)
- [Option 2: Import your own public key to Amazon EC2 \(p. 987\)](#)

Option 1: Create a key pair using Amazon EC2

You can create a key pair using one of the following methods.

New console

To create your key pair

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
3. Choose **Create key pair**.
4. For **Name**, enter a descriptive name for the key pair. Amazon EC2 associates the public key with the name that you specify as the key name. A key name can include up to 255 ASCII characters. It can't include leading or trailing spaces.
5. For **File format**, choose the format in which to save the private key. To save the private key in a format that can be used with OpenSSH, choose **pem**. To save the private key in a format that can be used with PuTTY, choose **ppk**.
6. Choose **Create key pair**.
7. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is determined by the file format you chose. Save the private key file in a safe place.

Important

This is the only chance for you to save the private key file.

8. If you will use an SSH client on a macOS or Linux computer to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.

```
chmod 400 my-key-pair.pem
```

If you do not set these permissions, then you cannot connect to your instance using this key pair. For more information, see [Error: Unprotected private key file \(p. 1252\)](#).

Old console

To create your key pair

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.

3. Choose **Create Key Pair**.
4. For **Key pair name**, enter a descriptive name for the key pair, and then choose **Create**. A key name can include up to 255 ASCII characters. It can't include leading or trailing spaces.
5. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is **.pem**. Save the private key file in a safe place.

Important

This is the only chance for you to save the private key file.

6. If you will use an SSH client on a macOS or Linux computer to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.

```
chmod 400 my-key-pair.pem
```

If you do not set these permissions, then you cannot connect to your instance using this key pair. For more information, see [Error: Unprotected private key file \(p. 1252\)](#).

AWS CLI

To create your key pair

1. Use the [create-key-pair](#) AWS CLI command as follows to generate the key and save it to a **.pem** file.

```
aws ec2 create-key-pair --key-name my-key-pair --query 'KeyMaterial' --output text
> my-key-pair.pem
```

2. If you will use an SSH client on a macOS or Linux computer to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.

```
chmod 400 my-key-pair.pem
```

If you do not set these permissions, then you cannot connect to your instance using this key pair. For more information, see [Error: Unprotected private key file \(p. 1252\)](#).

PowerShell

To create your key pair

Use the [New-EC2KeyPair](#) AWS Tools for Windows PowerShell command as follows to generate the key and save it to a **.pem** file.

```
PS C:\> (New-EC2KeyPair -KeyName "my-key-pair").KeyMaterial | Out-File -Encoding ascii
-FilePath C:\path\my-key-pair.pem
```

Option 2: Import your own public key to Amazon EC2

Instead of using Amazon EC2 to create your key pair, you can create an RSA key pair using a third-party tool and then import the public key to Amazon EC2. For example, you can use **ssh-keygen** (a tool provided with the standard OpenSSH installation) to create a key pair. Alternatively, Java, Ruby, Python, and many other programming languages provide standard libraries that you can use to create an RSA key pair.

Requirements

- The following formats are supported:
 - OpenSSH public key format (the format in `~/.ssh/authorized_keys`). If you connect using SSH while using the EC2 Instance Connect API, the SSH2 format is also supported.
 - Base64 encoded DER format
 - SSH public key file format as specified in [RFC4716](#)
 - SSH private key file format must be PEM (for example, use `ssh-keygen -m PEM` to convert the OpenSSH key into the PEM format)
- Create an RSA key. Amazon EC2 does not accept DSA keys.
- The supported lengths are 1024, 2048, and 4096. If you connect using SSH while using the EC2 Instance Connect API, the supported lengths are 2048 and 4096.

To create a key pair using a third-party tool

1. Generate a key pair with a third-party tool of your choice.
2. Save the public key to a local file. For example, `~/.ssh/my-key-pair.pub` (Linux) or `C:\keys\my-key-pair.pub` (Windows). The file name extension for this file is not important.
3. Save the private key to a different local file that has the `.pem` extension. For example, `~/.ssh/my-key-pair.pem` (Linux) or `C:\keys\my-key-pair.pem` (Windows). Save the private key file in a safe place. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

After you have created the key pair, use one of the following methods to import your key pair to Amazon EC2.

New console

To import the public key

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Key Pairs**.
3. Choose **Import key pair**.
4. For **Name**, enter a descriptive name for the key pair. The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.
5. Either choose **Browse** to navigate to and select your public key, or paste the contents of your public key into the **Public key contents** field.
6. Choose **Import key pair**.
7. Verify that the key pair you imported appears in the list of key pairs.

Old console

To import the public key

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
3. Choose **Import Key Pair**.
4. In the **Import Key Pair** dialog box, choose **Browse**, and select the public key file that you saved previously. Enter a name for the key pair in the **Key pair name** field, and choose **Import**. The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.
5. Verify that the key pair you imported appears in the list of key pairs.

AWS CLI

To import the public key

Use the [import-key-pair](#) AWS CLI command.

To verify that the key pair was imported successfully

Use the [describe-key-pairs](#) AWS CLI command.

PowerShell

To import the public key

Use the [Import-EC2KeyPair](#) AWS Tools for Windows PowerShell command.

To verify that the key pair was imported successfully

Use the [Get-EC2KeyPair](#) AWS Tools for Windows PowerShell command.

Tagging a key pair

To help categorize and manage your existing key pairs, you can tag them with custom metadata. For more information about how tags work, see [Tagging your Amazon EC2 resources \(p. 1229\)](#).

You can view, add, and delete tags using the new console and the command line tools.

New console

To view, add, or delete a tag for an existing key pair

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Key Pairs**.
3. Select a key pair, and then choose **Actions, Manage tags**.
4. The **Manage tags** section displays any tags that are assigned to the key pair.
 - To add a tag, choose **Add tag**, and then enter the tag key and value. You can add up to 50 tags per key pair. For more information, see [Tag restrictions \(p. 1233\)](#).
 - To delete a tag, choose **Remove** next to the tag that you want to delete.
5. Choose **Save changes**.

AWS CLI

To view key pair tags

Use the [describe-tags](#) AWS CLI command. In the following example, you describe the tags for all of your key pairs.

```
$ aws ec2 describe-tags --filters "Name=resource-type,Values=key-pair"
```

```
{
  "Tags": [
    {
      "Key": "Environment",
      "ResourceId": "key-0123456789EXAMPLE",
      "ResourceType": "key-pair",
      "Value": "Production"
    }
  ]
}
```

```
        },
        {
            "Key": "Environment",
            "ResourceId": "key-9876543210EXAMPLE",
            "ResourceType": "key-pair",
            "Value": "Production"
        }]
}
```

To describe the tags for a specific key pair

Use the [describe-key-pairs](#) AWS CLI command.

```
$ aws ec2 describe-key-pairs --key-pair-ids key-0123456789EXAMPLE
```

```
{
    "KeyPairs": [
        {
            "KeyName": "MyKeyPair",
            "KeyFingerprint": "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f",
            "KeyId": "key-0123456789EXAMPLE",
            "Tags": [
                {
                    "Key": "Environment",
                    "Value": "Production"
                }
            ]
        }
    ]
}
```

To tag an existing key pair

Use the [create-tags](#) AWS CLI command. In the following example, the existing key pair is tagged with Key=Cost-Center and Value=CC-123.

```
$ aws ec2 create-tags --resources key-0123456789EXAMPLE --tags Key=Cost-Center,Value=CC-123
```

To delete a tag from a key pair

Use the [delete-tags](#) AWS CLI command. For examples, see [Examples in the AWS CLI Command Reference](#).

PowerShell

To view key pair tags

Use the [Get-EC2Tag](#) command.

To describe the tags for a specific key pair

Use the [Get-EC2KeyPair](#) command.

To tag an existing key pair

Use the [New-EC2Tag](#) command.

To delete a tag from a key pair

Use the [Remove-EC2Tag](#) command.

Retrieving the public key for your key pair

On your local Linux or macOS computer, you can use the **ssh-keygen** command to retrieve the public key for your key pair. Specify the path where you downloaded your private key (the `.pem` file).

```
ssh-keygen -y -f /path_to_key_pair/my-key-pair.pem
```

The command returns the public key, as shown in the following example.

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBITntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

If the command fails, run the following command to ensure that you've changed the permissions on your key pair file so that only you can view it.

```
chmod 400 my-key-pair.pem
```

Retrieving the public key for your key pair through instance metadata

The public key that you specified when you launched an instance is also available to you through its instance metadata. To view the public key that you specified when launching the instance, use the following command from your instance:

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-
ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-
data/public-keys/0/openssh-key
```

The following is an example output.

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBITntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

The following is an example output.

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBITntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
```

```
BQoQzd8v7yeb70z1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

If you change the key pair that you use to connect to the instance, we don't update the instance metadata to show the new public key. Instead, the instance metadata continues to show the public key for the key pair that you specified when you launched the instance. For more information, see [Retrieving instance metadata \(p. 665\)](#).

Alternatively, on a Linux instance, the public key content is placed in an entry within `~/.ssh/authorized_keys`. You can open this file in an editor. The following is an example entry for the key pair named **my-key-pair**. It consists of the public key followed by the name of the key pair.

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCLKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706Vhz2ITxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXrlsLnBITntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzzqaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3RBQoQzd8v7yeb70z1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

Locating the public key on an instance

When you launch an instance, you are [prompted for a key pair \(p. 504\)](#). If you plan to connect to the instance using SSH, you must specify a key pair. You can choose an existing key pair or create a new one. When your instance boots for the first time, the content of the public key that you specified at launch is placed on your Linux instance in an entry within `~/.ssh/authorized_keys`.

To locate the public key on an instance

1. Connect to your instance. For more information, see [Connect to your Linux instance \(p. 563\)](#).
2. In the terminal window, open the `authorized_keys` file using your favorite text editor (such as **vim** or **nano**).

```
[ec2-user ~]$ nano ~/.ssh/authorized_keys
```

The `authorized_keys` file opens, displaying the public key, as shown in the following example.

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCLKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706Vhz2ITxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXrlsLnBITntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzzqaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3RBQoQzd8v7yeb70z1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

Identifying the key pair that was specified at launch

When you launch an instance, you are [prompted for a key pair \(p. 504\)](#). If you plan to connect to the instance using SSH, you must specify a key pair.

To identify the key pair that was specified at launch

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. On the **Description** tab, the **Key pair name** field displays the name of the key pair that you specified when you launched the instance. The value of the **Key pair name** does not change even if you change the public key on the instance, or add key pairs.

(Optional) Verifying your key pair's fingerprint

On the **Key Pairs** page in the Amazon EC2 console, the **Fingerprint** column displays the fingerprints generated from your key pairs. AWS calculates the fingerprint differently depending on whether the key pair was generated by AWS or a third-party tool. If you created the key pair using AWS, the fingerprint is calculated using an SHA-1 hash function. If you created the key pair with a third-party tool and uploaded the public key to AWS, or if you generated a new public key from an existing AWS-created private key and uploaded it to AWS, the fingerprint is calculated using an MD5 hash function.

You can use the SSH2 fingerprint that's displayed on the **Key Pairs** page to verify that the private key you have on your local machine matches the public key stored in AWS. From the computer where you downloaded the private key file, generate an SSH2 fingerprint from the private key file. The output should match the fingerprint that's displayed in the console.

If you created your key pair using AWS, you can use the OpenSSL tools to generate a fingerprint as shown in the following example.

```
$ openssl pkcs8 -in path_to_private_key -inform PEM -outform DER -topk8 -nocrypt | openssl sha1 -c
```

If you created a key pair using a third-party tool and uploaded the public key to AWS, you can use the OpenSSL tools to generate the fingerprint as shown in the following example.

```
$ openssl rsa -in path_to_private_key -pubout -outform DER | openssl md5 -c
```

If you created an OpenSSH key pair using OpenSSH 7.8 or later and uploaded the public key to AWS, you can use **ssh-keygen** to generate the fingerprint as shown in the following example.

```
$ ssh-keygen -ef path_to_private_key -m PEM | openssl rsa -RSAPublicKey_in -outform DER | openssl md5 -c
```

Adding or replacing a key pair for your instance

You can change the key pair that is used to access the default system account of your instance. For example, if a user in your organization requires access to the system user account using a separate key pair, you can add that key pair to your instance. Or, if someone has a copy of the .pem file and you want to prevent them from connecting to your instance (for example, if they've left your organization), you can replace the key pair with a new one.

To add or replace a key pair, you must be able to connect to your instance. If you've lost your existing private key or you launched your instance without a key pair, you won't be able to connect to your instance and therefore won't be able to add or replace a key pair. If you've lost your existing private key, you might be able to retrieve it. For more information, see [Connecting to your Linux instance if you lose your private key \(p. 994\)](#). If you launched your instance without a key pair, you won't be able to connect to the instance unless you chose an AMI that is configured to allow users another way to log in.

Note

These procedures are for modifying the key pair for the default user account, such as ec2-user. For more information about adding user accounts to your instance, see [Managing user accounts on your Amazon Linux instance \(p. 620\)](#).

To add or replace a key pair

1. Create a new key pair using the [Amazon EC2 console \(p. 986\)](#) or a [third-party tool \(p. 987\)](#).

2. Retrieve the public key from your new key pair. For more information, see [Retrieving the public key for your key pair \(p. 991\)](#).
3. Connect to your instance using your existing private key file.
4. Using a text editor of your choice, open the `.ssh/authorized_keys` file on the instance. Paste the public key information from your new key pair underneath the existing public key information. Save the file.
5. Disconnect from your instance, and test that you can connect to your instance using the new private key file.
6. (Optional) If you're replacing an existing key pair, connect to your instance and delete the public key information for the original key pair from the `.ssh/authorized_keys` file.

Note

If you're using an Auto Scaling group, ensure that the key pair you're replacing is not specified in your launch template or launch configuration. Amazon EC2 Auto Scaling launches a replacement instance if it detects an unhealthy instance; however, the instance launch fails if the key pair cannot be found.

Connecting to your Linux instance if you lose your private key

If you lose the private key for an EBS-backed instance, you can regain access to your instance. You must stop the instance, detach its root volume and attach it to another instance as a data volume, modify the `authorized_keys` file with a new public key, move the volume back to the original instance, and restart the instance. For more information about launching, connecting to, and stopping instances, see [Instance lifecycle \(p. 493\)](#).

This procedure is not supported for instances with instance-store backed root volumes. To determine the root device type of your instance, open the Amazon EC2 console, choose **Instances**, select the instance, and check the value of **Root device type** in the details pane. The value is either `ebs` or `instance store`. If the root device is an instance store volume, you cannot use this procedure to regain access to your instance; you must have the private key to connect to the instance.

Steps for connecting to an EBS-backed instance with a different key pair

- [Step 1: Create a new key pair \(p. 994\)](#)
- [Step 2: Get information about the original instance and its root volume \(p. 995\)](#)
- [Step 3: Stop the original instance \(p. 995\)](#)
- [Step 4: Launch a temporary instance \(p. 995\)](#)
- [Step 5: Detach the root volume from the original instance and attach it to the temporary instance \(p. 996\)](#)
- [Step 6: Add the new public key to `authorized_keys` on the original volume mounted to the temporary instance \(p. 996\)](#)
- [Step 7: Unmount and detach the original volume from the temporary instance, and reattach it to the original instance \(p. 998\)](#)
- [Step 8: Connect to the original instance using the new key pair \(p. 998\)](#)
- [Step 9: Clean up \(p. 998\)](#)

Step 1: Create a new key pair

Create a new key pair using either the Amazon EC2 console or a third-party tool. If you want to name your new key pair exactly the same as the lost private key, you must first delete the existing key pair. For

information about creating a new key pair, see [Option 1: Create a key pair using Amazon EC2 \(p. 986\)](#) or [Option 2: Import your own public key to Amazon EC2 \(p. 987\)](#).

Step 2: Get information about the original instance and its root volume

New console

To get information about your original instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Instances** in the navigation pane, and then select the instance that you'd like to connect to. (We'll refer to this as the *original* instance.)
3. On the **Details** tab, write down the instance ID and AMI ID.
4. On the **Networking** tab, write down the Availability Zone.
5. On the **Storage** tab, note the device name for the root volume in **Root device name** (for example, /dev/xvda). Find this device name under **Block devices** and write down the volume ID (for example, vol-0a1234b5678c910de).

Old console

To get information about your original instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Instances** in the navigation pane, and then select the instance that you'd like to connect to. (We'll refer to this as the *original* instance.)
3. From the **Description** tab, save the following information that you need to complete this procedure.
 - Write down the instance ID, AMI ID, and Availability Zone of the original instance.
 - In the **Root device** field, take note of the device name for the root volume (for example, /dev/sda1 or /dev/xvda). Choose the link and write down the volume ID in the **EBS ID** field (vol-xxxxxxxxxxxxxx).

Step 3: Stop the original instance

Choose **Actions**, select **Instance State**, and then select **Stop**. If **Stop** is disabled, either the instance is already stopped or its root device is an instance store volume.

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

Step 4: Launch a temporary instance

Choose **Launch Instance**, and then use the launch wizard to launch a *temporary* instance with the following options:

- On the **Choose an AMI** page, select the same AMI that you used to launch the original instance. If this AMI is unavailable, you can create an AMI that you can use from the stopped instance. For more information, see [Creating an Amazon EBS-backed Linux AMI \(p. 123\)](#).
- On the **Choose an Instance Type** page, leave the default instance type that the wizard selects for you.

-
- On the **Configure Instance Details** page, specify the same Availability Zone as the original instance. If you're launching an instance in a VPC, select a subnet in this Availability Zone.
 - On the **Add Tags** page, add the tag **Name=Temporary** to the instance to indicate that this is a temporary instance.
 - On the **Review** page, choose **Launch**. Create a new key pair, download it to a safe location on your computer, and then choose **Launch Instances**.

Step 5: Detach the root volume from the original instance and attach it to the temporary instance

1. In the navigation pane, choose **Volumes** and select the root device volume for the original instance (you wrote down its volume ID in a previous step). Choose **Actions**, **Detach Volume**, and then select **Yes, Detach**. Wait for the state of the volume to become available. (You might need to choose the **Refresh** icon.)
2. With the volume still selected, choose **Actions**, and then select **Attach Volume**. Select the instance ID of the temporary instance, write down the device name specified under **Device** (for example, `/dev/sdf`), and then choose **Attach**.

Note

If you launched your original instance from an AWS Marketplace AMI and your volume contains AWS Marketplace codes, you must first stop the temporary instance before you can attach the volume.

Step 6: Add the new public key to `authorized_keys` on the original volume mounted to the temporary instance

1. Connect to the temporary instance.
2. From the temporary instance, mount the volume that you attached to the instance so that you can access its file system. For example, if the device name is `/dev/sdf`, use the following commands to mount the volume as `/mnt/tempvol`.

Note

The device name might appear differently on your instance. For example, devices mounted as `/dev/sdf` might show up as `/dev/xvdf` on the instance. Some versions of Red Hat (or its variants, such as CentOS) might even increment the trailing letter by 4 characters, where `/dev/sdf` becomes `/dev/xvdg`.

- a. Use the `lsblk` command to determine if the volume is partitioned.

```
[ec2-user ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda   202:0    0   8G  0 disk
##xvda1 202:1    0   8G  0 part /
xvdf   202:80   0 101G  0 disk
##xvdf1 202:81   0 101G  0 part
xvdg   202:96   0   30G  0 disk
```

In the preceding example, `/dev/xvda` and `/dev/xvdf` are partitioned volumes, and `/dev/xvdg` is not. If your volume is partitioned, you mount the partition (`/dev/xvdf1`) instead of the raw device (`/dev/xvdf`) in the next steps.

- b. Create a temporary directory to mount the volume.

```
[ec2-user ~]$ sudo mkdir /mnt/tempvol
```

-
- c. Mount the volume (or partition) at the temporary mount point, using the volume name or device name that you identified earlier. The required command depends on your operating system's file system.

- Amazon Linux, Ubuntu, and Debian

```
[ec2-user ~]$ sudo mount /dev/xvdf1 /mnt/tempvol
```

- Amazon Linux 2, CentOS, SUSE Linux 12, and RHEL 7.x

```
[ec2-user ~]$ sudo mount -o nouuid /dev/xvdf1 /mnt/tempvol
```

Note

If you get an error stating that the file system is corrupt, run the following command to use the **fsck** utility to check the file system and repair any issues:

```
[ec2-user ~]$ sudo fsck /dev/xvdf1
```

3. From the temporary instance, use the following command to update `authorized_keys` on the mounted volume with the new public key from the `authorized_keys` for the temporary instance.

Important

The following examples use the Amazon Linux user name `ec2-user`. You might need to substitute a different user name, such as `ubuntu` for Ubuntu instances.

```
[ec2-user ~]$ cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```

If this copy succeeded, you can go to the next step.

(Optional) Otherwise, if you don't have permission to edit files in `/mnt/tempvol`, you must update the file using **sudo** and then check the permissions on the file to verify that you are able to log into the original instance. Use the following command to check the permissions on the file.

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh
total 4
-rw----- 1 222 500 398 Sep 13 22:54 authorized_keys
```

In this example output, `222` is the user ID and `500` is the group ID. Next, use **sudo** to re-run the copy command that failed.

```
[ec2-user ~]$ sudo cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/
authorized_keys
```

Run the following command again to determine whether the permissions changed.

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh
```

If the user ID and group ID have changed, use the following command to restore them.

```
[ec2-user ~]$ sudo chown 222:500 /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```

Step 7: Unmount and detach the original volume from the temporary instance, and reattach it to the original instance

1. From the temporary instance, unmount the volume that you attached so that you can reattach it to the original instance. For example, use the following command to unmount the volume at /mnt/tempvol.

```
[ec2-user ~]$ sudo umount /mnt/tempvol
```

2. Detach the volume from the temporary instance (you unmounted it in the previous step): From the Amazon EC2 console, select the root device volume for the original instance (you wrote down volume ID in a previous step), choose **Actions**, **Detach Volume**, and then select **Yes, Detach**. Wait for the state of the volume to become available. (You might need to choose the **Refresh** icon.)
3. Reattach the volume to the original instance: With the volume still selected, choose **Actions**, **Attach Volume**. Select the instance ID of the original instance, specify the device name that you noted earlier in [Step 2 \(p. 995\)](#) for the original root device attachment (/dev/sda1 or /dev/xvda), and then choose **Attach**.

Important

If you don't specify the same device name as the original attachment, you cannot start the original instance. Amazon EC2 expects the root device volume at sda1 or /dev/xvda.

Step 8: Connect to the original instance using the new key pair

Select the original instance, choose **Actions**, select **Instance State**, and then choose **Start**. After the instance enters the **running** state, you can connect to it using the private key file for your new key pair.

Note

If the name of your new key pair and corresponding private key file is different from the name of the original key pair, ensure that you specify the name of the new private key file when you connect to your instance.

Step 9: Clean up

(Optional) You can terminate the temporary instance if you have no further use for it. Select the temporary instance, choose **Actions**, select **Instance State**, and then choose **Terminate**.

Deleting your key pair

When you delete a key pair, you are only deleting the Amazon EC2 copy of the public key. Deleting a key pair doesn't affect the private key on your computer or the public key on any instances that already launched using that key pair. You can't launch a new instance using a deleted key pair, but you can continue to connect to any instances that you launched using a deleted key pair, as long as you still have the private key (.pem) file.

If you're using an Auto Scaling group (for example, in an Elastic Beanstalk environment), ensure that the key pair you're deleting is not specified in your launch configuration. Amazon EC2 Auto Scaling launches a replacement instance if it detects an unhealthy instance; however, the instance launch fails if the key pair cannot be found.

You can delete a key pair using one of the following methods.

New console

To delete your key pair

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Key Pairs**.
3. Select the key pair to delete and choose **Delete**.
4. In the confirmation field, enter **Delete** and then choose **Delete**.

Old console

To delete your key pair

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
3. Select the key pair and choose **Delete**.
4. When prompted, choose **Yes**.

AWS CLI

To delete your key pair

Use the [delete-key-pair](#) AWS CLI command.

PowerShell

To delete your key pair

Use the [Remove-EC2KeyPair](#) AWS Tools for Windows PowerShell command.

If you create a Linux AMI from an instance, and then use the AMI to launch a new instance in a different Region or account, the new instance includes the public key from the original instance. This enables you to connect to the new instance using the same private key file as your original instance. You can remove this public key from your instance by removing its entry from the `.ssh/authorized_keys` file using a text editor of your choice. For more information about managing users on your instance and providing remote access using a specific key pair, see [Managing user accounts on your Amazon Linux instance \(p. 620\)](#).

Amazon EC2 security groups for Linux instances

A *security group* acts as a virtual firewall for your EC2 instances to control incoming and outgoing traffic. Inbound rules control the incoming traffic to your instance, and outbound rules control the outgoing traffic from your instance. When you launch an instance, you can specify one or more security groups. If you don't specify a security group, Amazon EC2 uses the default security group. You can add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time. New and modified rules are automatically applied to all instances that are associated with the security group. When Amazon EC2 decides whether to allow traffic to reach an instance, it evaluates all of the rules from all of the security groups that are associated with the instance.

When you launch an instance in a VPC, you must specify a security group that's created for that VPC. After you launch an instance, you can change its security groups. Security groups are associated with network interfaces. Changing an instance's security groups changes the security groups associated with the primary network interface (`eth0`). For more information, see [Changing an instance's security groups](#) in the *Amazon VPC User Guide*. You can also change the security groups associated with any other network interface. For more information, see [Modifying network interface attributes \(p. 807\)](#).

Security is a shared responsibility between AWS and you. For more information, see [Security in Amazon EC2 \(p. 914\)](#). AWS provides security groups as one of the tools for securing your instances, and you need to configure them to meet your security needs. If you have requirements that aren't fully met by

security groups, you can maintain your own firewall on any of your instances in addition to using security groups.

To allow traffic to a Windows instance, see [Amazon EC2 security groups for Windows instances](#) in the [Amazon EC2 User Guide for Windows Instances](#).

Contents

- [Security group rules \(p. 1000\)](#)
 - [Connection tracking \(p. 1002\)](#)
- [Default security groups \(p. 1003\)](#)
- [Custom security groups \(p. 1003\)](#)
- [Working with security groups \(p. 1004\)](#)
 - [Creating a security group \(p. 1004\)](#)
 - [Copying a security group \(p. 1005\)](#)
 - [Viewing your security groups \(p. 1005\)](#)
 - [Adding rules to a security group \(p. 1006\)](#)
 - [Updating Security Group Rules \(p. 1009\)](#)
 - [Deleting rules from a security group \(p. 1010\)](#)
 - [Deleting a security group \(p. 1010\)](#)
- [Security group rules reference \(p. 1011\)](#)
 - [Web server rules \(p. 1011\)](#)
 - [Database server rules \(p. 1012\)](#)
 - [Rules to connect to instances from your computer \(p. 1013\)](#)
 - [Rules to connect to instances from an instance with the same security group \(p. 1014\)](#)
 - [Rules for ping/ICMP \(p. 1014\)](#)
 - [DNS server rules \(p. 1015\)](#)
 - [Amazon EFS rules \(p. 1015\)](#)
 - [Elastic Load Balancing rules \(p. 1015\)](#)
 - [VPC peering rules \(p. 1016\)](#)

Security group rules

The rules of a security group control the inbound traffic that's allowed to reach the instances that are associated with the security group. The rules also control the outbound traffic that's allowed to leave them.

The following are the characteristics of security group rules:

- By default, security groups allow all outbound traffic.
- Security group rules are always permissive; you can't create rules that deny access.
- Security group rules enable you to filter traffic based on protocols and port numbers.
- Security groups are stateful—if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. For VPC security groups, this also means that responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules. For more information, see [Connection tracking \(p. 1002\)](#).
- You can add and remove rules at any time. Your changes are automatically applied to the instances that are associated with the security group.

The effect of some rule changes can depend on how the traffic is tracked. For more information, see [Connection tracking \(p. 1002\)](#).

- When you associate multiple security groups with an instance, the rules from each security group are effectively aggregated to create one set of rules. Amazon EC2 uses this set of rules to determine whether to allow access.

You can assign multiple security groups to an instance. Therefore, an instance can have hundreds of rules that apply. This might cause problems when you access the instance. We recommend that you condense your rules as much as possible.

For each rule, you specify the following:

- Name:** The name for the security group (for example, `my-security-group`).

A name can be up to 255 characters in length. Allowed characters are a-z, A-Z, 0-9, spaces, and `_:-/()`. When the name contains trailing spaces, we trim the spaces when we save the name. For example, if you enter "Test Security Group " for the name, we store it as "Test Security Group".

- Protocol:** The protocol to allow. The most common protocols are 6 (TCP), 17 (UDP), and 1 (ICMP).
- Port range:** For TCP, UDP, or a custom protocol, the range of ports to allow. You can specify a single port number (for example, 22), or range of port numbers (for example, 7000–8000).
- ICMP type and code:** For ICMP, the ICMP type and code.
- Source or destination:** The source (inbound rules) or destination (outbound rules) for the traffic. Specify one of these options:
 - An individual IPv4 address. You must use the /32 prefix length; for example, `203.0.113.1/32`.
 - An individual IPv6 address. You must use the /128 prefix length; for example, `2001:db8:1234:1a00::123/128`.
 - A range of IPv4 addresses, in CIDR block notation; for example, `203.0.113.0/24`.
 - A range of IPv6 addresses, in CIDR block notation; for example, `2001:db8:1234:1a00::/64`.
 - A prefix list ID, for example, `p1-1234abc1234abc123`. For more information, see [Prefix lists](#) in the *Amazon VPC User Guide*.
- Another security group. This allows instances that are associated with the specified security group to access instances associated with this security group. Choosing this option does not add rules from the source security group to this security group. You can specify one of the following security groups:
 - The current security group
 - A different security group for the same VPC
 - A different security group for a peer VPC in a VPC peering connection
- (Optional) Description:** You can add a description for the rule, which can help you identify it later. A description can be up to 255 characters in length. Allowed characters are a-z, A-Z, 0-9, spaces, and `_:-/()`.

When you specify a security group as the source or destination for a rule, the rule affects all instances that are associated with the security group. Incoming traffic is allowed based on the private IP addresses of the instances that are associated with the source security group (and not the public IP or Elastic IP addresses). For more information about IP addresses, see [Amazon EC2 instance IP addressing \(p. 759\)](#). If your security group rule references a security group in a peer VPC, and the referenced security group or VPC peering connection is deleted, the rule is marked as stale. For more information, see [Working with Stale Security Group Rules](#) in the *Amazon VPC Peering Guide*.

If there is more than one rule for a specific port, Amazon EC2 applies the most permissive rule. For example, if you have a rule that allows access to TCP port 22 (SSH) from IP address `203.0.113.1`, and another rule that allows access to TCP port 22 from everyone, everyone has access to TCP port 22.

Connection tracking

Your security groups use connection tracking to track information about traffic to and from the instance. Rules are applied based on the connection state of the traffic to determine if the traffic is allowed or denied. This approach allows security groups to be stateful. This means that responses to inbound traffic are allowed to flow out of the instance regardless of outbound security group rules, and vice versa. For example, if you initiate an ICMP ping command to your instance from your home computer, and your inbound security group rules allow ICMP traffic, information about the connection (including the port information) is tracked. Response traffic from the instance for the ping command is not tracked as a new request, but rather as an established connection and is allowed to flow out of the instance, even if your outbound security group rules restrict outbound ICMP traffic.

Not all flows of traffic are tracked. If a security group rule permits TCP or UDP flows for all traffic (`0.0.0.0/0` or `::/0`) and there is a corresponding rule in the other direction that permits all response traffic (`0.0.0.0/0` or `::/0`) for all ports (0-65535), then that flow of traffic is not tracked. The response traffic is therefore allowed to flow based on the inbound or outbound rule that permits the response traffic, and not on tracking information.

In the following example, the security group has specific inbound rules for TCP and ICMP traffic, and outbound rules that allow all outbound IPv4 and IPv6 traffic.

Inbound rules		
Protocol type	Port number	Source IP
TCP	22 (SSH)	203.0.113.1/32
TCP	80 (HTTP)	0.0.0.0/0
TCP	80 (HTTP)	::/0
ICMP	All	0.0.0.0/0

Outbound rules		
Protocol type	Port number	Destination IP
All	All	0.0.0.0/0
All	All	::/0

TCP traffic on port 22 (SSH) to and from the instance is tracked, because the inbound rule allows traffic from `203.0.113.1/32` only, and not all IP addresses (`0.0.0.0/0`). TCP traffic on port 80 (HTTP) to and from the instance is not tracked, because both the inbound and outbound rules allow all traffic (`0.0.0.0/0` or `::/0`). ICMP traffic is always tracked, regardless of rules. If you remove the outbound rule from the security group, all traffic to and from the instance is tracked, including traffic on port 80 (HTTP).

An untracked flow of traffic is immediately interrupted if the rule that enables the flow is removed or modified. For example, if you have an open (`0.0.0.0/0`) outbound rule, and you remove a rule that allows all (`0.0.0.0/0`) inbound SSH (TCP port 22) traffic to the instance (or modify it such that the connection would no longer be permitted), your existing SSH connections to the instance are immediately dropped. The connection was not previously being tracked, so the change will break the connection. On the other hand, if you have a narrower inbound rule that initially allows the SSH connection (meaning that the connection was tracked), but change that rule to no longer allow new connections from the address of the current SSH client, the existing connection will not be broken by changing the rule.

For protocols other than TCP, UDP, or ICMP, only the IP address and protocol number is tracked. If your instance sends traffic to another host (host B), and host B initiates the same type of traffic to your

instance in a separate request within 600 seconds of the original request or response, your instance accepts it regardless of inbound security group rules. Your instance accepts it because it's regarded as response traffic.

To ensure that traffic is immediately interrupted when you remove a security group rule, or to ensure that all inbound traffic is subject to firewall rules, you can use a network ACL for your subnet. Network ACLs are stateless and therefore do not automatically allow response traffic. For more information, see [Network ACLs](#) in the *Amazon VPC User Guide*.

Default security groups

Your AWS account automatically has a *default security group* for the default VPC in each Region. If you don't specify a security group when you launch an instance, the instance is automatically associated with the default security group for the VPC.

A default security group is named `default`, and it has an ID assigned by AWS. The following are the default rules for each default security group:

- Allows all inbound traffic from other instances associated with the default security group. The security group specifies itself as a source security group in its inbound rules.
- Allows all outbound traffic from the instance.

You can add or remove inbound and outbound rules for any default security group.

You can't delete a default security group. If you try to delete a default security group, you see the following error: `Client.CannotDelete: the specified group: "sg-51530134" name: "default" cannot be deleted by a user.`

Custom security groups

If you don't want your instances to use the default security group, you can create your own security groups and specify them when you launch your instances. You can create multiple security groups to reflect the different roles that your instances play; for example, a web server or a database server.

When you create a security group, you must provide it with a name and a description. Security group names and descriptions can be up to 255 characters in length, and are limited to the following characters:

a-z, A-Z, 0-9, spaces, and `._-:/()#@[]+=&;{}!$*`

A security group name cannot start with `sg-`. A security group name must be unique for the VPC.

The following are the default rules for a security group that you create:

- Allows no inbound traffic
- Allows all outbound traffic

After you've created a security group, you can change its inbound rules to reflect the type of inbound traffic that you want to reach the associated instances. You can also change its outbound rules.

For more information about the rules you can add to a security group, see [Security group rules reference \(p. 1011\)](#).

Working with security groups

You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group.

After you launch an instance, you can change its security groups. For more information, see [Changing an Instance's Security Groups](#) in the *Amazon VPC User Guide*.

You can create, view, update, and delete security groups and security group rules using the Amazon EC2 console and the command line tools.

Tasks

- [Creating a security group \(p. 1004\)](#)
- [Copying a security group \(p. 1005\)](#)
- [Viewing your security groups \(p. 1005\)](#)
- [Adding rules to a security group \(p. 1006\)](#)
- [Updating Security Group Rules \(p. 1009\)](#)
- [Deleting rules from a security group \(p. 1010\)](#)
- [Deleting a security group \(p. 1010\)](#)

Creating a security group

You can create a custom security group using one of the following methods. You must specify the VPC for which you're creating the security group.

New console

To create a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Choose **Create security group**.
4. In the **Basic details** section, do the following.
 - a. Enter a descriptive name and brief description for the security group. The name and description can be up to 255 characters long, and they can include a-z, A-Z, 0-9, spaces, and ._-:/()#@[]+=&;{}!\$*.
 - b. For **VPC**, choose the VPC in which to create the security group. The security group can only be used in the VPC in which it is created.
5. You can add security group rules now, or you can add them at any time after you have created the security group. For more information about adding security group rules, see [Adding rules to a security group \(p. 1006\)](#).
6. Choose **Create**.

Old console

To create a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Choose **Create Security Group**.

4. Specify a name and description for the security group.
5. For **VPC**, choose the ID of the VPC.
6. You can start adding rules, or you can choose **Create** to create the security group now (you can always add rules later). For more information about adding rules, see [Adding rules to a security group \(p. 1006\)](#).

Command line

To create a security group

Use one of the following commands:

- [create-security-group](#) (AWS CLI)
- [New-EC2SecurityGroup](#) (AWS Tools for Windows PowerShell)

Copying a security group

You can create a new security group by creating a copy of an existing one. When you copy a security group, the copy is created with the same inbound and outbound rules as the original security group. If the original security group is in a VPC, the copy is created in the same VPC unless you specify a different one.

The copy receives a new unique security group ID and you must give it a name. You can also add a description.

You can't copy a security group from one Region to another Region.

You can create a copy of a security group using one of the following methods.

New console

To copy a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group to copy and choose **Actions, Copy to new security group**.
4. Specify a name and optional description, and change the VPC and security group rules if needed.
5. Choose **Create**.

Old console

To copy a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group you want to copy, choose **Actions, Copy to new**.
4. The **Create Security Group** dialog opens, and is populated with the rules from the existing security group. Specify a name and description for your new security group. For **VPC**, choose the ID of the VPC. When you are done, choose **Create**.

Viewing your security groups

You can view information about your security groups using one of the following methods.

New console

To view your security groups

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Your security groups are listed. To view the details for a specific security group, including its inbound and outbound rules, choose its ID in the **Security group ID** column.

Old console

To view your security groups

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. (Optional) Select **VPC ID** from the filter list, then choose the ID of the VPC.
4. Select a security group. General information is displayed on the **Description** tab, inbound rules on the **Inbound** tab, outbound rules on the **Outbound** tab, and tags on the **Tags** tab.

Command line

To view your security groups

Use one of the following commands.

- [describe-security-groups \(AWS CLI\)](#)
- [Get-EC2SecurityGroup \(AWS Tools for Windows PowerShell\)](#)

Adding rules to a security group

When you add a rule to a security group, the new rule is automatically applied to any instances that are associated with the security group. There might be a short delay before the rule is applied. For more information about choosing security group rules for specific types of access, see [Security group rules reference \(p. 1011\)](#). For security group rule quotas, see [Amazon VPC quotas](#) in the *Amazon VPC User Guide*.

You can add rules to a security group using one of the following methods.

New console

To add an inbound rule to a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. In the list, select the security group and choose **Actions, Edit inbound rules**.
4. Choose **Add rule** and do the following.
 - a. For **Type**, choose the type of protocol to allow.
 - If you choose a custom TCP or UDP protocol, you must manually enter the port range to allow.
 - If you choose a custom ICMP protocol, you must choose the ICMP type name from **Protocol**, and, if applicable, the code name from **Port range**.
 - If you choose any other type, the protocol and port range are configured automatically.

- b. For **Source**, do one of the following.
 - Choose **Custom** and then enter an IP address in CIDR notation, a CIDR block, another security group, or a prefix list from which to allow inbound traffic.
 - Choose **Anywhere** to allow all inbound traffic of the specified protocol to reach your instance. This option automatically adds the 0.0.0.0/0 IPv4 CIDR block as an allowed source. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, authorize only a specific IP address or range of addresses to access your instance.
- If your security group is in a VPC that's enabled for IPv6, this option automatically adds a second rule for IPv6 traffic (::/0).
- Choose **My IP** to allow inbound traffic from only your local computer's public IPv4 address.
- c. For **Description**, optionally specify a brief description for the rule.
5. Choose **Preview changes**, **Save rules**.

To add an outbound rule to a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. In the list, select the security group and choose **Actions**, **Edit outbound rules**.
4. Choose **Add rule** and do the following.
 - a. For **Type**, choose the type of protocol to allow.
 - If you choose a custom TCP or UDP protocol, you must manually enter the port range to allow.
 - If you choose a custom ICMP protocol, you must choose the ICMP type name from **Protocol**, and, if applicable, the code name from **Port range**.
 - If you choose any other type, the protocol and port range are configured automatically.
 - b. For **Destination**, do one of the following.
 - Choose **Custom** and then enter an IP address in CIDR notation, a CIDR block, another security group, or a prefix list for which to allow outbound traffic.
 - Choose **Anywhere** to allow outbound traffic to all IP addresses. This option automatically adds the 0.0.0.0/0 IPv4 CIDR block as an allowed source.
 - If your security group is in a VPC that's enabled for IPv6, this option automatically adds a second rule for IPv6 traffic (::/0).
 - Choose **My IP** to allow outbound traffic only to your local computer's public IPv4 address.
- c. For **Description**, optionally specify a brief description for the rule.
5. Choose **Preview changes**, **Confirm**.

Old console

To add rules to a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups** and select the security group.
3. On the **Inbound** tab, choose **Edit**.
4. In the dialog, choose **Add Rule** and do the following:

- For **Type**, select the protocol.
- If you select a custom TCP or UDP protocol, specify the port range in **Port Range**.
- If you select a custom ICMP protocol, choose the ICMP type name from **Protocol**, and, if applicable, the code name from **Port Range**.
- For **Source**, choose one of the following:
 - **Custom**: in the provided field, you must specify an IP address in CIDR notation, a CIDR block, or another security group.
 - **Anywhere**: automatically adds the 0 . 0 . 0 . 0 /0 IPv4 CIDR block. This option enables all traffic of the specified type to reach your instance. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, authorize only a specific IP address or range of addresses to access your instance.

If your security group is in a VPC that's enabled for IPv6, the **Anywhere** option creates two rules—one for IPv4 traffic (0 . 0 . 0 . 0 /0) and one for IPv6 traffic (:: /0).

- **My IP**: automatically adds the public IPv4 address of your local computer.
- For **Description**, you can optionally specify a description for the rule.

For more information about the types of rules that you can add, see [Security group rules reference \(p. 1011\)](#).

5. Choose **Save**.
6. You can also specify outbound rules. On the **Outbound tab**, choose **Edit**, **Add Rule**, and do the following:
 - For **Type**, select the protocol.
 - If you select a custom TCP or UDP protocol, specify the port range in **Port Range**.
 - If you select a custom ICMP protocol, choose the ICMP type name from **Protocol**, and, if applicable, the code name from **Port Range**.
 - For **Destination**, choose one of the following:
 - **Custom**: in the provided field, you must specify an IP address in CIDR notation, a CIDR block, or another security group.
 - **Anywhere**: automatically adds the 0 . 0 . 0 . 0 /0 IPv4 CIDR block. This option enables outbound traffic to all IP addresses.
7. Choose **Save**.

Command line

To add rules to a security group

Use one of the following commands.

- [authorize-security-group-ingress](#) (AWS CLI)
- [Grant-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

To add one or more egress rules to a security group

Use one of the following commands.

- [authorize-security-group-egress \(AWS CLI\)](#)
- [Grant-EC2SecurityGroupEgress \(AWS Tools for Windows PowerShell\)](#)

Updating Security Group Rules

You can update a security group rule using one of the following methods.

New console

When you modify the protocol, port range, or source or destination of an existing security group rule using the console, the console deletes the existing rule and adds a new one for you.

To update a security group rule

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group to update, choose **Actions**, and then choose **Edit inbound rules** to update a rule for inbound traffic or **Edit outbound rules** to update a rule for outbound traffic.
4. Update the rule as required and then choose **Preview changes, Confirm**.

Old console

When you modify the protocol, port range, or source or destination of an existing security group rule using the console, the console deletes the existing rule and adds a new one for you.

To update a security group rule

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group to update, and choose the **Inbound** tab to update a rule for inbound traffic or the **Outbound** tab to update a rule for outbound traffic.
4. Choose **Edit**.
5. Modify the rule entry as required and choose **Save**.

Command line

You cannot modify the protocol, port range, or source or destination of an existing rule using the Amazon EC2 API or a command line tools. Instead, you must delete the existing rule and add a new rule. You can, however, update the description of an existing rule.

To update the description for an existing inbound rule

Use one of the following commands.

- [update-security-group-rule-descriptions-ingress \(AWS CLI\)](#)
- [Update-EC2SecurityGroupRuleIngressDescription \(AWS Tools for Windows PowerShell\)](#)

To update the description for an existing outbound rule

Use one of the following commands.

- [update-security-group-rule-descriptions-egress \(AWS CLI\)](#)
- [Update-EC2SecurityGroupRuleEgressDescription \(AWS Tools for Windows PowerShell\)](#)

Deleting rules from a security group

When you delete a rule from a security group, the change is automatically applied to any instances associated with the security group.

You can delete rules from a security group using one of the following methods.

New console

To delete a security group rule

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group to update, choose **Actions**, and then choose **Edit inbound rules** to remove an inbound rule or **Edit outbound rules** to remove an outbound rule.
4. Choose the remove button to the right of the rule to delete.
5. Choose **Preview changes**, **Confirm**.

Old console

To delete a security group rule

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select a security group.
4. On the **Inbound** tab (for inbound rules) or **Outbound** tab (for outbound rules), choose **Edit**. Choose **Delete** (a cross icon) next to each rule to delete.
5. Choose **Save**.

Command line

To remove one or more ingress rules from a security group

Use one of the following commands.

- [revoke-security-group-ingress](#) (AWS CLI)
- [Revoke-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

To remove one or more egress rules from a security group

Use one of the following commands.

- [revoke-security-group-egress](#) (AWS CLI)
- [Revoke-EC2SecurityGroupEgress](#) (AWS Tools for Windows PowerShell)

Deleting a security group

You can't delete a security group that is associated with an instance. You can't delete the default security group. You can't delete a security group that is referenced by a rule in another security group in the same VPC. If your security group is referenced by one of its own rules, you must delete the rule before you can delete the security group.

New console

To delete a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group to delete and choose **Actions, Delete security group, Delete**.

Old console

To delete a security group

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select a security group and choose **Actions, Delete Security Group**.
4. Choose **Yes, Delete**.

Command line

To delete a security group

Use one of the following commands.

- [delete-security-group \(AWS CLI\)](#)
- [Remove-EC2SecurityGroup \(AWS Tools for Windows PowerShell\)](#)

Security group rules reference

You can create a security group and add rules that reflect the role of the instance that's associated with the security group. For example, an instance that's configured as a web server needs security group rules that allow inbound HTTP and HTTPS access. Likewise, a database instance needs rules that allow access for the type of database, such as access over port 3306 for MySQL.

The following are examples of the kinds of rules that you can add to security groups for specific kinds of access.

Examples

- [Web server rules \(p. 1011\)](#)
- [Database server rules \(p. 1012\)](#)
- [Rules to connect to instances from your computer \(p. 1013\)](#)
- [Rules to connect to instances from an instance with the same security group \(p. 1014\)](#)
- [Rules for ping/ICMP \(p. 1014\)](#)
- [DNS server rules \(p. 1015\)](#)
- [Amazon EFS rules \(p. 1015\)](#)
- [Elastic Load Balancing rules \(p. 1015\)](#)
- [VPC peering rules \(p. 1016\)](#)

Web server rules

The following inbound rules allow HTTP and HTTPS access from any IP address. If your VPC is enabled for IPv6, you can add rules to control inbound HTTP and HTTPS traffic from IPv6 addresses.

Protocol type	Protocol number	Port	Source IP	Notes
TCP	6	80 (HTTP)	0.0.0.0/0	Allows inbound HTTP access from any IPv4 address
TCP	6	443 (HTTPS)	0.0.0.0/0	Allows inbound HTTPS access from any IPv4 address
TCP	6	80 (HTTP)	::/0	Allows inbound HTTP access from any IPv6 address
TCP	6	443 (HTTPS)	::/0	Allows inbound HTTPS access from any IPv6 address

Database server rules

The following inbound rules are examples of rules you might add for database access, depending on what type of database you're running on your instance. For more information about Amazon RDS instances, see the [Amazon RDS User Guide](#).

For the source IP, specify one of the following:

- A specific IP address or range of IP addresses (in CIDR block notation) in your local network
- A security group ID for a group of instances that access the database

Protocol type	Protocol number	Port	Notes
TCP	6	1433 (MS SQL)	The default port to access a Microsoft SQL Server database, for example, on an Amazon RDS instance
TCP	6	3306 (MySQL/Aurora)	The default port to access a MySQL or Aurora database, for example, on an Amazon RDS instance
TCP	6	5439 (Redshift)	The default port to access an Amazon Redshift cluster database.
TCP	6	5432 (PostgreSQL)	The default port to access a PostgreSQL database, for example, on an Amazon RDS instance

Protocol type	Protocol number	Port	Notes
TCP	6	1521 (Oracle)	The default port to access an Oracle database, for example, on an Amazon RDS instance

You can optionally restrict outbound traffic from your database servers. For example, you might want to allow access to the internet for software updates, but restrict all other kinds of traffic. You must first remove the default outbound rule that allows all outbound traffic.

Protocol type	Protocol number	Port	Destination IP	Notes
TCP	6	80 (HTTP)	0.0.0.0/0	Allows outbound HTTP access to any IPv4 address
TCP	6	443 (HTTPS)	0.0.0.0/0	Allows outbound HTTPS access to any IPv4 address
TCP	6	80 (HTTP)	::/0	(IPv6-enabled VPC only) Allows outbound HTTP access to any IPv6 address
TCP	6	443 (HTTPS)	::/0	(IPv6-enabled VPC only) Allows outbound HTTPS access to any IPv6 address

Rules to connect to instances from your computer

To connect to your instance, your security group must have inbound rules that allow SSH access (for Linux instances) or RDP access (for Windows instances).

Protocol type	Protocol number	Port	Source IP
TCP	6	22 (SSH)	The public IPv4 address of your computer, or a range of IP addresses (in CIDR block notation) in your local network. If your VPC is enabled for IPv6 and your instance has an IPv6 address, you can enter an IPv6 address or range.
TCP	6	3389 (RDP)	The public IPv4 address of your computer, or a

Protocol type	Protocol number	Port	Source IP
			range of IP addresses (in CIDR block notation) in your local network. If your VPC is enabled for IPv6 and your instance has an IPv6 address, you can enter an IPv6 address or range.

Rules to connect to instances from an instance with the same security group

To allow instances that are associated with the same security group to communicate with each other, you must explicitly add rules for this.

The following table describes the inbound rule for a security group that enables associated instances to communicate with each other. The rule allows all types of traffic.

Protocol type	Protocol number	Ports	Source IP
-1 (All)	-1 (All)	-1 (All)	The ID of the security group

Rules for ping/ICMP

The `ping` command is a type of ICMP traffic. To ping your instance, you must add the following inbound ICMP rule.

Protocol type	Protocol number	ICMP type	ICMP code	Source IP
ICMP	1	8 (Echo)	N/A	The public IPv4 address of your computer, or a range of IPv4 addresses (in CIDR block notation) in your local network

To use the `ping6` command to ping the IPv6 address for your instance, you must add the following inbound ICMPv6 rule.

Protocol type	Protocol number	ICMP type	ICMP code	Source IP
ICMPv6	58	128 (Echo)	0	The IPv6 address of your computer, or a range of IPv6 addresses (in CIDR block notation) in your local network

DNS server rules

If you've set up your EC2 instance as a DNS server, you must ensure that TCP and UDP traffic can reach your DNS server over port 53.

For the source IP, specify one of the following:

- An IP address or range of IP addresses (in CIDR block notation) in a network
- The ID of a security group for the set of instances in your network that require access to the DNS server

Protocol type	Protocol number	Port
TCP	6	53
UDP	17	53

Amazon EFS rules

If you're using an Amazon EFS file system with your Amazon EC2 instances, the security group that you associate with your Amazon EFS mount targets must allow traffic over the NFS protocol.

Protocol type	Protocol number	Ports	Source IP	Notes
TCP	6	2049 (NFS)	The ID of the security group.	Allows inbound NFS access from resources (including the mount target) associated with this security group.

To mount an Amazon EFS file system on your Amazon EC2 instance, you must connect to your instance. Therefore, the security group associated with your instance must have rules that allow inbound SSH from your local computer or local network.

Protocol type	Protocol number	Ports	Source IP	Notes
TCP	6	22 (SSH)	The IP address range of your local computer, or the range of IP addresses (in CIDR block notation) for your network.	Allows inbound SSH access from your local computer.

Elastic Load Balancing rules

If you're using a load balancer, the security group associated with your load balancer must have rules that allow communication with your instances or targets.

Inbound				
Protocol type	Protocol number	Port	Source IP	Notes
TCP	6	The listener port	For an Internet-facing load-balancer: 0.0.0.0/0 (all IPv4 addresses)	Allow inbound traffic on the load balancer listener port.
			For an internal load-balancer: the IPv4 CIDR block of the VPC	
Outbound				
Protocol type	Protocol number	Port	Destination IP	Notes
TCP	6	The instance listener port	The ID of the instance security group	Allow outbound traffic to instances on the instance listener port.
TCP	6	The health check port	The ID of the instance security group	Allow outbound traffic to instances on the health check port.

The security group rules for your instances must allow the load balancer to communicate with your instances on both the listener port and the health check port.

Inbound				
Protocol type	Protocol number	Port	Source IP	Notes
TCP	6	The instance listener port	The ID of the load balancer security group	Allow traffic from the load balancer on the instance listener port.
TCP	6	The health check port	The ID of the load balancer security group	Allow traffic from the load balancer on the health check port.

For more information, see [Configure security groups for your Classic Load Balancer](#) in the *User Guide for Classic Load Balancers*, and [Security groups for your Application Load Balancer](#) in the *User Guide for Application Load Balancers*.

VPC peering rules

You can update the inbound or outbound rules for your VPC security groups to reference security groups in the peered VPC. Doing so allows traffic to flow to and from instances that are associated with the referenced security group in the peered VPC. For more information about how to configure security groups for VPC peering, see [Updating your security groups to reference peer VPC groups](#).

Update management in Amazon EC2

We recommend that you regularly patch, update, and secure the operating system and applications on your EC2 instances. You can use [AWS Systems Manager Patch Manager](#) to automate the process of installing security-related updates for both the operating system and applications. Alternatively, you can use any automatic update services or recommended processes for installing updates that are provided by the application vendor.

Compliance validation for Amazon EC2

Third-party auditors assess the security and compliance of Amazon EC2 as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see [AWS Services in Scope by Compliance Program](#). For general information, see [AWS Compliance Programs](#).

You can download third-party audit reports using AWS Artifact. For more information, see [Downloading Reports in AWS Artifact](#).

Your compliance responsibility when using Amazon EC2 is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- [Security and Compliance Quick Start Guides](#) – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- [Architecting for HIPAA Security and Compliance Whitepaper](#) – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- [AWS Compliance Resources](#) – This collection of workbooks and guides might apply to your industry and location.
- [Evaluating Resources with Rules](#) in the *AWS Config Developer Guide* – AWS Config; assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- [AWS Security Hub](#) – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

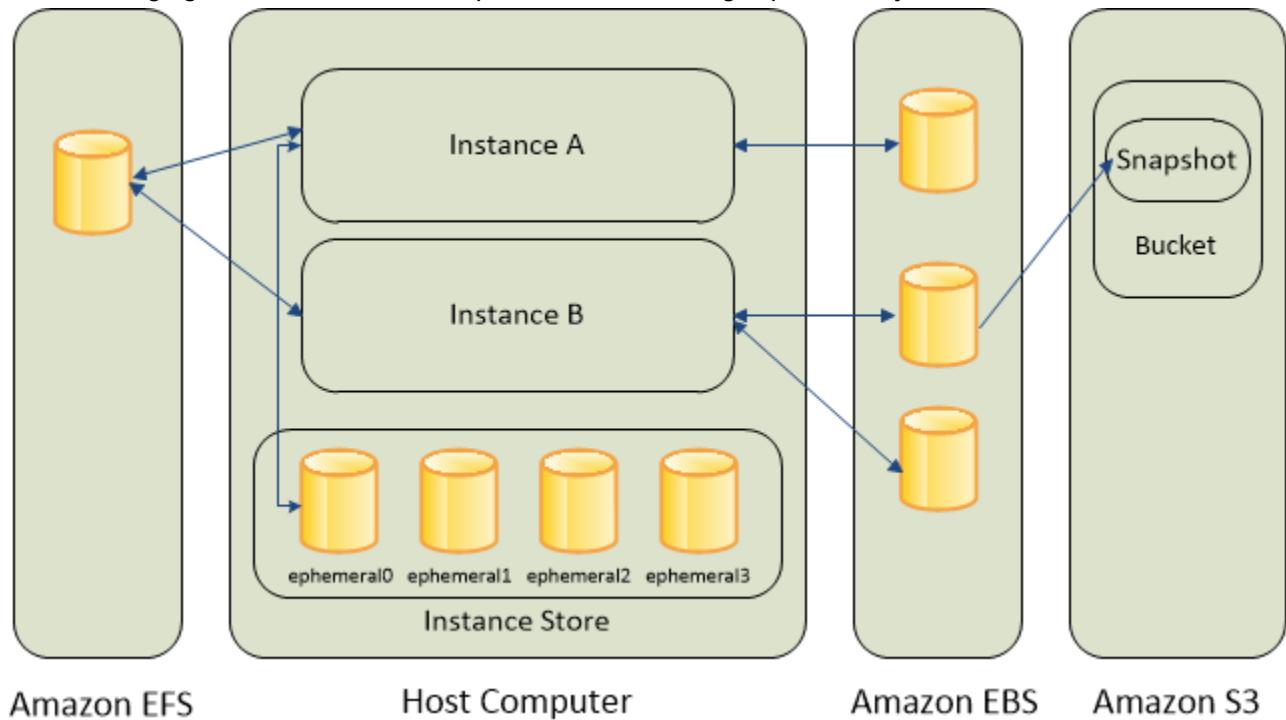
Storage

Amazon EC2 provides you with flexible, cost effective, and easy-to-use data storage options for your instances. Each option has a unique combination of performance and durability. These storage options can be used independently or in combination to suit your requirements.

After reading this section, you should have a good understanding about how you can use the data storage options supported by Amazon EC2 to meet your specific requirements. These storage options include the following:

- [Amazon Elastic Block Store \(p. 1019\)](#)
- [Amazon EC2 instance store \(p. 1188\)](#)
- [Using Amazon EFS with Amazon EC2 \(p. 1206\)](#)
- [Using Amazon S3 with Amazon EC2 \(p. 1204\)](#)

The following figure shows the relationship between these storage options and your instance.



Amazon EBS

Amazon EBS provides durable, block-level storage volumes that you can attach to a running instance. You can use Amazon EBS as a primary storage device for data that requires frequent and granular updates. For example, Amazon EBS is the recommended storage option when you run a database on an instance.

An EBS volume behaves like a raw, unformatted, external block device that you can attach to a single instance. The volume persists independently from the running life of an instance. After an EBS volume

is attached to an instance, you can use it like any other physical hard drive. As illustrated in the previous figure, multiple volumes can be attached to an instance. You can also detach an EBS volume from one instance and attach it to another instance. You can dynamically change the configuration of a volume attached to an instance. EBS volumes can also be created as encrypted volumes using the Amazon EBS encryption feature. For more information, see [Amazon EBS encryption \(p. 1121\)](#).

To keep a backup copy of your data, you can create a *snapshot* of an EBS volume, which is stored in Amazon S3. You can create an EBS volume from a snapshot, and attach it to another instance. For more information, see [Amazon Elastic Block Store \(p. 1019\)](#).

Amazon EC2 instance store

Many instances can access storage from disks that are physically attached to the host computer. This disk storage is referred to as *instance store*. Instance store provides temporary block-level storage for instances. The data on an instance store volume persists only during the life of the associated instance; if you stop or terminate an instance, any data on instance store volumes is lost. For more information, see [Amazon EC2 instance store \(p. 1188\)](#).

Amazon EFS file system

Amazon EFS provides scalable file storage for use with Amazon EC2. You can create an EFS file system and configure your instances to mount the file system. You can use an EFS file system as a common data source for workloads and applications running on multiple instances. For more information, see [Using Amazon EFS with Amazon EC2 \(p. 1206\)](#).

Amazon S3

Amazon S3 provides access to reliable and inexpensive data storage infrastructure. It is designed to make web-scale computing easier by enabling you to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web. For example, you can use Amazon S3 to store backup copies of your data and applications. Amazon EC2 uses Amazon S3 to store EBS snapshots and instance store-backed AMIs. For more information, see [Using Amazon S3 with Amazon EC2 \(p. 1204\)](#).

Adding storage

Every time you launch an instance from an AMI, a root storage device is created for that instance. The root storage device contains all the information necessary to boot the instance. You can specify storage volumes in addition to the root device volume when you create an AMI or launch an instance using *block device mapping*. For more information, see [Block device mapping \(p. 1211\)](#).

You can also attach EBS volumes to a running instance. For more information, see [Attaching an Amazon EBS volume to an instance \(p. 1042\)](#).

Storage pricing

For information about storage pricing, open [AWS Pricing](#), scroll down to **Services Pricing**, choose **Storage**, and then choose the storage option to open that storage option's pricing page. For information about estimating the cost of storage, see the [AWS Pricing Calculator](#).

Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances. EBS volumes behave like raw, unformatted block devices. You can mount these volumes as devices on your instances. EBS volumes that are attached to an instance are exposed as storage volumes that persist independently from the life of the instance. You can create a file system on top

of these volumes, or use them in any way you would use a block device (such as a hard drive). You can dynamically change the configuration of a volume attached to an instance.

We recommend Amazon EBS for data that must be quickly accessible and requires long-term persistence. EBS volumes are particularly well-suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage. Amazon EBS is well suited to both database-style applications that rely on random reads and writes, and to throughput-intensive applications that perform long, continuous reads and writes.

With Amazon EBS, you pay only for what you use. For more information about Amazon EBS pricing, see the Projecting Costs section of the [Amazon Elastic Block Store page](#).

Contents

- [Features of Amazon EBS \(p. 1020\)](#)
- [Amazon EBS volumes \(p. 1021\)](#)
- [Amazon EBS snapshots \(p. 1060\)](#)
- [Amazon EBS data services \(p. 1109\)](#)
- [Amazon EBS and NVMe on Linux instances \(p. 1136\)](#)
- [Amazon EBS-optimized instances \(p. 1139\)](#)
- [Amazon EBS volume performance on Linux instances \(p. 1156\)](#)
- [Amazon CloudWatch metrics for Amazon EBS \(p. 1172\)](#)
- [Amazon CloudWatch Events for Amazon EBS \(p. 1178\)](#)
- [Amazon EBS quotas \(p. 1188\)](#)

Features of Amazon EBS

- EBS volumes are created in a specific Availability Zone, and can then be attached to any instances in that same Availability Zone. To make a volume available outside of the Availability Zone, you can create a snapshot and restore that snapshot to a new volume anywhere in that Region. You can copy snapshots to other Regions and then restore them to new volumes there, making it easier to leverage multiple AWS Regions for geographical expansion, data center migration, and disaster recovery.
- Amazon EBS provides the following volume types: General Purpose SSD (gp2), Provisioned IOPS SSD (io1 and io2), Throughput Optimized HDD (st1), and Cold HDD (sc1). The following is a summary of performance and use cases for each volume type.
 - General Purpose SSD volumes offer a base performance of 3 IOPS/GiB, with the ability to burst to 3,000 IOPS for extended periods of time. These volumes are ideal for a broad range of use cases such as boot volumes, small and medium-size databases, and development and test environments. For more information, see [General Purpose SSD \(gp2\) volumes \(p. 1026\)](#).
 - Provisioned IOPS SSD volumes support up to 64,000 IOPS and 1,000 MiB/s of throughput. This allows you to predictably scale to tens of thousands of IOPS per EC2 instance. For more information, see [Provisioned IOPS SSD \(io1 and io2\) volumes \(p. 1029\)](#).
 - Throughput Optimized HDD volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. These volumes are ideal for large, sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing. For more information, see [Throughput Optimized HDD \(st1\) volumes \(p. 1030\)](#).
 - Cold HDD volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. These volumes are ideal for large, sequential, cold-data workloads. If you require infrequent access to your data and are looking to save costs, these volumes provides inexpensive block storage. For more information, see [Cold HDD \(sc1\) volumes \(p. 1033\)](#).
- You can create your EBS volumes as encrypted volumes, in order to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. When you create an encrypted EBS volume and attach it to a supported instance type, data stored at rest on the volume, disk I/O, and

snapshots created from the volume are all encrypted. The encryption occurs on the servers that host EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage. For more information, see [Amazon EBS encryption \(p. 1121\)](#).

- You can create point-in-time snapshots of EBS volumes, which are persisted to Amazon S3. Snapshots protect data for long-term durability, and they can be used as the starting point for new EBS volumes. The same snapshot can be used to instantiate as many volumes as you wish. These snapshots can be copied across AWS Regions. For more information, see [Amazon EBS snapshots \(p. 1060\)](#).
- Performance metrics, such as bandwidth, throughput, latency, and average queue length, are available through the AWS Management Console. These metrics, provided by Amazon CloudWatch, allow you to monitor the performance of your volumes to make sure that you are providing enough performance for your applications without paying for resources you don't need. For more information, see [Amazon EBS volume performance on Linux instances \(p. 1156\)](#).

Amazon EBS volumes

An Amazon EBS volume is a durable, block-level storage device that you can attach to your instances. After you attach a volume to an instance, you can use it as you would use a physical hard drive. EBS volumes are flexible. For current-generation volumes attached to current-generation instance types, you can dynamically increase size, modify the provisioned IOPS capacity, and change volume type on live production volumes.

You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. You can also use them for throughput-intensive applications that perform continuous disk scans. EBS volumes persist independently from the running life of an EC2 instance.

You can attach multiple EBS volumes to a single instance. The volume and instance must be in the same Availability Zone. Depending on the volume and instance types, you can use [Multi-Attach \(p. 1043\)](#) to mount a volume to multiple instances at the same time.

Amazon EBS provides the following volume types: General Purpose SSD (`gp2`), Provisioned IOPS SSD (`io1` and `io2`), Throughput Optimized HDD (`st1`), Cold HDD (`sc1`), and Magnetic (`standard`, a previous-generation type). They differ in performance characteristics and price, allowing you to tailor your storage performance and cost to the needs of your applications. For more information, see [Amazon EBS volume types \(p. 1023\)](#).

Your account has a limit on the number of EBS volumes that you can use, and the total storage available to you. For more information about these limits, and how to request an increase in your limits, see [Amazon EC2 service quotas \(p. 1241\)](#).

For more information about pricing, see [Amazon EBS Pricing](#).

Contents

- [Benefits of using EBS volumes \(p. 1022\)](#)
- [Amazon EBS volume types \(p. 1023\)](#)
- [Constraints on the size and configuration of an EBS volume \(p. 1037\)](#)
- [Creating an Amazon EBS volume \(p. 1040\)](#)
- [Attaching an Amazon EBS volume to an instance \(p. 1042\)](#)
- [Attaching a volume to multiple instances with Amazon EBS Multi-Attach \(p. 1043\)](#)
- [Making an Amazon EBS volume available for use on Linux \(p. 1046\)](#)
- [Viewing information about an Amazon EBS volume \(p. 1049\)](#)
- [Replacing an Amazon EBS volume using a previous snapshot \(p. 1050\)](#)
- [Monitoring the status of your volumes \(p. 1051\)](#)

- [Detaching an Amazon EBS volume from a Linux instance \(p. 1058\)](#)
- [Deleting an Amazon EBS volume \(p. 1060\)](#)

Benefits of using EBS volumes

EBS volumes provide benefits that are not provided by instance store volumes.

Data availability

When you create an EBS volume, it is automatically replicated within its Availability Zone to prevent data loss due to failure of any single hardware component. You can attach an EBS volume to any EC2 instance in the same Availability Zone. After you attach a volume, it appears as a native block device similar to a hard drive or other physical device. At that point, the instance can interact with the volume just as it would with a local drive. You can connect to the instance and format the EBS volume with a file system, such as ext3, and then install applications.

If you attach multiple volumes to a device that you have named, you can stripe data across the volumes for increased I/O and throughput performance.

You can attach an `io1` EBS volume to up to 16 Nitro-based instances. For more information, see [Attaching a volume to multiple instances with Amazon EBS Multi-Attach \(p. 1043\)](#). Otherwise, you can attach an EBS volume to a single instance.

You can get monitoring data for your EBS volumes, including root device volumes for EBS-backed instances, at no additional charge. For more information about monitoring metrics, see [Amazon CloudWatch metrics for Amazon EBS \(p. 1172\)](#). For information about tracking the status of your volumes, see [Amazon CloudWatch Events for Amazon EBS \(p. 1178\)](#).

Data persistence

An EBS volume is off-instance storage that can persist independently from the life of an instance. You continue to pay for the volume usage as long as the data persists.

EBS volumes that are attached to a running instance can automatically detach from the instance with their data intact when the instance is terminated if you uncheck the **Delete on Termination** check box when you configure EBS volumes for your instance on the EC2 console. The volume can then be reattached to a new instance, enabling quick recovery. If the check box for **Delete on Termination** is checked, the volume(s) will delete upon termination of the EC2 instance. If you are using an EBS-backed instance, you can stop and restart that instance without affecting the data stored in the attached volume. The volume remains attached throughout the stop-start cycle. This enables you to process and store the data on your volume indefinitely, only using the processing and storage resources when required. The data persists on the volume until the volume is deleted explicitly. The physical block storage used by deleted EBS volumes is overwritten with zeroes before it is allocated to another account. If you are dealing with sensitive data, you should consider encrypting your data manually or storing the data on a volume protected by Amazon EBS encryption. For more information, see [Amazon EBS encryption \(p. 1121\)](#).

By default, the root EBS volume that is created and attached to an instance at launch is deleted when that instance is terminated. You can modify this behavior by changing the value of the flag `DeleteOnTermination` to `false` when you launch the instance. This modified value causes the volume to persist even after the instance is terminated, and enables you to attach the volume to another instance.

By default, additional EBS volumes that are created and attached to an instance at launch are not deleted when that instance is terminated. You can modify this behavior by changing the value of the flag `DeleteOnTermination` to `true` when you launch the instance. This modified value causes the volumes to be deleted when the instance is terminated.

Data encryption

For simplified data encryption, you can create encrypted EBS volumes with the Amazon EBS encryption feature. All EBS volume types support encryption. You can use encrypted EBS volumes to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. Amazon EBS encryption uses 256-bit Advanced Encryption Standard algorithms (AES-256) and an Amazon-managed key infrastructure. The encryption occurs on the server that hosts the EC2 instance, providing encryption of data-in-transit from the EC2 instance to Amazon EBS storage. For more information, see [Amazon EBS encryption \(p. 1121\)](#).

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) master keys when creating encrypted volumes and any snapshots created from your encrypted volumes. The first time you create an encrypted EBS volume in a region, a default master key is created for you automatically. This key is used for Amazon EBS encryption unless you select a customer master key (CMK) that you created separately using AWS KMS. Creating your own CMK gives you more flexibility, including the ability to create, rotate, disable, define access controls, and audit the encryption keys used to protect your data. For more information, see the [AWS Key Management Service Developer Guide](#).

Snapshots

Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon S3, where it is stored redundantly in multiple Availability Zones. The volume does not need to be attached to a running instance in order to take a snapshot. As you continue to write data to a volume, you can periodically create a snapshot of the volume to use as a baseline for new volumes. These snapshots can be used to create multiple new EBS volumes or move volumes across Availability Zones. Snapshots of encrypted EBS volumes are automatically encrypted.

When you create a new volume from a snapshot, it's an exact copy of the original volume at the time the snapshot was taken. EBS volumes that are created from encrypted snapshots are automatically encrypted. By optionally specifying a different Availability Zone, you can use this functionality to create a duplicate volume in that zone. The snapshots can be shared with specific AWS accounts or made public. When you create snapshots, you incur charges in Amazon S3 based on the volume's total size. For a successive snapshot of the volume, you are only charged for any additional data beyond the volume's original size.

Snapshots are incremental backups, meaning that only the blocks on the volume that have changed after your most recent snapshot are saved. If you have a volume with 100 GiB of data, but only 5 GiB of data have changed since your last snapshot, only the 5 GiB of modified data is written to Amazon S3. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot.

To help categorize and manage your volumes and snapshots, you can tag them with metadata of your choice. For more information, see [Tagging your Amazon EC2 resources \(p. 1229\)](#).

To back up your volumes automatically, you can use [Amazon Data Lifecycle Manager \(p. 1097\)](#) or [AWS Backup](#).

Flexibility

EBS volumes support live configuration changes while in production. You can modify volume type, volume size, and IOPS capacity without service interruptions. For more information, see [Amazon EBS Elastic Volumes \(p. 1109\)](#).

Amazon EBS volume types

Amazon EBS provides the following volume types, which differ in performance characteristics and price, so that you can tailor your storage performance and cost to the needs of your applications. The volume types fall into two categories:

- SSD-backed volumes optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS
- HDD-backed volumes optimized for large streaming workloads where throughput (measured in MiB/s) is a better performance measure than IOPS

There are several factors that can affect the performance of EBS volumes, such as instance configuration, I/O characteristics, and workload demand. For more information about getting the most out of your EBS volumes, see [Amazon EBS volume performance on Linux instances \(p. 1156\)](#).

For more information about pricing, see [Amazon EBS Pricing](#).

Volume characteristics

The following table describes the use cases and performance characteristics for each volume type. The default volume type is General Purpose SSD (gp2).

	Solid-state drives (SSD)			Hard disk drives (HDD)	
Volume type	General Purpose SSD (gp2)	Provisioned IOPS SSD		Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads		Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.999% durability (0.001% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)
Use cases	<ul style="list-style-type: none"> • Recommended for most workloads • System boot volumes • Virtual desktops • Low-latency interactive apps • Development and test environments 	<ul style="list-style-type: none"> • Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume • Large database workloads, such as: <ul style="list-style-type: none"> • MongoDB • Cassandra • Microsoft SQL Server • MySQL • PostgreSQL • Oracle 	<ul style="list-style-type: none"> • Streaming workloads requiring consistent, fast throughput at a low price • Big data • Data warehouses • Log processing • Cannot be a boot volume 	<ul style="list-style-type: none"> • Throughput-oriented storage for large volumes of data that is infrequently accessed • Scenarios where the lowest storage cost is important • Cannot be a boot volume 	

	Solid-state drives (SSD)			Hard disk drives (HDD)	
Amazon EBS Multi-attach	Not supported	Not Supported	Supported	Not supported	Not supported
API name	gp2	io2	io1	st1	sc1
Volume size	1 GiB - 16 TiB	4 GiB - 16 TiB			500 GiB - 16 TiB
Dominant performance attribute	IOPS	IOPS			MiB/s
Max IOPS per volume	16,000 (16 KiB I/O) *	64,000 (16 KiB I/O) †			500 (1 MiB I/O)
Max throughput per volume	250 MiB/s *	1,000 MiB/s †			500 MiB/s
Max IOPS per instance ‡‡	160,000				
Max throughput per instance ‡‡	4,750 MB/s				

* The throughput limit is between 128 MiB/s and 250 MiB/s, depending on the volume size. Volumes smaller than or equal to 170 GiB deliver a maximum throughput of 128 MiB/s. Volumes larger than 170 GiB but smaller than 334 GiB deliver a maximum throughput of 250 MiB/s if burst credits are available. Volumes larger than or equal to 334 GiB deliver 250 MiB/s regardless of burst credits. Older gp2 volumes might not reach full performance unless you modify the volume. For more information, see [Amazon EBS Elastic Volumes \(p. 1109\)](#).

† Maximum IOPS and throughput are guaranteed only on [Instances built on the Nitro System \(p. 205\)](#) provisioned with more than 32,000 IOPS. Other instances guarantee up to 32,000 IOPS and 500 MiB/s. Older io1 volumes might not reach full performance unless you modify the volume. For more information, see [Amazon EBS Elastic Volumes \(p. 1109\)](#).

‡‡ To achieve this throughput, you must have an instance that supports [EBS optimization \(p. 1139\)](#).

Previous generation volume types

The following table describes previous-generation EBS volume types. If you need higher performance or performance consistency than previous-generation volumes can provide, we recommend that you consider using General Purpose SSD (gp2) or other current volume types. For more information, see [Previous Generation Volumes](#).

Hard disk drives (HDD)	
Volume type	Magnetic
Use cases	Workloads where data is infrequently accessed
API name	standard
Volume size	1 GiB-1 TiB

Hard disk drives (HDD)	
Max IOPS per volume	40–200
Max throughput per volume	40–90 MiB/s
Max IOPS per instance	80,000
Max throughput per instance	1,750 MB/s
Dominant performance attribute	IOPS

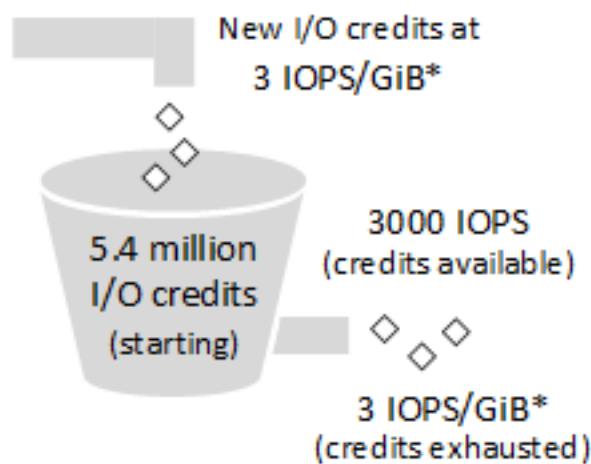
General Purpose SSD (gp2) volumes

General Purpose SSD (gp2) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. Between a minimum of 100 IOPS (at 33.33 GiB and below) and a maximum of 16,000 IOPS (at 5,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size. AWS designs gp2 volumes to deliver their provisioned performance 99% of the time. A gp2 volume can range in size from 1 GiB to 16 TiB.

I/O Credits and burst performance

The performance of gp2 volumes is tied to volume size, which determines the baseline performance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher baseline performance levels and accumulate I/O credits faster. I/O credits represent the available bandwidth that your gp2 volume can use to burst large amounts of I/O when more than the baseline performance is needed. The more credits your volume has for I/O, the more time it can burst beyond its baseline performance level and the better it performs when more performance is needed. The following diagram shows the burst-bucket behavior for gp2.

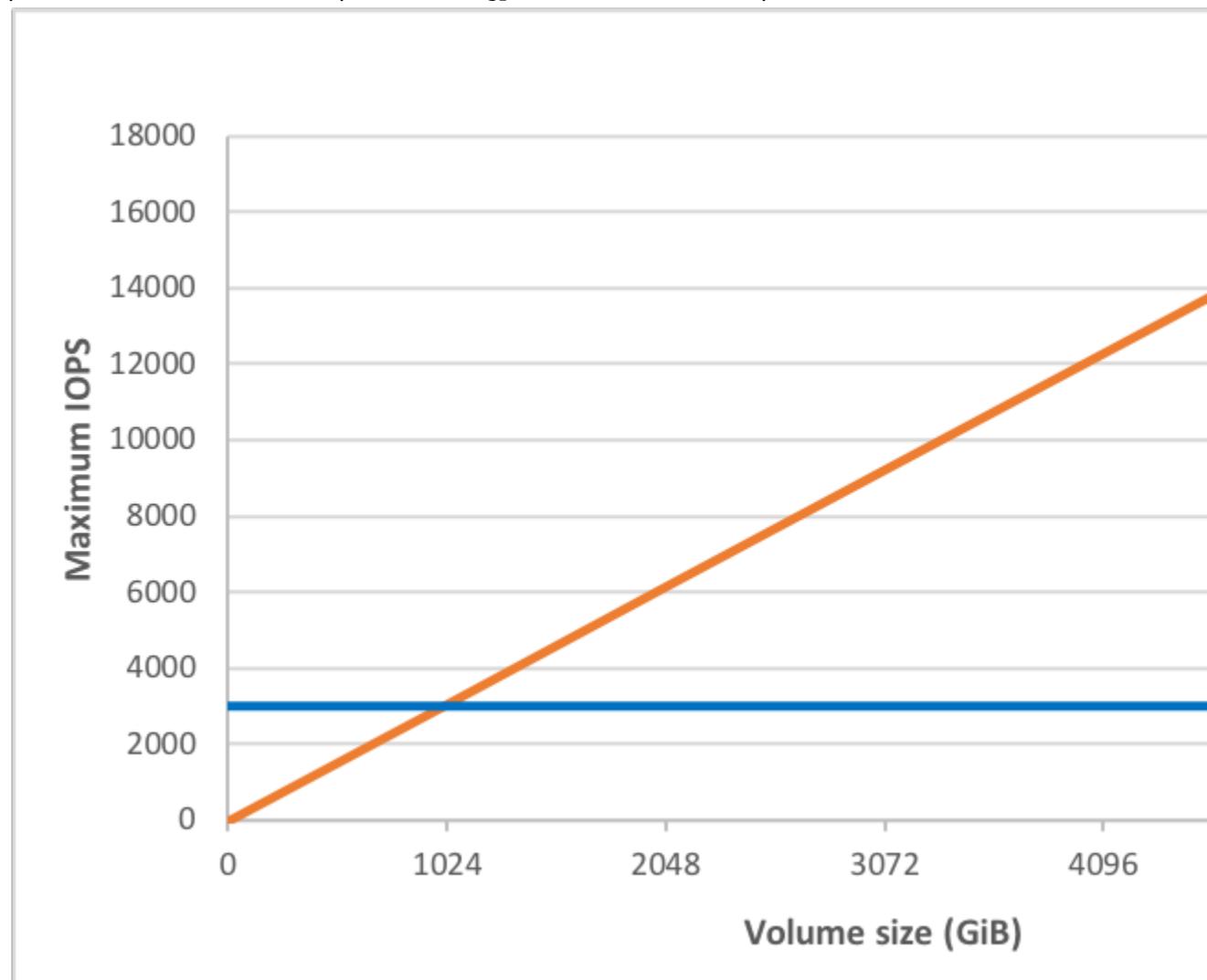
GP2 burst bucket



* Scaling linearly between minimum 100 IOPS and maximum 16,000 IOPS

Each volume receives an initial I/O credit balance of 5.4 million I/O credits, which is enough to sustain the maximum burst performance of 3,000 IOPS for at least 30 minutes. This initial credit balance is

designed to provide a fast initial boot cycle for boot volumes and to provide a good bootstrapping experience for other applications. Volumes earn I/O credits at the baseline performance rate of 3 IOPS per GiB of volume size. For example, a 100 GiB gp2 volume has a baseline performance of 300 IOPS.



When your volume requires more than the baseline performance I/O level, it draws on I/O credits in the credit balance to burst to the required performance level, up to a maximum of 3,000 IOPS. When your volume uses fewer I/O credits than it earns in a second, unused I/O credits are added to the I/O credit balance. The maximum I/O credit balance for a volume is equal to the initial credit balance (5.4 million I/O credits).

When the baseline performance of a volume is higher than maximum burst performance, I/O credits are never spent. If the volume is attached to an instance built on the [Nitro System \(p. 205\)](#), the burst balance is not reported. For other instances, the reported burst balance is 100%.

The burst duration of a volume is dependent on the size of the volume, the burst IOPS required, and the credit balance when the burst begins. This is shown in the following equation:

$$\text{Burst duration} = \frac{(\text{Credit balance})}{(\text{Burst IOPS}) - 3(\text{Volume size in GiB})}$$

The following table lists several volume sizes and the associated baseline performance of the volume (which is also the rate at which it accumulates I/O credits), the burst duration at the 3,000 IOPS maximum (when starting with a full credit balance), and the time in seconds that the volume would take to refill an empty credit balance.

Volume size (GiB)	Baseline performance (IOPS)	Burst duration when driving sustained 3,000 IOPS (second)	Seconds to fill empty credit balance when driving no IO
1	100	1,802	54,000
100	300	2,000	18,000
250	750	2,400	7,200
334 (Min. size for max throughput)	1,002	2,703	5,389
500	1,500	3,600	3,600
750	2,250	7,200	2,400
1,000	3,000	N/A*	N/A*
5,334 (Min. size for max IOPS)	16,000	N/A*	N/A*
16,384 (16 TiB, max volume size)	16,000	N/A*	N/A*

* The baseline performance of the volume exceeds the maximum burst performance.

What happens if I empty my I/O credit balance?

If your gp2 volume uses all of its I/O credit balance, the maximum IOPS performance of the volume remains at the baseline IOPS performance level (the rate at which your volume earns credits) and the volume's maximum throughput is reduced to the baseline IOPS multiplied by the maximum I/O size. Throughput can never exceed 250 MiB/s. When I/O demand drops below the baseline level and unused credits are added to the I/O credit balance, the maximum IOPS performance of the volume again exceeds the baseline. For example, a 100 GiB gp2 volume with an empty credit balance has a baseline performance of 300 IOPS and a throughput limit of 75 MiB/s (300 I/O operations per second * 256 KiB per I/O operation = 75 MiB/s). The larger a volume is, the greater the baseline performance is and the faster it replenishes the credit balance. For more information about how IOPS are measured, see [I/O characteristics and monitoring \(p. 1159\)](#).

If you notice that your volume performance is frequently limited to the baseline level (due to an empty I/O credit balance), you should consider using a larger gp2 volume (with a higher baseline performance level) or switching to an io1 or io2 volume for workloads that require sustained IOPS performance greater than 16,000 IOPS.

For information about using CloudWatch metrics and alarms to monitor your burst bucket balance, see [Monitoring the burst bucket balance for gp2, st1, and sc1 volumes \(p. 1037\)](#).

Throughput performance

Throughput for a gp2 volume can be calculated using the following formula, up to the throughput limit of 250 MiB/s:

Throughput in MiB/s = ((Volume size in GiB) × (IOPS per GiB) × (I/O size in KiB))

Assuming V = volume size, I = I/O size, R = I/O rate, and T = throughput, this can be simplified to:

$$T = V \cdot I \cdot R$$

The smallest volume size that achieves the maximum throughput is given by:

$$\begin{aligned} V &= \frac{T}{I \cdot R} \\ &= \frac{250 \text{ MiB/s}}{(256 \text{ KiB})(3 \text{ IOPS/GiB})} \\ &= \frac{[(250)(2^{20})(\text{Bytes})]/\text{s}}{(256)(2^{10})(\text{Bytes})([3 \text{ IOP/s}]/[(2^{30})(\text{Bytes})])} \\ &= \frac{(250)(2^{20})(2^{30})(\text{Bytes})}{(256)(2^{10})(3)} \\ &= 357,913,941,333 \text{ Bytes} \\ &= 333\# \text{ GiB (334 GiB in practice because volumes are provisioned in whole gibibytes)} \end{aligned}$$

Provisioned IOPS SSD (io1 and io2) volumes

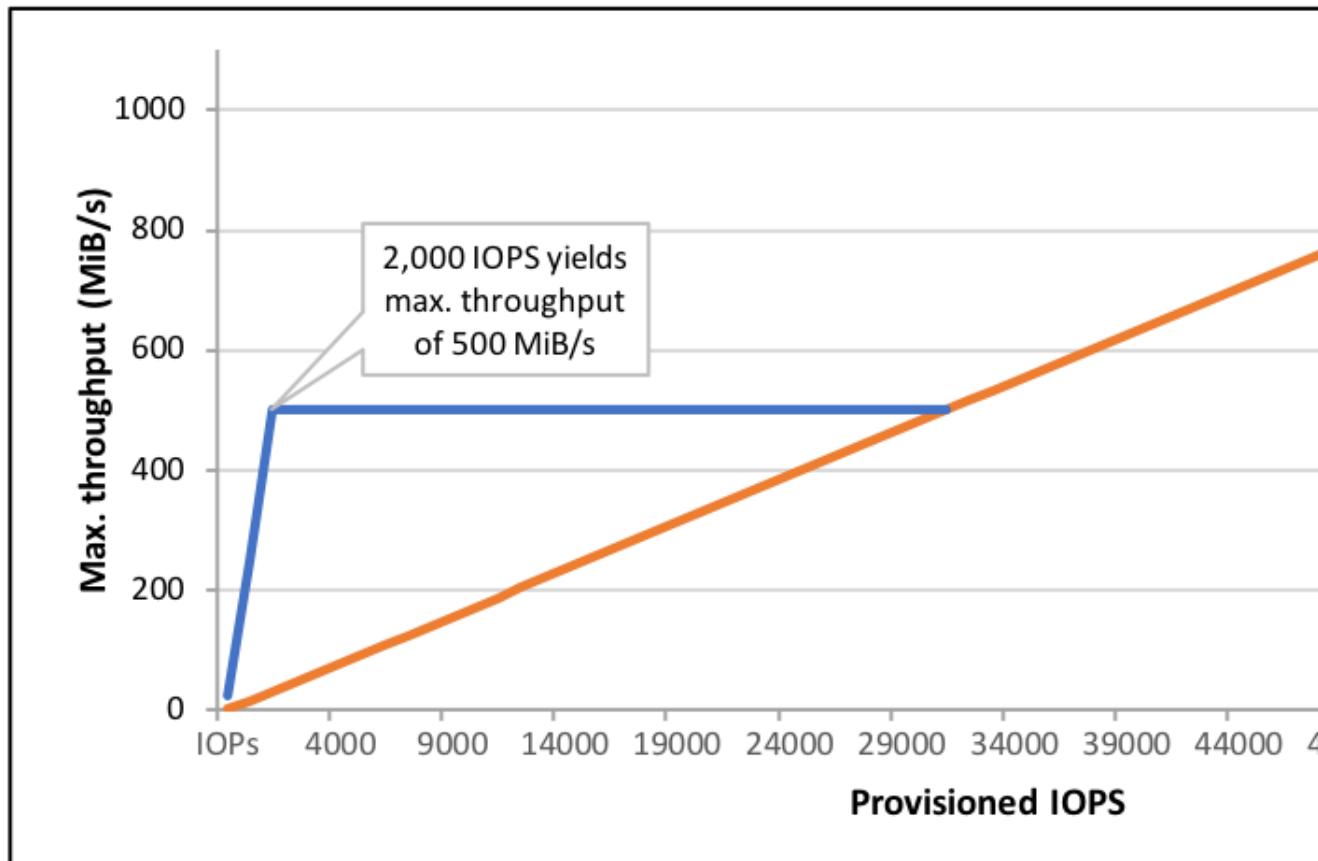
Provisioned IOPS SSD (io1 and io2) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency. Unlike gp2, which uses a bucket and credit model to calculate performance, io1 and io2 volumes allow you to specify a consistent IOPS rate when you create volumes, and Amazon EBS delivers the provisioned performance 99.9 percent of the time.

io1 volumes are designed to provide 99.8 to 99.9 percent volume durability with an annual failure rate (AFR) no higher than 0.2 percent, which translates to a maximum of two volume failures per 1,000 running volumes over a one-year period. io2 volumes are designed to provide 99.999 percent volume durability with an AFR no higher than 0.001 percent, which translates to a single volume failure per 100,000 running volumes over a one-year period.

io1 and io2 volumes can range in size from 4 GiB to 16 TiB. You can provision from 100 IOPS up to 64,000 IOPS per volume on [Instances built on the Nitro System \(p. 205\)](#) and up to 32,000 on other instances. The maximum ratio of provisioned IOPS to requested volume size (in GiB) is 50:1 for io1 volumes, and 500:1 for io2 volumes. For example, a 100 GiB io1 volume can be provisioned with up to 5,000 IOPS, while a 100 GiB io2 volume can be provisioned with up to 50,000 IOPS. On a supported instance type, the following volume sizes allow provisioning up to the 64,000 IOPS maximum:

- io1 volume 1,280 GiB in size or greater ($50 \times 1,280 \text{ GiB} = 64,000 \text{ IOPS}$)
- io2 volume 128 GiB in size or greater ($500 \times 128 \text{ GiB} = 64,000 \text{ IOPS}$)

io1 and io2 volumes provisioned with up to 32,000 IOPS support a maximum I/O size of 256 KiB and yield as much as 500 MiB/s of throughput. With the I/O size at the maximum, peak throughput is reached at 2,000 IOPS. A volume provisioned with more than 32,000 IOPS (up to the cap of 64,000 IOPS) supports a maximum I/O size of 16 KiB and yields as much as 1,000 MiB/s of throughput. The following graph illustrates these performance characteristics:



Your per-I/O latency experience depends on the provisioned IOPS and on your workload profile. For the best I/O latency experience, ensure that you provision IOPS to meet the I/O profile of your workload.

Note

Some AWS accounts created before 2012 might have access to Availability Zones in us-west-1 or ap-northeast-1 that do not support Provisioned IOPS SSD (`io1`) volumes. If you are unable to create an `io1` volume (or launch an instance with an `io1` volume in its block device mapping) in one of these Regions, try a different Availability Zone in the Region. You can verify that an Availability Zone supports `io1` volumes by creating a 4 GiB `io1` volume in that zone.

Throughput Optimized HDD (`st1`) volumes

Throughput Optimized HDD (`st1`) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. This volume type is a good fit for large, sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing. Bootable `st1` volumes are not supported.

Throughput Optimized HDD (`st1`) volumes, though similar to Cold HDD (`sc1`) volumes, are designed to support *frequently* accessed data.

This volume type is optimized for workloads involving large, sequential I/O, and we recommend that customers with workloads performing small, random I/O use `gp2`. For more information, see [Inefficiency of small read/writes on HDD \(p. 1037\)](#).

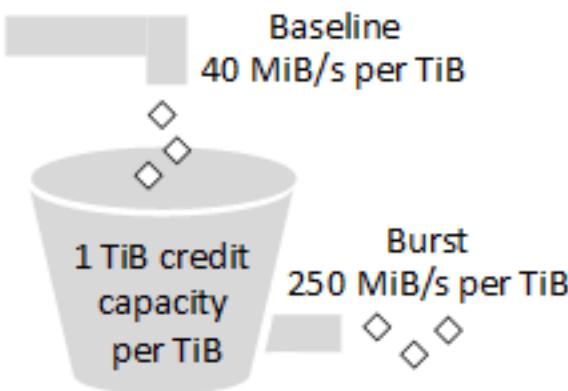
Throughput credits and burst performance

Like `gp2`, `st1` uses a burst-bucket model for performance. Volume size determines the baseline throughput of your volume, which is the rate at which the volume accumulates throughput credits. Volume size also determines the burst throughput of your volume, which is the rate at which you can

spend credits when they are available. Larger volumes have higher baseline and burst throughput. The more credits your volume has, the longer it can drive I/O at the burst level.

The following diagram shows the burst-bucket behavior for st1.

ST1 burst bucket



Subject to throughput and throughput-credit caps, the available throughput of an st1 volume is expressed by the following formula:

$$(\text{Volume size}) \times (\text{Credit accumulation rate per TiB}) = \text{Throughput}$$

For a 1-TiB st1 volume, burst throughput is limited to 250 MiB/s, the bucket fills with credits at 40 MiB/s, and it can hold up to 1 TiB-worth of credits.

Larger volumes scale these limits linearly, with throughput capped at a maximum of 500 MiB/s. After the bucket is depleted, throughput is limited to the baseline rate of 40 MiB/s per TiB.

On volume sizes ranging from 0.5 to 16 TiB, baseline throughput varies from 20 to a cap of 500 MiB/s, which is reached at 12.5 TiB as follows:

$$12.5 \text{ TiB} \times \frac{40 \text{ MiB/s}}{1 \text{ TiB}} = 500 \text{ MiB/s}$$

Burst throughput varies from 125 MiB/s to a cap of 500 MiB/s, which is reached at 2 TiB as follows:

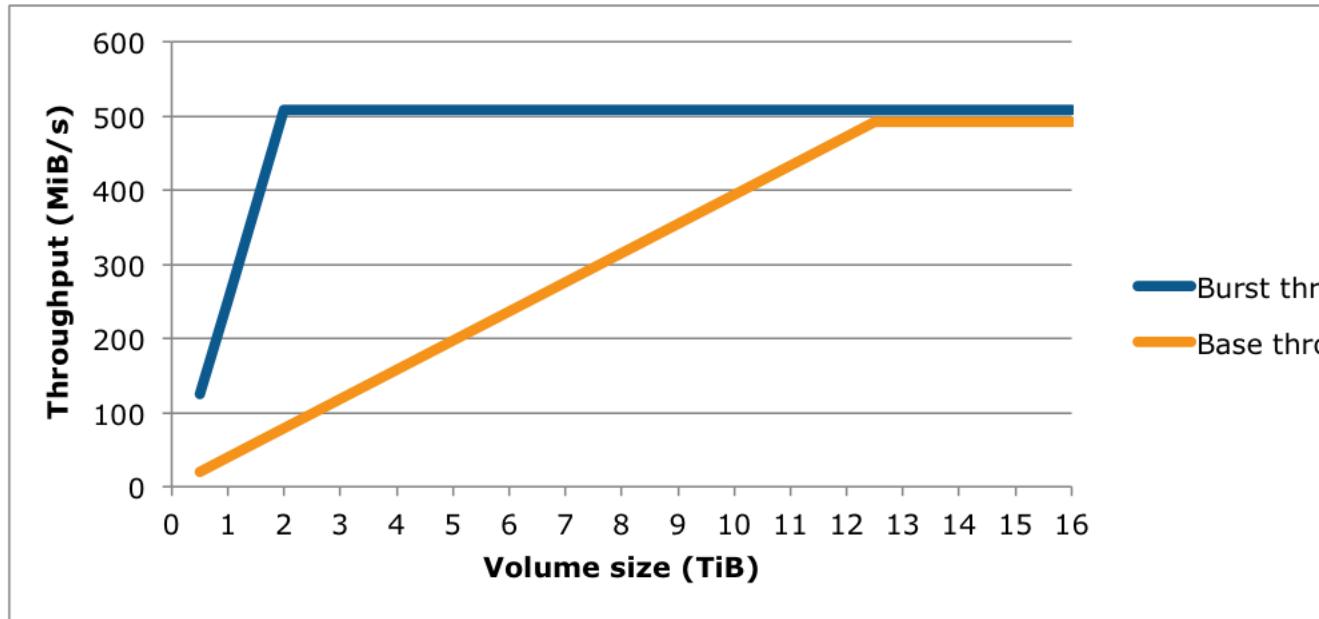
$$2 \text{ TiB} \times \frac{250 \text{ MiB/s}}{1 \text{ TiB}} = 500 \text{ MiB/s}$$

The following table states the full range of base and burst throughput values for st1:

Volume size (TiB)	ST1 base throughput (MiB/s)	ST1 burst throughput (MiB/s)
0.5	20	125
1	40	250
2	80	500
3	120	500

Volume size (TiB)	ST1 base throughput (MiB/s)	ST1 burst throughput (MiB/s)
4	160	500
5	200	500
6	240	500
7	280	500
8	320	500
9	360	500
10	400	500
11	440	500
12	480	500
12.5	500	500
13	500	500
14	500	500
15	500	500
16	500	500

The following diagram plots the table values:



Note

When you create a snapshot of a Throughput Optimized HDD (st1) volume, performance may drop as far as the volume's baseline value while the snapshot is in progress.

For information about using CloudWatch metrics and alarms to monitor your burst bucket balance, see [Monitoring the burst bucket balance for gp2, st1, and sc1 volumes \(p. 1037\)](#).

Cold HDD (sc1) volumes

Cold HDD (sc1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With a lower throughput limit than st1, sc1 is a good fit for large, sequential cold-data workloads. If you require infrequent access to your data and are looking to save costs, sc1 provides inexpensive block storage. Bootable sc1 volumes are not supported.

Cold HDD (sc1) volumes, though similar to Throughput Optimized HDD (st1) volumes, are designed to support *infrequently* accessed data.

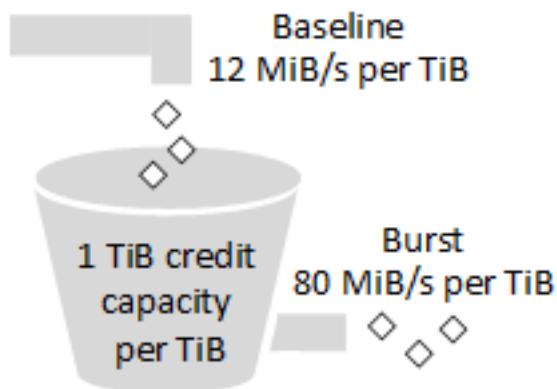
Note

This volume type is optimized for workloads involving large, sequential I/O, and we recommend that customers with workloads performing small, random I/O use gp2. For more information, see [Inefficiency of small read/writes on HDD \(p. 1037\)](#).

Throughput credits and burst performance

Like gp2, sc1 uses a burst-bucket model for performance. Volume size determines the baseline throughput of your volume, which is the rate at which the volume accumulates throughput credits. Volume size also determines the burst throughput of your volume, which is the rate at which you can spend credits when they are available. Larger volumes have higher baseline and burst throughput. The more credits your volume has, the longer it can drive I/O at the burst level.

SC1 burst bucket



Subject to throughput and throughput-credit caps, the available throughput of an sc1 volume is expressed by the following formula:

$$(\text{Volume size}) \times (\text{Credit accumulation rate per TiB}) = \text{Throughput}$$

For a 1-TiB sc1 volume, burst throughput is limited to 80 MiB/s, the bucket fills with credits at 12 MiB/s, and it can hold up to 1 TiB-worth of credits.

Larger volumes scale these limits linearly, with throughput capped at a maximum of 250 MiB/s. After the bucket is depleted, throughput is limited to the baseline rate of 12 MiB/s per TiB.

On volume sizes ranging from 0.5 to 16 TiB, baseline throughput varies from 6 MiB/s to a maximum of 192 MiB/s, which is reached at 16 TiB as follows:

$$12 \text{ MiB/s} \\ 16 \text{ TiB} \times \frac{12 \text{ MiB/s}}{1 \text{ TiB}} = 192 \text{ MiB/s}$$

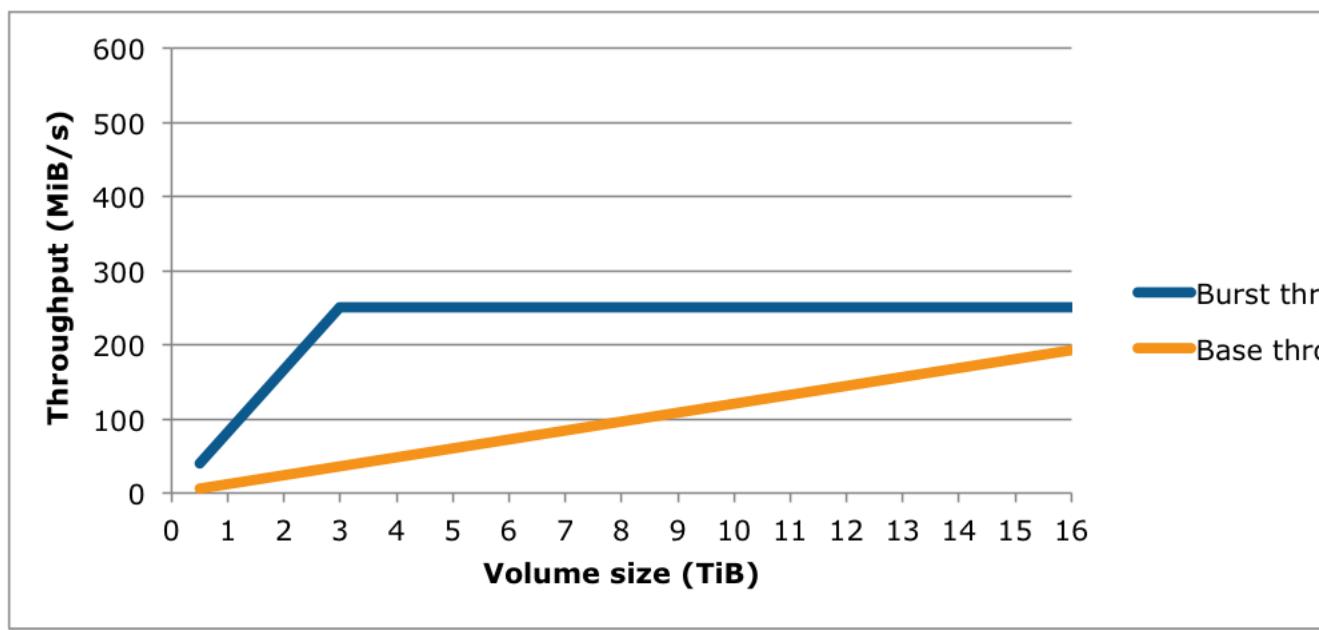
Burst throughput varies from 40 MiB/s to a cap of 250 MiB/s, which is reached at 3.125 TiB as follows:

$$\frac{80 \text{ MiB/s}}{3.125 \text{ TiB}} = 250 \text{ MiB/s}$$
$$\frac{1 \text{ TiB}}{1 \text{ TiB}}$$

The following table states the full range of base and burst throughput values for sc1:

Volume Size (TiB)	SC1 Base Throughput (MiB/s)	SC1 Burst Throughput (MiB/s)
0.5	6	40
1	12	80
2	24	160
3	36	240
3.125	37.5	250
4	48	250
5	60	250
6	72	250
7	84	250
8	96	250
9	108	250
10	120	250
11	132	250
12	144	250
13	156	250
14	168	250
15	180	250
16	192	250

The following diagram plots the table values:



Note

When you create a snapshot of a Cold HDD (sc1) volume, performance may drop as far as the volume's baseline value while the snapshot is in progress.

For information about using CloudWatch metrics and alarms to monitor your burst bucket balance, see [Monitoring the burst bucket balance for gp2, st1, and sc1 volumes \(p. 1037\)](#).

Magnetic (standard)

Magnetic volumes are backed by magnetic drives and are suited for workloads where data is accessed infrequently, and scenarios where low-cost storage for small volume sizes is important. These volumes deliver approximately 100 IOPS on average, with burst capability of up to hundreds of IOPS, and they can range in size from 1 GiB to 1 TiB.

Note

Magnetic is a previous generation volume type. For new applications, we recommend using one of the newer volume types. For more information, see [Previous Generation Volumes](#).

For information about using CloudWatch metrics and alarms to monitor your burst bucket balance, see [Monitoring the burst bucket balance for gp2, st1, and sc1 volumes \(p. 1037\)](#).

Performance considerations when using HDD volumes

For optimal throughput results using HDD volumes, plan your workloads with the following considerations in mind.

Throughput Optimized HDD vs. Cold HDD

The st1 and sc1 bucket sizes vary according to volume size, and a full bucket contains enough tokens for a full volume scan. However, larger st1 and sc1 volumes take longer for the volume scan to complete due to per-instance and per-volume throughput limits. Volumes attached to smaller instances are limited to the per-instance throughput rather than the st1 or sc1 throughput limits.

Both st1 and sc1 are designed for performance consistency of 90% of burst throughput 99% of the time. Non-compliant periods are approximately uniformly distributed, targeting 99% of expected total throughput each hour.

The following table shows ideal scan times for volumes of various size, assuming full buckets and sufficient instance throughput.

In general, scan times are expressed by this formula:

$$\frac{\text{Volume size}}{\text{Throughput}} = \frac{\text{Scan time}}{\text{Throughput}}$$

For example, taking the performance consistency guarantees and other optimizations into account, an **st1** customer with a 5-TiB volume can expect to complete a full volume scan in 2.91 to 3.27 hours.

$$\frac{5 \text{ TiB}}{500 \text{ MiB/s}} = \frac{5 \text{ TiB}}{0.00047684 \text{ TiB/s}} = 10,486 \text{ s} = 2.91 \text{ hours (optimal)}$$

$$2.91 \text{ hours} + \frac{2.91 \text{ hours}}{(0.90)(0.99)} = 3.27 \text{ hours (minimum expected)}$$

(0.90)(0.99) <-- From expected performance of 90% of burst 99% of the time

Similarly, an **sc1** customer with a 5-TiB volume can expect to complete a full volume scan in 5.83 to 6.54 hours.

$$\frac{5 \text{ TiB}}{0.000238418 \text{ TiB/s}} = 20972 \text{ s} = 5.83 \text{ hours (optimal)}$$

$$\frac{5.83 \text{ hours}}{(0.90)(0.99)} = 6.54 \text{ hours (minimum expected)}$$

Volume size (TiB)	ST1 scan time with burst (hours)*	SC1 scan time with burst (hours)*
1	1.17	3.64
2	1.17	3.64
3	1.75	3.64
4	2.33	4.66
5	2.91	5.83
6	3.50	6.99
7	4.08	8.16
8	4.66	9.32
9	5.24	10.49
10	5.83	11.65
11	6.41	12.82
12	6.99	13.98

Volume size (TiB)	ST1 scan time with burst (hours)*	SC1 scan time with burst (hours)*
13	7.57	15.15
14	8.16	16.31
15	8.74	17.48
16	9.32	18.64

* These scan times assume an average queue depth (rounded to the nearest whole number) of four or more when performing 1 MiB of sequential I/O.

Therefore if you have a throughput-oriented workload that needs to complete scans quickly (up to 500 MiB/s), or requires several full volume scans a day, use st1. If you are optimizing for cost, your data is relatively infrequently accessed, and you don't need more than 250 MiB/s of scanning performance, then use sc1.

Inefficiency of small read/writes on HDD

The performance model for st1 and sc1 volumes is optimized for sequential I/Os, favoring high-throughput workloads, offering acceptable performance on workloads with mixed IOPS and throughput, and discouraging workloads with small, random I/O.

For example, an I/O request of 1 MiB or less counts as a 1 MiB I/O credit. However, if the I/Os are sequential, they are merged into 1 MiB I/O blocks and count only as a 1 MiB I/O credit.

Limitations on per-instance throughput

Throughput for st1 and sc1 volumes is always determined by the smaller of the following:

- Throughput limits of the volume
- Throughput limits of the instance

As for all Amazon EBS volumes, we recommend that you select an appropriate EBS-optimized EC2 instance in order to avoid network bottlenecks. For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#).

Monitoring the burst bucket balance for gp2, st1, and sc1 volumes

You can monitor the burst-bucket level for gp2, st1, and sc1 volumes using the EBS BurstBalance metric available in Amazon CloudWatch. This metric shows the percentage of I/O credits (for gp2) or throughput credits (for st1 and sc1) remaining in the burst bucket. For more information about the `BurstBalance` metric and other metrics related to I/O, see [I/O characteristics and monitoring \(p. 1159\)](#). CloudWatch also allows you to set an alarm that notifies you when the `BurstBalance` value falls to a certain level. For more information, see [Creating Amazon CloudWatch Alarms](#).

Constraints on the size and configuration of an EBS volume

The size of an Amazon EBS volume is constrained by the physics and arithmetic of block data storage, as well as by the implementation decisions of operating system (OS) and file system designers. AWS imposes additional limits on volume size to safeguard the reliability of its services.

The following sections describe the most important factors that limit the usable size of an EBS volume and offer recommendations for configuring your EBS volumes.

Contents

- [Storage capacity \(p. 1038\)](#)
- [Service limitations \(p. 1038\)](#)
- [Partitioning schemes \(p. 1038\)](#)
- [Data block sizes \(p. 1039\)](#)

Storage capacity

The following table summarizes the theoretical and implemented storage capacities for the most commonly used file systems on Amazon EBS, assuming a 4,096 byte block size.

Partitioning scheme	Max addressable blocks	Theoretical max size (blocks × block size)	Ext4 implemented max size*	XFS implemented max size**	NTFS implemented max size	Max supported by EBS
MBR	2^{32}	2 TiB	2 TiB	2 TiB	2 TiB	2 TiB
GPT	2^{64}	64 ZiB	1 EiB = 1024^2 TiB (50 TiB certified on RHEL7)	500 TiB (certified on RHEL7)	256 TiB	16 TiB

* https://ext4.wiki.kernel.org/index.php/Ext4_Howto and <https://access.redhat.com/solutions/1532>

** <https://access.redhat.com/solutions/1532>

Service limitations

Amazon EBS abstracts the massively distributed storage of a data center into virtual hard disk drives. To an operating system installed on an EC2 instance, an attached EBS volume appears to be a physical hard disk drive containing 512-byte disk sectors. The OS manages the allocation of data blocks (or clusters) onto those virtual sectors through its storage management utilities. The allocation is in conformity with a volume partitioning scheme, such as master boot record (MBR) or GUID partition table (GPT), and within the capabilities of the installed file system (ext4, NTFS, and so on).

EBS is not aware of the data contained in its virtual disk sectors; it only ensures the integrity of the sectors. This means that AWS actions and OS actions are independent of each other. When you are selecting a volume size, be aware of the capabilities and limits of both, as in the following cases:

- EBS currently supports a maximum volume size of 16 TiB. This means that you can create an EBS volume as large as 16 TiB, but whether the OS recognizes all of that capacity depends on its own design characteristics and on how the volume is partitioned.
- Linux boot volumes may use either the MBR or GPT partitioning scheme. MBR supports boot volumes up to 2047 GiB (2 TiB - 1 GiB). GPT with GRUB 2 supports boot volumes 2 TiB or larger. If your Linux AMI uses MBR, your boot volume is limited to 2047 GiB, but your non-boot volumes do not have this limit. For more information, see [Making an Amazon EBS volume available for use on Linux \(p. 1046\)](#).

Partitioning schemes

Among other impacts, the partitioning scheme determines how many logical data blocks can be uniquely addressed in a single volume. For more information, see [Data block sizes \(p. 1039\)](#). The common

partitioning schemes in use are *master boot record* (MBR) and *GUID partition table* (GPT). The important differences between these schemes can be summarized as follows.

MBR

MBR uses a 32-bit data structure to store block addresses. This means that each data block is mapped with one of 2^{32} possible integers. The maximum addressable size of a volume is given by:

$$(2^{32} - 1) \times \text{Block size} = \text{Number of addressable blocks}$$

The block size for MBR volumes is conventionally limited to 512 bytes. Therefore:

$$(2^{32} - 1) \times 512 \text{ bytes} = 2 \text{ TiB} - 512 \text{ bytes}$$

Engineering workarounds to increase this 2-TiB limit for MBR volumes have not met with widespread industry adoption. Consequently, Linux and Windows never detect an MBR volume as being larger than 2 TiB even if AWS shows its size to be larger.

GPT

GPT uses a 64-bit data structure to store block addresses. This means that each data block is mapped with one of 2^{64} possible integers. The maximum addressable size of a volume is given by:

$$(2^{64} - 1) \times \text{Block size} = \text{Number of addressable blocks}$$

The block size for GPT volumes is commonly 4,096 bytes. Therefore:

$$\begin{aligned} (2^{64} - 1) \times 4,096 \text{ bytes} \\ = 2^{64} \times 4,096 \text{ bytes} - 1 \times 4,096 \text{ bytes} \\ = 2^{64} \times 2^{12} \text{ bytes} - 4,096 \text{ bytes} \\ = 2^{70} \times 2^6 \text{ bytes} - 4,096 \text{ bytes} \\ = 64 \text{ Zib} - 4,096 \text{ bytes} \end{aligned}$$

Real-world computer systems don't support anything close to this theoretical maximum. Implemented file-system size is currently limited to 50 TiB for ext4 and 256 TiB for NTFS—both of which exceed the 16-TiB limit imposed by AWS.

Data block sizes

Data storage on a modern hard drive is managed through *logical block addressing*, an abstraction layer that allows the operating system to read and write data in logical blocks without knowing much about the underlying hardware. The OS relies on the storage device to map the blocks to its physical sectors. EBS advertises 512-byte sectors to the operating system, which reads and writes data to disk using data blocks that are a multiple of the sector size.

The industry default size for logical data blocks is currently 4,096 bytes (4 KiB). Because certain workloads benefit from a smaller or larger block size, file systems support non-default block sizes that can be specified during formatting. Scenarios in which non-default block sizes should be used are outside the scope of this topic, but the choice of block size has consequences for the storage capacity of the volume. The following table shows storage capacity as a function of block size:

Block size	Max volume size
4 KiB (default)	16 TiB
8 KiB	32 TiB
16 KiB	64 TiB

Block size	Max volume size
32 KiB	128 TiB
64 KiB (maximum)	256 TiB

The EBS-imposed limit on volume size (16 TiB) is currently equal to the maximum size enabled by 4-KiB data blocks.

Creating an Amazon EBS volume

You can create an Amazon EBS volume and then attach it to any EC2 instance in the same Availability Zone. If you create an encrypted EBS volume, you can only attach it to supported instance types. For more information, see [Supported instance types \(p. 1122\)](#).

If you are creating a volume for a high-performance storage scenario, you should make sure to use a Provisioned IOPS SSD (io1 or io2) volume and attach it to an instance with enough bandwidth to support your application, such as an EBS-optimized instance or an instance with 10-Gigabit network connectivity. The same advice holds for Throughput Optimized HDD (st1) and Cold HDD (sc1) volumes. For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#).

Empty EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). However, storage blocks on volumes that were created from snapshots must be initialized (pulled down from Amazon S3 and written to the volume) before you can access the block. This preliminary action takes time and can cause a significant increase in the latency of an I/O operation the first time each block is accessed. Volume performance is achieved after all blocks have been downloaded and written to the volume. For most applications, amortizing this cost over the lifetime of the volume is acceptable. To avoid this initial performance hit in a production environment, you can force immediate initialization of the entire volume or enable fast snapshot restore. For more information, see [Initializing Amazon EBS volumes \(p. 1161\)](#).

Methods of creating a volume

- Create and attach EBS volumes when you launch instances by specifying a block device mapping. For more information, see [Launching an instance using the Launch Instance Wizard \(p. 499\)](#) and [Block device mapping \(p. 1211\)](#).
- Create an empty EBS volume and attach it to a running instance. For more information, see [Creating an empty volume \(p. 1040\)](#) below.
- Create an EBS volume from a previously created snapshot and attach it to a running instance. For more information, see [Creating a volume from a snapshot \(p. 1041\)](#) below.

Creating an empty volume

Empty volumes receive their maximum performance the moment that they are available and do not require initialization.

To create a empty EBS volume using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region in which you would like to create your volume. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can't. For more information, see [Resource locations \(p. 1222\)](#).
3. In the navigation pane, choose **ELASTIC BLOCK STORE, Volumes**.
4. Choose **Create Volume**.

5. For **Volume Type**, choose a volume type. For more information, see [Amazon EBS volume types \(p. 1023\)](#).
6. For **Size (GiB)**, type the size of the volume. For more information, see [Constraints on the size and configuration of an EBS volume \(p. 1037\)](#).
7. With a Provisioned IOPS SSD volume, for **IOPS**, type the maximum number of input/output operations per second (IOPS) that the volume should support.
8. For **Availability Zone**, choose the Availability Zone in which to create the volume. EBS volumes can only be attached to EC2 instances within the same Availability Zone.
9. (Optional) If the instance type supports EBS encryption and you want to encrypt the volume, select **Encrypt this volume** and choose a CMK. If encryption by default is enabled in this Region, EBS encryption is enabled and the default CMK for EBS encryption is chosen. You can choose a different CMK from **Master Key** or paste the full ARN of any key that you can access. For more information, see [Amazon EBS encryption \(p. 1121\)](#).
10. (Optional) Choose **Create additional tags** to add tags to the volume. For each tag, provide a tag key and a tag value. For more information, see [Tagging your Amazon EC2 resources \(p. 1229\)](#).
11. Choose **Create Volume**. The volume is ready for use when the volume status is **Available**.
12. To use your new volume, attach it to an instance, format it, and mount it. For more information, see [Attaching an Amazon EBS volume to an instance \(p. 1042\)](#).

To create an empty EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [create-volume](#) (AWS CLI)
- [New-EC2Volume](#) (AWS Tools for Windows PowerShell)

Creating a volume from a snapshot

Volumes created from snapshots load lazily in the background. This means that there is no need to wait for all of the data to transfer from Amazon S3 to your EBS volume before the instance can start accessing an attached volume and all its data. If your instance accesses data that hasn't yet been loaded, the volume immediately downloads the requested data from Amazon S3, and then continues loading the rest of the volume data in the background. Volume performance is achieved after all blocks are downloaded and written to the volume. To avoid the initial performance hit in a production environment, see [Initializing Amazon EBS volumes \(p. 1161\)](#).

New EBS volumes that are created from encrypted snapshots are automatically encrypted. You can also encrypt a volume on-the-fly while restoring it from an unencrypted snapshot. Encrypted volumes can only be attached to instance types that support EBS encryption. For more information, see [Supported instance types \(p. 1122\)](#).

Use the following procedure to create a volume from a snapshot.

To create an EBS volume from a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region that your snapshot is located in.

To use the snapshot to create a volume in a different region, copy your snapshot to the new Region and then use it to create a volume in that Region. For more information, see [Copying an Amazon EBS snapshot \(p. 1068\)](#).

3. In the navigation pane, choose **ELASTIC BLOCK STORE, Volumes**.
4. Choose **Create Volume**.

5. For **Volume Type**, choose a volume type. For more information, see [Amazon EBS volume types \(p. 1023\)](#).
6. For **Snapshot ID**, start typing the ID or description of the snapshot from which you are restoring the volume, and choose it from the list of suggested options.
7. (Optional) Select **Encrypt this volume** to change the encryption state of your volume. This is optional if [encryption by default \(p. 1124\)](#) is enabled. Select a CMK from **Master Key** to specify a CMK other than the default CMK for EBS encryption.
8. For **Size (GiB)**, type the size of the volume, or verify that the default size of the snapshot is adequate.

If you specify both a volume size and a snapshot, the size must be equal to or greater than the snapshot size. When you select a volume type and a snapshot, the minimum and maximum sizes for the volume are shown next to **Size**. For more information, see [Constraints on the size and configuration of an EBS volume \(p. 1037\)](#).

9. With a Provisioned IOPS SSD volume, for **IOPS**, type the maximum number of input/output operations per second (IOPS) that the volume should support.
10. For **Availability Zone**, choose the Availability Zone in which to create the volume. EBS volumes can only be attached to EC2 instances in the same Availability Zone.
11. (Optional) Choose **Create additional tags** to add tags to the volume. For each tag, provide a tag key and a tag value.
12. Choose **Create Volume**.
13. To use your new volume, attach it to an instance and mount it. For more information, see [Attaching an Amazon EBS volume to an instance \(p. 1042\)](#).
14. If you created a volume that is larger than the snapshot, you must extend the file system on the volume to take advantage of the extra space. For more information, see [Amazon EBS Elastic Volumes \(p. 1109\)](#).

To create an EBS volume from a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [create-volume](#) (AWS CLI)
- [New-EC2Volume](#) (AWS Tools for Windows PowerShell)

Attaching an Amazon EBS volume to an instance

You can attach an available EBS volume to one or more of your instances that is in the same Availability Zone as the volume.

Prerequisites

- Determine how many volumes you can attach to your instance. For more information, see [Instance volume limits \(p. 1209\)](#).
- Determine whether you can attach your volume to multiple instances and enable Multi-Attach. For more information, see [Attaching a volume to multiple instances with Amazon EBS Multi-Attach \(p. 1043\)](#).
- If a volume is encrypted, it can only be attached to an instance that supports Amazon EBS encryption. For more information, see [Supported instance types \(p. 1122\)](#).
- If a volume has an AWS Marketplace product code:
 - The volume can only be attached to a stopped instance.
 - You must be subscribed to the AWS Marketplace code that is on the volume.

- The configuration (instance type, operating system) of the instance must support that specific AWS Marketplace code. For example, you cannot take a volume from a Windows instance and attach it to a Linux instance.
- AWS Marketplace product codes are copied from the volume to the instance.

To attach an EBS volume to an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic Block Store, Volumes**.
3. Select an available volume and choose **Actions, Attach Volume**.
4. For **Instance**, start typing the name or ID of the instance. Select the instance from the list of options (only instances that are in the same Availability Zone as the volume are displayed).
5. For **Device**, you can keep the suggested device name, or type a different supported device name. For more information, see [Device naming on Linux instances \(p. 1210\)](#).
6. Choose **Attach**.
7. Connect to your instance and mount the volume. For more information, see [Making an Amazon EBS volume available for use on Linux \(p. 1046\)](#).

To attach an EBS volume to an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [attach-volume](#) (AWS CLI)
- [Add-EC2Volume](#) (AWS Tools for Windows PowerShell)

Attaching a volume to multiple instances with Amazon EBS Multi-Attach

Amazon EBS Multi-Attach enables you to attach a single Provisioned IOPS SSD (io1) volume to up to 16 Nitro-based instances that are in the same Availability Zone. You can attach multiple Multi-Attach enabled volumes to an instance or set of instances. Each instance to which the volume is attached has full read and write permission to the shared volume. Multi-Attach makes it easier for you to achieve higher application availability in clustered Linux applications that manage concurrent write operations.

Contents

- [Considerations and limitations \(p. 302\)](#)
- [Performance \(p. 1044\)](#)
- [Working with Multi-Attach \(p. 1044\)](#)
- [Monitoring \(p. 1046\)](#)
- [Pricing and billing \(p. 1046\)](#)

Considerations and limitations

- Multi-Attach enabled volumes can be attached to up to 16 Linux instances built on the [Nitro System \(p. 205\)](#) that are in the same Availability Zone. You can attach a volume that is Multi-Attach enabled to Windows instances, but the operating system does not recognize the data on the volume that is shared between the instances.
- Multi-Attach is supported exclusively on [Provisioned IOPS SSD \(io1\) volumes \(p. 1029\)](#). It is not supported on Provisioned IOPS SSD (io2) volumes.

- Multi-Attach is available only in the `us-east-1`, `us-west-2`, `eu-west-1`, and `ap-northeast-2` Regions.
- Multi-Attach enabled volumes do not support I/O fencing. I/O fencing protocols control write access in a shared storage environment to maintain data consistency. Your applications must provide write ordering for the attached instances to maintain data consistency.
- Multi-Attach enabled volumes can't be created as boot volumes.
- Multi-Attach enabled volumes can be attached to one block device mapping per instance.
- You can't enable or disable Multi-Attach after volume creation.
- You can't change the volume type, size, or Provisioned IOPS of a Multi-Attach enabled volume.
- Multi-Attach can't be enabled during instance launch using either the Amazon EC2 console or `RunInstances` API.
- Multi-Attach enabled volumes that have an issue at the Amazon EBS infrastructure layer are unavailable to all attached instances. Issues at the Amazon EC2 or networking layer might only impact some attached instances.

Performance

Each attached instance is able to drive its maximum IOPS performance up to the volume's maximum provisioned performance. However, the aggregate performance of all of the attached instances can't exceed the volume's maximum provisioned performance. If the attached instances' demand for IOPS is higher than the volume's Provisioned IOPS, the volume will not exceed its provisioned performance.

For example, say you create an `io1` Multi-Attach enabled volume with 50,000 Provisioned IOPS and you attach it to an `m5.8xlarge` instance and a `c5.12xlarge` instance. The `m5.8xlarge` and `c5.12xlarge` instances support a maximum of 30,000 and 40,000 IOPS respectively. Each instance can drive its maximum IOPS as it is less than the volume's Provisioned IOPS of 50,000. However, if both instances drive I/O to the volume simultaneously, their combined IOPS can't exceed the volume's provisioned performance of 50,000 IOPS. The volume will not exceed 50,000 IOPS.

To achieve consistent performance, it is best practice to balance I/O driven from attached instances across the sectors of a Multi-Attach enabled volume.

Working with Multi-Attach

Multi-Attach enabled volumes can be managed in much the same way that you would manage any other Amazon EBS volume. However, in order to use the Multi-Attach functionality, you must enable it for the volume. When you create a new volume, Multi-Attach is disabled by default.

Contents

- [Enabling Multi-Attach \(p. 1044\)](#)
- [Attaching a volume to instances \(p. 1045\)](#)
- [Deleting on termination \(p. 1045\)](#)

Enabling Multi-Attach

You can enable Multi-Attach for an Amazon EBS volume during creation only.

Use one of the following methods to enable Multi-Attach for an Amazon EBS volume during creation.

Console

To enable Multi-Attach during volume creation

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Volumes**.
3. Choose **Create Volume**.
4. For **Volume Type**, choose **Provisioned IOPS SSD (io1)**.
5. For **Size** and **IOPS**, choose the required volume size and the number of IOPS to provision.
6. For **Availability Zone**, choose the same Availability Zone that the instances are in.
7. For **Multi-Attach**, choose **Enable**.
8. Choose **Create Volume**.

Command line

To enable Multi-Attach during volume creation

Use the [create-volume](#) command and specify the `--multi-attach-enabled` parameter.

```
$ aws ec2 create-volume --volume-type io1 --multi-attach-enabled --size 100 --iops 2000  
--region us-west-2 --availability-zone us-west-2b
```

Attaching a volume to instances

You attach a Multi-Attach enabled volume to an instance in the same way that you attach any other EBS volume. For more information, see [Attaching an Amazon EBS volume to an instance \(p. 1042\)](#).

Deleting on termination

Multi-Attach enabled volumes are deleted on instance termination if the last attached instance is terminated and if that instance is configured to delete the volume on termination. If the volume is attached to multiple instances that have different delete on termination settings in their volume block device mappings, the last attached instance's block device mapping setting determines the delete on termination behavior.

To ensure predictable delete on termination behavior, enable or disable delete on termination for all of the instances to which the volume is attached.

By default, when a volume is attached to an instance the delete on termination setting for the block device mapping is set to false. If you want to turn on delete on termination for a Multi-Attach enabled volume, modify the block device mapping.

If you want the volume to be deleted when the attached instances are terminated, enable delete on termination in the block device mapping for all of the attached instances. If you want to retain the volume after the attached instances have been terminated, disable delete on termination in the block device mapping for all of the attached instances. For more information, see [Preserving Amazon EBS volumes on instance termination \(p. 610\)](#).

You can modify an instance's delete on termination setting at launch or after it has launched. If you enable or disable delete on termination during instance launch, the settings apply only to volumes that are attached at launch. If you attach a volume to an instance after launch, you must explicitly set the delete on termination behavior for that volume.

You can modify an instance's delete on termination setting using the command line tools only.

To modify the delete on termination setting for an existing instance

Use the [modify-instance-attribute](#) command and specify the `DeleteOnTermination` attribute in the `--block-device-mappings` option.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --block-device-mappings file://mapping.json
```

Specify the following in `mapping.json`.

```
[  
  {  
    "DeviceName": "/dev/sdf",  
    "Ebs": {  
      "DeleteOnTermination": true/false  
    }  
}
```

Monitoring

You can monitor a Multi-Attach enabled volume using the CloudWatch Metrics for Amazon EBS volumes. For more information, see [Amazon CloudWatch metrics for Amazon EBS \(p. 1172\)](#).

Data is aggregated across all of the attached instances. You can't monitor metrics for individual attached instances.

Pricing and billing

There are no additional charges for using Amazon EBS Multi-Attach. You are billed the standard charges that apply to Provisioned IOPS SSD (io1) volumes. For more information, see [Amazon EBS pricing](#).

Making an Amazon EBS volume available for use on Linux

After you attach an Amazon EBS volume to your instance, it is exposed as a block device. You can format the volume with any file system and then mount it. After you make the EBS volume available for use, you can access it in the same ways that you access any other volume. Any data written to this file system is written to the EBS volume and is transparent to applications using the device.

You can take snapshots of your EBS volume for backup purposes or to use as a baseline when you create another volume. For more information, see [Amazon EBS snapshots \(p. 1060\)](#).

You can get directions for volumes on a Windows instance from [Making a Volume Available for Use on Windows](#) in the *Amazon EC2 User Guide for Windows Instances*.

Format and mount an attached volume

Suppose that you have an EC2 instance with an EBS volume for the root device, `/dev/xvda`, and that you have just attached an empty EBS volume to the instance using `/dev/sdf`. Use the following procedure to make the newly attached volume available for use.

To format and mount an EBS volume on Linux

1. Connect to your instance using SSH. For more information, see [Connect to your Linux instance \(p. 563\)](#).
2. The device could be attached to the instance with a different device name than you specified in the block device mapping. For more information, see [Device naming on Linux instances \(p. 1210\)](#). Use the `lsblk` command to view your available disk devices and their mount points (if applicable) to help you determine the correct device name to use. The output of `lsblk` removes the `/dev/` prefix from full device paths.

The following is example output for an instance built on the [Nitro System \(p. 205\)](#), which exposes EBS volumes as NVMe block devices. The root device is `/dev/nvme0n1`. The attached volume is `/dev/nvme1n1`, which is not yet mounted.

```
[ec2-user ~]$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme1n1    259:0    0   10G  0 disk
nvme0n1    259:1    0   8G  0 disk
-nvme0n1p1  259:2    0   8G  0 part /
-nvme0n1p128 259:3    0   1M  0 part
```

The following is example output for a T2 instance. The root device is `/dev/xvda`. The attached volume is `/dev/xvdf`, which is not yet mounted.

```
[ec2-user ~]$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda     202:0    0   8G  0 disk
-xvda1   202:1    0   8G  0 part /
xvdf     202:80   0  10G  0 disk
```

3. Determine whether there is a file system on the volume. New volumes are raw block devices, and you must create a file system on them before you can mount and use them. Volumes that were created from snapshots likely have a file system on them already; if you create a new file system on top of an existing file system, the operation overwrites your data.

Use the `file -s` command to get information about a device, such as its file system type. If the output shows simply data, as in the following example output, there is no file system on the device and you must create one.

```
[ec2-user ~]$ sudo file -s /dev/xvdf
/dev/xvdf: data
```

If the device has a file system, the command shows information about the file system type. For example, the following output shows a root device with the XFS file system.

```
[ec2-user ~]$ sudo file -s /dev/xvda1
/dev/xvda1: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)
```

4. (Conditional) If you discovered that there is a file system on the device in the previous step, skip this step. If you have an empty volume, use the `mkfs -t` command to create a file system on the volume.

Warning

Do not use this command if you're mounting a volume that already has data on it (for example, a volume that was created from a snapshot). Otherwise, you'll format the volume and delete the existing data.

```
[ec2-user ~]$ sudo mkfs -t xfs /dev/xvdf
```

If you get an error that `mkfs.xfs` is not found, use the following command to install the XFS tools and then repeat the previous command:

```
[ec2-user ~]$ sudo yum install xfsprogs
```

5. Use the `mkdir` command to create a mount point directory for the volume. The mount point is where the volume is located in the file system tree and where you read and write files to after you mount the volume. The following example creates a directory named `/data`.

```
[ec2-user ~]$ sudo mkdir /data
```

6. Use the following command to mount the volume at the directory you created in the previous step.

```
[ec2-user ~]$ sudo mount /dev/xvdf /data
```

7. Review the file permissions of your new volume mount to make sure that your users and applications can write to the volume. For more information about file permissions, see [File security at The Linux Documentation Project](#).
8. The mount point is not automatically preserved after rebooting your instance. To automatically mount this EBS volume after reboot, see [Automatically mount an attached volume after reboot \(p. 1048\)](#).

Automatically mount an attached volume after reboot

To mount an attached EBS volume on every system reboot, add an entry for the device to the `/etc/fstab` file.

You can use the device name, such as `/dev/xvdf`, in `/etc/fstab`, but we recommend using the device's 128-bit universally unique identifier (UUID) instead. Device names can change, but the UUID persists throughout the life of the partition. By using the UUID, you reduce the chances that the system becomes unbootable after a hardware reconfiguration. For more information, see [Identifying the EBS device \(p. 1137\)](#).

To mount an attached volume automatically after reboot

1. (Optional) Create a backup of your `/etc/fstab` file that you can use if you accidentally destroy or delete this file while editing it.

```
[ec2-user ~]$ sudo cp /etc/fstab /etc/fstab.orig
```

2. Use the `blkid` command to find the UUID of the device.

```
[ec2-user ~]$ sudo blkid  
/dev/xvda1: LABEL="/" UUID="ca774df7-756d-4261-a3f1-76038323e572" TYPE="xfs"  
PARTLABEL="Linux" PARTUUID="02dc367-e87c-4f2e-9a72-a3cf8f299c10"  
/dev/xvdf: UUID="aebf131c-6957-451e-8d34-ec978d9581ae" TYPE="xfs"
```

For Ubuntu 18.04 use the `lsblk` command.

```
[ec2-user ~]$ sudo lsblk -o +UUID
```

3. Open the `/etc/fstab` file using any text editor, such as `nano` or `vim`.

```
[ec2-user ~]$ sudo vim /etc/fstab
```

4. Add the following entry to `/etc/fstab` to mount the device at the specified mount point. The fields are the UUID value returned by `blkid` (or `lsblk` for Ubuntu 18.04), the mount point, the file system, and the recommended file system mount options. For more information, see the manual page for `fstab` (run `man fstab`).

```
UUID=aebf131c-6957-451e-8d34-ec978d9581ae /data xfs defaults,nofail 0 2
```

Note

If you ever boot your instance without this volume attached (for example, after moving the volume to another instance), the `nofail` mount option enables the instance to boot even if there are errors mounting the volume. Debian derivatives, including Ubuntu versions earlier than 16.04, must also add the `nobootwait` mount option.

5. To verify that your entry works, run the following commands to unmount the device and then mount all file systems in /etc/fstab. If there are no errors, the /etc/fstab file is OK and your file system will mount automatically after it is rebooted.

```
[ec2-user ~]$ sudo umount /data
[ec2-user ~]$ sudo mount -a
```

If you receive an error message, address the errors in the file.

Warning

Errors in the /etc/fstab file can render a system unbootable. Do not shut down a system that has errors in the /etc/fstab file.

If you are unsure how to correct errors in /etc/fstab and you created a backup file in the first step of this procedure, you can restore from your backup file using the following command.

```
[ec2-user ~]$ sudo mv /etc/fstab.orig /etc/fstab
```

Viewing information about an Amazon EBS volume

You can view descriptive information about your EBS volumes. For example, you can view information about all volumes in a specific Region or view detailed information about a single volume, including its size, volume type, whether the volume is encrypted, which master key was used to encrypt the volume, and the specific instance to which the volume is attached.

You can get additional information about your EBS volumes, such as how much disk space is available, from the operating system on the instance.

Viewing volume information

To view information about an EBS volume using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. (Optional) Use the filter options in the search field to display only the volumes that interest you. For example, if you know the instance ID, choose **Instance ID** from the search field menu, and then choose the instance ID from the list provided. To remove a filter, choose it again.
4. Select the volume.
5. In the details pane, you can inspect the information provided about the volume. **Attachment information** shows the instance ID this volume is attached to and the device name under which it is attached.
6. (Optional) Choose the **Attachment information** link to view additional details about the instance.

To view the EBS volumes that are attached to an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. In the **Storage** tab, view the information provided about root and block devices.
5. (Optional) Choose a link in the **Volume ID** column to view additional details for the volume.

To view information about an EBS volume using the command line

You can use one of the following commands to view volume attributes. For more information, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-volumes](#) (AWS CLI)
- [Get-EC2Volume](#) (AWS Tools for Windows PowerShell)

Volume state

Volume state describes the availability of an Amazon EBS volume. You can view the volume state in the **State** column on the **Volumes** page in the console, or by using the [describe-volumes](#) AWS CLI command.

The possible volume states are:

creating

The volume is being created.

available

The volume is not attached to an instance.

in-use

The volume is attached to an instance.

deleting

The volume is being deleted.

deleted

The volume is deleted.

error

The underlying hardware related to your EBS volume has failed, and the data associated with the volume is unrecoverable. For information about how to restore the volume or recover the data on the volume, see [My EBS volume has a status of "error"](#).

Viewing volume metrics

You can get additional information about your EBS volumes from Amazon CloudWatch. For more information, see [Amazon CloudWatch metrics for Amazon EBS \(p. 1172\)](#).

Viewing free disk space

You can get additional information about your EBS volumes, such as how much disk space is available, from the Linux operating system on the instance. For example, use the following command:

```
[ec2-user ~]$ df -hT /dev/xvda1
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/xvda1      xfs       8.0G  1.2G  6.9G  15%  /
```

Replacing an Amazon EBS volume using a previous snapshot

Amazon EBS snapshots are the preferred backup tool on Amazon EC2 due to their speed, convenience, and cost. When creating a volume from a snapshot, you recreate its state at a specific point in the past with all data intact. By attaching a volume created from a snapshot to an instance, you can duplicate data across Regions, create test environments, replace a damaged or corrupted production volume in its entirety, or retrieve specific files and directories and transfer them to another attached volume. For more information, see [Amazon EBS snapshots \(p. 1060\)](#).

You can use the following procedure to replace an EBS volume with another volume created from a previous snapshot of that volume. You must detach the current volume and then attach the new volume.

Note that EBS volumes can only be attached to EC2 instances in the same Availability Zone.

To replace a volume

1. Create a volume from the snapshot and write down the ID of the new volume. For more information, see [Creating a volume from a snapshot \(p. 1041\)](#).
2. On the volumes page, select the check box for the volume to replace. On the **Description** tab, find **Attachment information** and write down the device name of the volume (for example, /dev/sda1 or /dev/xvda for a root volume, or /dev/sdb or xvdb) and the ID of the instance.
3. (Optional) Before you can detach the root volume of an instance, you must stop the instance. If you are not replacing the root volume, you can continue to the next step without stopping the instance. Otherwise, to stop the instance, from **Attachment information**, hover over the instance ID, right-click, and open the instance in a new browser tab. Choose **Actions, Instance State, Stop**. Leave the tab with the instances page open and return to the browser tab with the volumes page.
4. With the volume still selected, choose **Actions, Detach Volume**. When prompted for confirmation, choose **Yes, Detach**. Clear the check box for this volume.
5. Select the check box for the new volume that you created in step 1. Choose **Actions, Attach Volume**. Enter the instance ID and device name that you wrote down in step 2, and then choose **Attach**.
6. (Optional) If you stopped the instance, you must restart it. Return to the browser tab with the instances page and choose **Actions, Instance State, Start**.
7. Connect to your instance and mount the volume. For more information, see [Making an Amazon EBS volume available for use on Linux \(p. 1046\)](#).

Monitoring the status of your volumes

Amazon Web Services (AWS) automatically provides data that you can use to monitor your Amazon Elastic Block Store (Amazon EBS) volumes.

Contents

- [EBS volume status checks \(p. 1051\)](#)
- [EBS volume events \(p. 1053\)](#)
- [Working with an impaired volume \(p. 1055\)](#)
- [Working with the Auto-Enabled IO volume attribute \(p. 1057\)](#)

For additional monitoring information, see [Amazon CloudWatch metrics for Amazon EBS \(p. 1172\)](#) and [Amazon CloudWatch Events for Amazon EBS \(p. 1178\)](#).

EBS volume status checks

Volume status checks enable you to better understand, track, and manage potential inconsistencies in the data on an Amazon EBS volume. They are designed to provide you with the information that you need to determine whether your Amazon EBS volumes are impaired, and to help you control how a potentially inconsistent volume is handled.

Volume status checks are automated tests that run every 5 minutes and return a pass or fail status. If all checks pass, the status of the volume is `ok`. If a check fails, the status of the volume is `impaired`. If the status is `insufficient-data`, the checks may still be in progress on the volume. You can view the results of volume status checks to identify any impaired volumes and take any necessary actions.

When Amazon EBS determines that a volume's data is potentially inconsistent, the default is that it disables I/O to the volume from any attached EC2 instances, which helps to prevent data corruption.

After I/O is disabled, the next volume status check fails, and the volume status is **impaired**. In addition, you'll see an event that lets you know that I/O is disabled, and that you can resolve the impaired status of the volume by enabling I/O to the volume. We wait until you enable I/O to give you the opportunity to decide whether to continue to let your instances use the volume, or to run a consistency check using a command, such as `fsck`, before doing so.

Note

Volume status is based on the volume status checks, and does not reflect the volume state. Therefore, volume status does not indicate volumes in the **error** state (for example, when a volume is incapable of accepting I/O.) For information about volume states, see [Volume state \(p. 1050\)](#).

If the consistency of a particular volume is not a concern, and you'd prefer that the volume be made available immediately if it's impaired, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the **Auto-Enable IO** volume attribute (`autoEnableIO` in the API), the volume status check continues to pass. In addition, you'll see an event that lets you know that the volume was determined to be potentially inconsistent, but that its I/O was automatically enabled. This enables you to check the volume's consistency or replace it at a later time.

The I/O performance status check compares actual volume performance to the expected performance of a volume and alerts you if the volume is performing below expectations. This status check is only available for Provisioned IOPS SSD (`io1` and `io2`) volumes that are attached to an instance. It is not valid for General Purpose SSD (`gp2`), Throughput Optimized HDD (`st1`), Cold HDD (`sc1`), or Magnetic (standard) volumes. The I/O performance status check is performed once every minute and CloudWatch collects this data every 5 minutes, so it may take up to 5 minutes from the moment you attach an `io1` or `io2` volume to an instance for this check to report the I/O performance status.

Important

While initializing `io1` and `io2` volumes that were restored from snapshots, the performance of the volume may drop below 50 percent of its expected level, which causes the volume to display a warning state in the **I/O Performance** status check. This is expected, and you can ignore the warning state on `io1` and `io2` volumes while you are initializing them. For more information, see [Initializing Amazon EBS volumes \(p. 1161\)](#).

The following table lists statuses for Amazon EBS volumes.

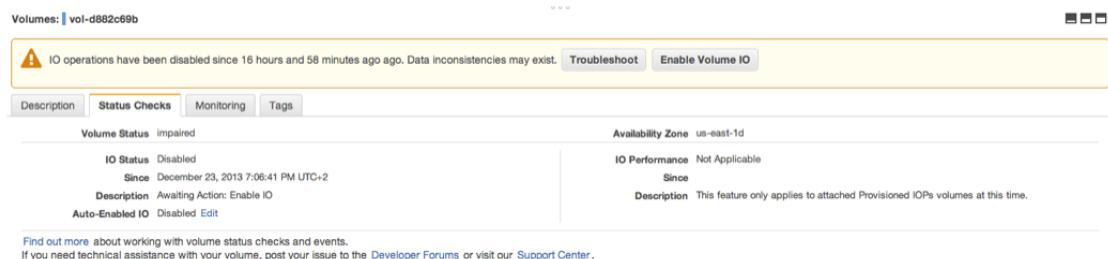
Volume status	I/O enabled status	I/O performance status (only available for Provisioned IOPS volumes)
ok	Enabled (I/O Enabled or I/O Auto-Enabled)	Normal (Volume performance is as expected)
warning	Enabled (I/O Enabled or I/O Auto-Enabled)	Degraded (Volume performance is below expectations) Severely Degraded (Volume performance is well below expectations)
impaired	Enabled (I/O Enabled or I/O Auto-Enabled) Disabled (Volume is offline and pending recovery, or is waiting for the user to enable I/O)	Stalled (Volume performance is severely impacted) Not Available (Unable to determine I/O performance because I/O is disabled)
insufficient-data	Enabled (I/O Enabled or I/O Auto-Enabled)	Insufficient Data

Volume status	I/O enabled status	I/O performance status (only available for Provisioned IOPS volumes)
	Insufficient Data	

To view and work with status checks, you can use the Amazon EC2 console, the API, or the command line interface.

To view status checks in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**. The **Volume Status** column displays the operational status of each volume.
3. To view the status details of a volume, select the volume and choose **Status Checks**.



4. If you have a volume with a failed status check (status is **impaired**), see [Working with an impaired volume \(p. 1055\)](#).

Alternatively, you can choose **Events** in the navigator to view all the events for your instances and volumes. For more information, see [EBS volume events \(p. 1053\)](#).

To view volume status information with the command line

You can use one of the following commands to view the status of your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- **describe-volume-status** (AWS CLI)
- **Get-EC2VolumeStatus** (AWS Tools for Windows PowerShell)

EBS volume events

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure.

To automatically enable I/O on a volume with potential data inconsistencies, change the setting of the **Auto-Enabled IO** volume attribute (`autoEnableIO` in the API). For more information about changing this attribute, see [Working with an impaired volume \(p. 1055\)](#).

Each event includes a start time that indicates the time at which the event occurred, and a duration that indicates how long I/O for the volume was disabled. The end time is added to the event when I/O for the volume is enabled.

Volume status events include one of the following descriptions:

Awaiting Action: Enable IO

Volume data is potentially inconsistent. I/O is disabled for the volume until you explicitly enable it. The event description changes to **IO Enabled** after you explicitly enable I/O.

IO Enabled

I/O operations were explicitly enabled for this volume.

IO Auto-Enabled

I/O operations were automatically enabled on this volume after an event occurred. We recommend that you check for data inconsistencies before continuing to use the data.

Normal

For `io1` and `io2` volumes only. Volume performance is as expected.

Degraded

For `io1` and `io2` volumes only. Volume performance is below expectations.

Severely Degraded

For `io1` and `io2` volumes only. Volume performance is well below expectations.

Stalled

For `io1` and `io2` volumes only. Volume performance is severely impacted.

You can view events for your volumes using the Amazon EC2 console, the API, or the command line interface.

To view events for your volumes in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**. All instances and volumes that have events are listed.
3. You can filter by volume to view only volume status. You can also filter on specific status types.
4. Select a volume to view its specific event.

Resource Name	Resource Type	Resource Id	Availability Zone	Event Type	Event Description	Event Status	Start Time	Duration	Event Progress
volume	volume	vol-0381c540	us-east-1d	potential-data-i...	Awaiting Action...	⚠ Awaiting A...	December 23, 2013	30 days, 15 ho...	IO Disabled
volume	volume	vol-3682c675	us-east-1d	potential-data-i...	Awaiting Action...	⚠ Awaiting A...	December 23, 2013	30 days, 15 ho...	IO Disabled

Event: vol-3682c675

⚠ IO operations have been disabled since 30 days, 15 hours and 22 minutes ago. Data inconsistencies may exist. [Enable Volume IO](#)

Availability Zone	us-east-1d
Event Type	potential-data-inconsistency
Event Status	Awaiting Action: Enable IO
IO Status	IO Disabled
Attached To	i-93aae4ea
Start Time	December 23, 2013 7:09:20 PM UTC+2
End Time	

Find out more about [monitoring volume events](#).

If you have a volume where I/O is disabled, see [Working with an impaired volume \(p. 1055\)](#). If you have a volume where I/O performance is below normal, this might be a temporary condition due to an action you have taken (for example, creating a snapshot of a volume during peak usage, running the volume on an instance that cannot support the I/O bandwidth required, accessing data on the volume for the first time, etc.).

To view events for your volumes with the command line

You can use one of the following commands to view event information for your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-volume-status](#) (AWS CLI)
- [Get-EC2VolumeStatus](#) (AWS Tools for Windows PowerShell)

Working with an impaired volume

Use the following options if a volume is impaired because the volume's data is potentially inconsistent.

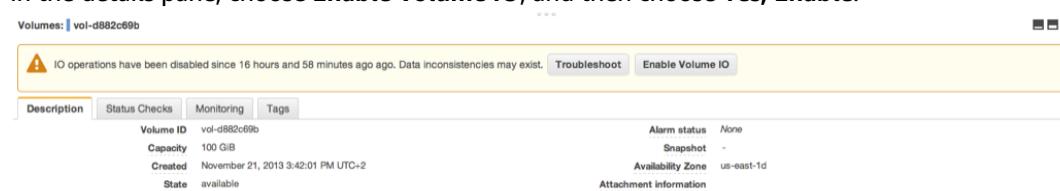
Options

- [Option 1: Perform a consistency check on the volume attached to its instance \(p. 1055\)](#)
- [Option 2: Perform a consistency check on the volume using another instance \(p. 1056\)](#)
- [Option 3: Delete the volume if you no longer need it \(p. 1056\)](#)

Option 1: Perform a consistency check on the volume attached to its instance

The simplest option is to enable I/O and then perform a data consistency check on the volume while the volume is still attached to its Amazon EC2 instance.

To perform a consistency check on an attached volume

1. Stop any applications from using the volume.
2. Enable I/O on the volume.
 - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 - b. In the navigation pane, choose **Volumes**.
 - c. Select the volume on which to enable I/O operations.
 - d. In the details pane, choose **Enable Volume IO**, and then choose **Yes, Enable**.

The screenshot shows the 'Volumes' page in the AWS Management Console. A yellow warning box at the top states: 'IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistencies may exist.' It includes 'Troubleshoot' and 'Enable Volume IO' buttons. Below the box, there are tabs for 'Description', 'Status Checks', 'Monitoring', and 'Tags'. The 'Description' tab is selected, showing detailed information for a volume with Volume ID 'vol-d882c69b'. The volume has a capacity of 100 GiB, was created on November 21, 2013, at 3:42:01 PM UTC-2, and is currently available. It is an 'io1' volume type and has product codes. The 'Status Checks' tab shows a warning message: 'IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistencies may exist.' The 'Monitoring' and 'Tags' tabs are also visible.

 3. Check the data on the volume.
 - a. Run the **fsck** command.
 - b. (Optional) Review any available application or system logs for relevant error messages.
 - c. If the volume has been impaired for more than 20 minutes, you can contact the AWS Support Center. Choose **Troubleshoot**, and then in the **Troubleshoot Status Checks** dialog box, choose **Contact Support** to submit a support case.

To enable I/O for a volume with the command line

You can use one of the following commands to view event information for your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [enable-volume-io](#) (AWS CLI)

- [Enable-EC2VolumeIO](#) (AWS Tools for Windows PowerShell)

Option 2: Perform a consistency check on the volume using another instance

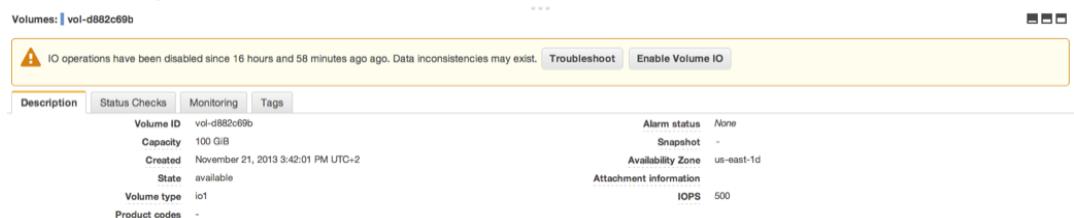
Use the following procedure to check the volume outside your production environment.

Important

This procedure may cause the loss of write I/Os that were suspended when volume I/O was disabled.

To perform a consistency check on a volume in isolation

1. Stop any applications from using the volume.
2. Detach the volume from the instance.
 - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 - b. In the navigation pane, choose **Volumes**.
 - c. Select the volume to detach.
 - d. Choose **Actions, Force Detach Volume**. You'll be prompted for confirmation.
3. Enable I/O on the volume.
 - a. In the navigation pane, choose **Volumes**.
 - b. Select the volume that you detached in the previous step.
 - c. In the details pane, choose **Enable Volume IO**, and then choose **Yes, Enable**.



4. Attach the volume to another instance. For more information, see [Launch your instance \(p. 497\)](#) and [Attaching an Amazon EBS volume to an instance \(p. 1042\)](#).
5. Check the data on the volume.
 - a. Run the **fsck** command.
 - b. (Optional) Review any available application or system logs for relevant error messages.
 - c. If the volume has been impaired for more than 20 minutes, you can contact the AWS Support Center. Choose **Troubleshoot**, and then in the troubleshooting dialog box, choose **Contact Support** to submit a support case.

To enable I/O for a volume with the command line

You can use one of the following commands to view event information for your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [enable-volume-io](#) (AWS CLI)
- [Enable-EC2VolumeIO](#) (AWS Tools for Windows PowerShell)

Option 3: Delete the volume if you no longer need it

If you want to remove the volume from your environment, simply delete it. For information about deleting a volume, see [Deleting an Amazon EBS volume \(p. 1060\)](#).

If you have a recent snapshot that backs up the data on the volume, you can create a new volume from the snapshot. For more information, see [Creating a volume from a snapshot \(p. 1041\)](#).

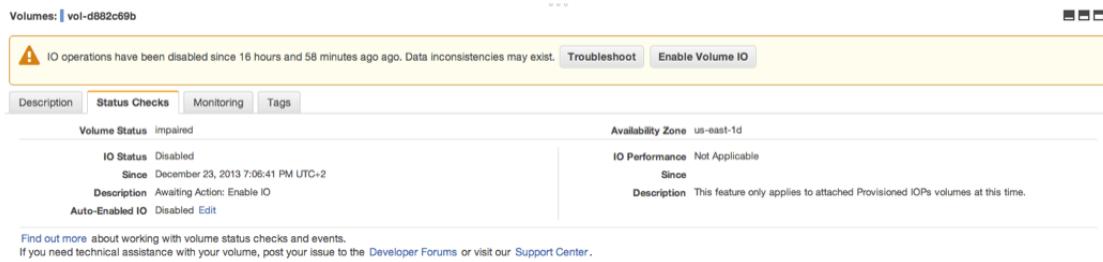
Working with the Auto-Enabled IO volume attribute

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure. If the consistency of a particular volume is not a concern, and you prefer that the volume be made available immediately if it's **impaired**, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the **Auto-Enabled IO** volume attribute (`autoEnableIO` in the API), I/O between the volume and the instance is automatically re-enabled and the volume's status check will pass. In addition, you'll see an event that lets you know that the volume was in a potentially inconsistent state, but that its I/O was automatically enabled. When this event occurs, you should check the volume's consistency and replace it if necessary. For more information, see [EBS volume events \(p. 1053\)](#).

This procedure explains how to view and modify the **Auto-Enabled IO** attribute of a volume.

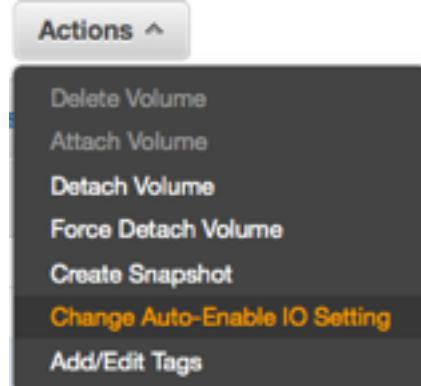
To view the Auto-Enabled IO attribute of a volume in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select the volume and choose **Status Checks**. **Auto-Enabled IO** displays the current setting (**Enabled** or **Disabled**) for your volume.

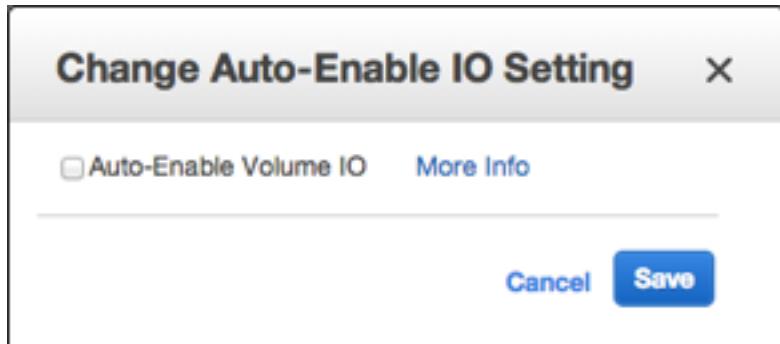


To modify the Auto-Enabled IO attribute of a volume in the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select the volume and choose **Actions, Change Auto-Enable IO Setting**. Alternatively, choose the **Status Checks** tab, and for **Auto-Enabled IO**, choose **Edit**.



4. Select the **Auto-Enable Volume IO** check box to automatically enable I/O for an impaired volume. To disable the feature, clear the check box.



5. Choose **Save**.

To view or modify the autoEnableIO attribute of a volume with the command line

You can use one of the following commands to view the `autoEnableIO` attribute of your Amazon EBS volumes. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-volume-attribute](#) (AWS CLI)
- [Get-EC2VolumeAttribute](#) (AWS Tools for Windows PowerShell)

To modify the `autoEnableIO` attribute of a volume, you can use one of the commands below.

- [modify-volume-attribute](#) (AWS CLI)
- [Edit-EC2VolumeAttribute](#) (AWS Tools for Windows PowerShell)

Detaching an Amazon EBS volume from a Linux instance

You can detach an Amazon EBS volume from an instance explicitly or by terminating the instance. However, if the instance is running, you must first unmount the volume from the instance.

If an EBS volume is the root device of an instance, you must stop the instance before you can detach the volume.

When a volume with an AWS Marketplace product code is detached from an instance, the product code is no longer associated with the instance.

Important

After you detach a volume, you are still charged for volume storage as long as the storage amount exceeds the limit of the AWS Free Tier. You must delete a volume to avoid incurring further charges. For more information, see [Deleting an Amazon EBS volume \(p. 1060\)](#).

This example unmounts the volume and then explicitly detaches it from the instance. This is useful when you want to terminate an instance or attach a volume to a different instance. To verify that the volume is no longer attached to the instance, see [Viewing information about an Amazon EBS volume \(p. 1049\)](#).

You can reattach a volume that you detached (without unmounting it), but it might not get the same mount point. If there were writes to the volume in progress when it was detached, the data on the volume might be out of sync.

You can get directions for volumes on a Windows instance from [Detaching a volume from a Windows instance](#) in the *Amazon EC2 User Guide for Windows Instances*.

To detach an EBS volume using the console

1. From your Linux instance, use the following command to unmount the `/dev/sdh` device.

```
[ec2-user ~]$ umount -d /dev/sdh
```

2. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
3. In the navigation pane, choose **Volumes**.
4. Select a volume and choose **Actions, Detach Volume**.
5. In the confirmation dialog box, choose **Yes, Detach**.

To detach an EBS volume from an instance using the command line

After unmounting the volume, you can use one of the following commands to detach it. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [detach-volume](#) (AWS CLI)
- [Dismount-EC2Volume](#) (AWS Tools for Windows PowerShell)

Troubleshooting

The following are common problems encountered when detaching volumes, and how to resolve them.

Note

To guard against the possibility of data loss, take a snapshot of your volume before attempting to unmount it. Forced detachment of a stuck volume can cause damage to the file system or the data it contains or an inability to attach a new volume using the same device name, unless you reboot the instance.

- If you encounter problems while detaching a volume through the Amazon EC2 console, it may be helpful to use the [describe-volumes](#) CLI command to diagnose the issue. For more information, see [describe-volumes](#).
- If your volume stays in the detaching state, you can force the detachment by choosing **Force Detach**. Use this option only as a last resort to detach a volume from a failed instance, or if you are detaching a volume with the intention of deleting it. The instance doesn't get an opportunity to flush file system caches or file system metadata. If you use this option, you must perform the file system check and repair procedures.
- If you've tried to force the volume to detach multiple times over several minutes and it stays in the detaching state, you can post a request for help to the [Amazon EC2 forum](#). To help expedite a resolution, include the volume ID and describe the steps that you've already taken.
- When you attempt to detach a volume that is still mounted, the volume can become stuck in the **busy** state while it is trying to detach. The following output from [describe-volumes](#) shows an example of this condition:

```
aws ec2 describe-volumes --region us-west-2 --volume-ids vol-1234abcd
{
    "Volumes": [
        {
            "AvailabilityZone": "us-west-2b",
            "Attachments": [
                {
                    "AttachTime": "2016-07-21T23:44:52.000Z",
                    "InstanceId": "i-fedc9876",
                    "VolumeId": "vol-1234abcd",
                    "State": "busy",
                    "DeleteOnTermination": false,
                    "Device": "/dev/sdf"
                }
            ....
        }
    ]
}
```

When you encounter this state, detachment can be delayed indefinitely until you unmount the volume, force detachment, reboot the instance, or all three.

Deleting an Amazon EBS volume

After you no longer need an Amazon EBS volume, you can delete it. After deletion, its data is gone and the volume can't be attached to any instance. However, before deletion, you can store a snapshot of the volume, which you can use to re-create the volume later.

Note

You can't delete a volume if it's attached to an instance. To delete a volume, you must first detach it. For more information, see [Detaching an Amazon EBS volume from a Linux instance \(p. 1058\)](#).

You can check if a volume is attached to an instance. In the console, on the **Volumes** page, you can view the state of your volumes.

- If a volume is attached to an instance, it's in the **in-use** state.
- If a volume is detached from an instance, it's in the **available** state. You can delete this volume.

To delete an EBS volume using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select a volume and choose **Actions, Delete Volume**. If **Delete Volume** is greyed out, the volume is attached to an instance.
4. In the confirmation dialog box, choose **Yes, Delete**.

To delete an EBS volume using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [delete-volume](#) (AWS CLI)
- [Remove-EC2Volume](#) (AWS Tools for Windows PowerShell)

Amazon EBS snapshots

You can back up the data on your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are *incremental* backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves on storage costs by not duplicating data. Each snapshot contains all of the information that is needed to restore your data (from the moment when the snapshot was taken) to a new EBS volume.

When you create an EBS volume based on a snapshot, the new volume begins as an exact replica of the original volume that was used to create the snapshot. The replicated volume loads data in the background so that you can begin using it immediately. If you access data that hasn't been loaded yet, the volume immediately downloads the requested data from Amazon S3, and then continues loading the rest of the volume's data in the background. For more information, see [Creating Amazon EBS snapshots \(p. 1063\)](#).

When you delete a snapshot, only the data unique to that snapshot is removed. For more information, see [Deleting an Amazon EBS snapshot \(p. 1066\)](#).

Snapshot events

You can track the status of your EBS snapshots through CloudWatch Events. For more information, see [EBS snapshot events \(p. 1181\)](#).

Multi-volume snapshots

Snapshots can be used to create a backup of critical workloads, such as a large database or a file system that spans across multiple EBS volumes. Multi-volume snapshots allow you to take exact point-in-time, data coordinated, and crash-consistent snapshots across multiple EBS volumes attached to an EC2 instance. You are no longer required to stop your instance or to coordinate between volumes to ensure crash consistency, because snapshots are automatically taken across multiple EBS volumes. For more information, see the steps for creating a multi-volume EBS snapshot under [Creating Amazon EBS snapshots \(p. 1063\)](#).

Snapshot pricing

Charges for your snapshots are based on the amount of data stored. Because snapshots are incremental, deleting a snapshot might not reduce your data storage costs. Data referenced exclusively by a snapshot is removed when that snapshot is deleted, but data referenced by other snapshots is preserved. For more information, see [Amazon Elastic Block Store Volumes and Snapshots](#) in the *AWS Billing and Cost Management User Guide*.

Contents

- [How incremental snapshots work \(p. 1061\)](#)
- [Copying and sharing snapshots \(p. 1062\)](#)
- [Encryption support for snapshots \(p. 1063\)](#)
- [Creating Amazon EBS snapshots \(p. 1063\)](#)
- [Deleting an Amazon EBS snapshot \(p. 1066\)](#)
- [Copying an Amazon EBS snapshot \(p. 1068\)](#)
- [Viewing Amazon EBS snapshot information \(p. 1072\)](#)
- [Sharing an Amazon EBS snapshot \(p. 1073\)](#)
- [Using EBS direct APIs to access the contents of an EBS snapshot \(p. 1076\)](#)
- [Automating the Amazon EBS snapshot lifecycle \(p. 1097\)](#)

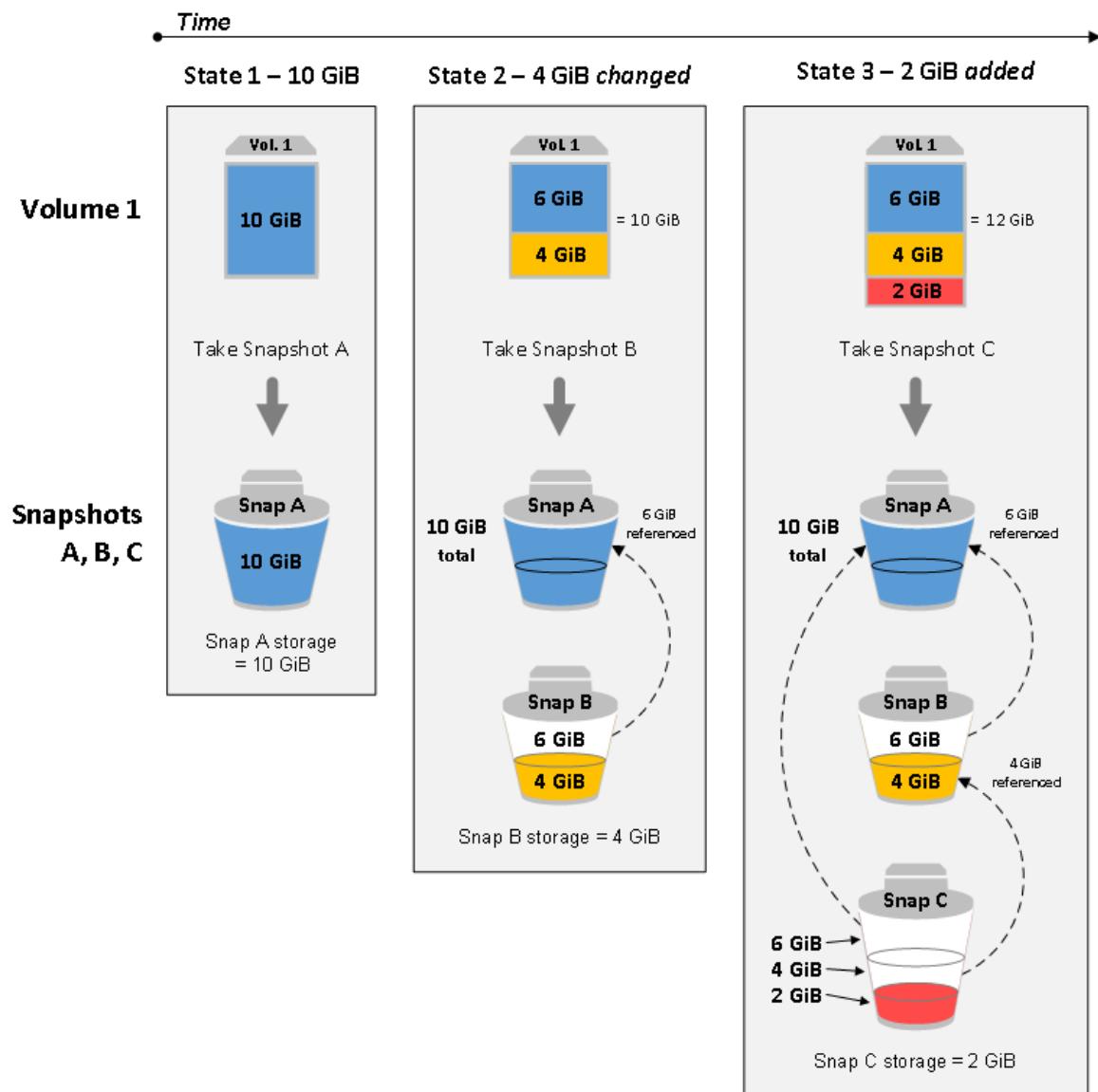
How incremental snapshots work

This section provides illustrations of how an EBS snapshot captures the state of a volume at a point in time, and also how successive snapshots of a changing volume create a history of those changes.

In the diagram below, Volume 1 is shown at three points in time. A snapshot is taken of each of these three volume states.

- In State 1, the volume has 10 GiB of data. Because Snap A is the first snapshot taken of the volume, the entire 10 GiB of data must be copied.
- In State 2, the volume still contains 10 GiB of data, but 4 GiB have changed. Snap B needs to copy and store only the 4 GiB that changed after Snap A was taken. The other 6 GiB of unchanged data, which are already copied and stored in Snap A, are *referenced* by Snap B rather than (again) copied. This is indicated by the dashed arrow.
- In State 3, 2 GiB of data have been added to the volume, for a total of 12 GiB. Snap C needs to copy the 2 GiB that were added after Snap B was taken. As shown by the dashed arrows, Snap C also references 4 GiB of data stored in Snap B, and 6 GiB of data stored in Snap A.
- The total storage required for the three snapshots is 16 GiB.

Relations among multiple snapshots of a volume



Note

If you copy a snapshot and encrypt it to a new CMK, a complete (non-incremental) copy is always created, resulting in additional delay and storage costs.

For more information about how data is managed when you delete a snapshot, see [Deleting an Amazon EBS snapshot \(p. 1066\)](#).

Copying and sharing snapshots

You can share a snapshot across AWS accounts by modifying its access permissions. You can make copies of your own snapshots as well as snapshots that have been shared with you. For more information, see [Sharing an Amazon EBS snapshot \(p. 1073\)](#).

A snapshot is constrained to the AWS Region where it was created. After you create a snapshot of an EBS volume, you can use it to create new volumes in the same Region. For more information, see [Creating a volume from a snapshot \(p. 1041\)](#). You can also copy snapshots across Regions, making it possible to use multiple Regions for geographical expansion, data center migration, and disaster recovery. You can copy

any accessible snapshot that has a completed status. For more information, see [Copying an Amazon EBS snapshot \(p. 1068\)](#).

Encryption support for snapshots

EBS snapshots fully support EBS encryption.

- Snapshots of encrypted volumes are automatically encrypted.
- Volumes that you create from encrypted snapshots are automatically encrypted.
- Volumes that you create from an unencrypted snapshot that you own or have access to can be encrypted on-the-fly.
- When you copy an unencrypted snapshot that you own, you can encrypt it during the copy process.
- When you copy an encrypted snapshot that you own or have access to, you can reencrypt it with a different key during the copy process.
- The first snapshot you take of an encrypted volume that has been created from an unencrypted snapshot is always a full snapshot.
- The first snapshot you take of a reencrypted volume, which has a different CMK compared to the source snapshot, is always a full snapshot.

Note

If you copy a snapshot and encrypt it to a new CMK, a complete (non-incremental) copy is always created, resulting in additional delay and storage costs.

Complete documentation of possible snapshot encryption scenarios is provided in [Creating Amazon EBS snapshots \(p. 1063\)](#) and in [Copying an Amazon EBS snapshot \(p. 1068\)](#).

For more information, see [Amazon EBS encryption \(p. 1121\)](#).

Creating Amazon EBS snapshots

You can create a point-in-time snapshot of an EBS volume and use it as a baseline for new volumes or for data backup. If you make periodic snapshots of a volume, the snapshots are incremental—the new snapshot saves only the blocks that have changed since your last snapshot.

Snapshots occur asynchronously; the point-in-time snapshot is created immediately, but the status of the snapshot is pending until the snapshot is complete (when all of the modified blocks have been transferred to Amazon S3), which can take several hours for large initial snapshots or subsequent snapshots where many blocks have changed. While it is completing, an in-progress snapshot is not affected by ongoing reads and writes to the volume.

You can take a snapshot of an attached volume that is in use. However, snapshots only capture data that has been written to your Amazon EBS volume at the time the snapshot command is issued. This might exclude any data that has been cached by any applications or the operating system. If you can pause any file writes to the volume long enough to take a snapshot, your snapshot should be complete. However, if you can't pause all file writes to the volume, you should unmount the volume from within the instance, issue the snapshot command, and then remount the volume to ensure a consistent and complete snapshot. You can remount and use your volume while the snapshot status is pending.

To make snapshot management easier, you can tag your snapshots during creation or add tags afterward. For example, you can apply tags describing the original volume from which the snapshot was created, or the device name that was used to attach the original volume to an instance. For more information, see [Tagging your Amazon EC2 resources \(p. 1229\)](#).

Snapshot encryption

Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. The data in your encrypted volumes and

any associated snapshots is protected both at rest and in motion. For more information, see [Amazon EBS encryption \(p. 1121\)](#).

By default, only you can create volumes from snapshots that you own. However, you can share your unencrypted snapshots with specific AWS accounts, or you can share them with the entire AWS community by making them public. For more information, see [Sharing an Amazon EBS snapshot \(p. 1073\)](#).

You can share an encrypted snapshot only with specific AWS accounts. For others to use your shared, encrypted snapshot, you must also share the CMK key that was used to encrypt it. Users with access to your encrypted snapshot must create their own personal copy of it and then use that copy. Your copy of a shared, encrypted snapshot can also be re-encrypted using a different key. For more information, see [Sharing an Amazon EBS snapshot \(p. 1073\)](#).

Note

If you copy a snapshot and encrypt it to a new CMK, a complete (non-incremental) copy is always created, resulting in additional delay and storage costs.

Multi-volume snapshots

You can create multi-volume snapshots, which are point-in-time snapshots for all EBS volumes attached to an EC2 instance. You can also create lifecycle policies to automate the creation and retention of multi-volume snapshots. For more information, see [Automating the Amazon EBS snapshot lifecycle \(p. 1097\)](#).

After the snapshots are created, each snapshot is treated as an individual snapshot. You can perform all snapshot operations, such as restore, delete, and copy across Regions or accounts, just as you would with a single volume snapshot. You can also tag your multi-volume snapshots as you would a single volume snapshot. We recommend you tag your multiple volume snapshots to manage them collectively during restore, copy, or retention.

Multi-volume, crash-consistent snapshots are typically restored as a set. It is helpful to identify the snapshots that are in a crash-consistent set by tagging your set with the instance ID, name, or other relevant details. You can also choose to automatically copy tags from the source volume to the corresponding snapshots. This helps you to set the snapshot metadata, such as access policies, attachment information, and cost allocation, to match the source volume.

After creating your snapshots, they appear in your EC2 console created at the exact point-in-time. The snapshots are collectively managed and, therefore, if any one snapshot for the volume set fails, all of the other snapshots display an error status.

Amazon Data Lifecycle Manager

You can create, retain, and delete snapshots manually, or you can use Amazon Data Lifecycle Manager to manage your snapshots for you. For more information, see [Automating snapshots \(p. 1097\)](#).

Considerations

The following considerations apply to creating snapshots:

- When you create a snapshot for an EBS volume that serves as a root device, you should stop the instance before taking the snapshot.
- You cannot create snapshots from instances for which hibernation is enabled.
- You cannot create snapshots from hibernated instances.
- Although you can take a snapshot of a volume while a previous snapshot of that volume is in the pending status, having multiple pending snapshots of a volume can result in reduced volume performance until the snapshots complete.
- There is a limit of five pending snapshots for a single gp2, io1, io2, or Magnetic volume, and one pending snapshot for a single st1 or sc1 volume. If you receive a

ConcurrentSnapshotLimitExceeded error while trying to create multiple concurrent snapshots of the same volume, wait for one or more of the pending snapshots to complete before creating another snapshot of that volume.

- When a snapshot is created from a volume with an AWS Marketplace product code, the product code is propagated to the snapshot.

Creating a snapshot

Use the following procedure to create a snapshot from the specified volume.

To create a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** under **Elastic Block Store** in the navigation pane.
3. Choose **Create Snapshot**.
4. For **Select resource type**, choose **Volume**.
5. For **Volume**, select the volume.
6. (Optional) Enter a description for the snapshot.
7. (Optional) Choose **Add Tag** to add tags to your snapshot. For each tag, provide a tag key and a tag value.
8. Choose **Create Snapshot**.

To create a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [create-snapshot \(AWS CLI\)](#)
- [New-EC2Snapshot \(AWS Tools for Windows PowerShell\)](#)

Creating a multi-volume snapshot

Use the following procedure to create a snapshot from the volumes of an instance.

To create multi-volume snapshots using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** under **Elastic Block Store** in the navigation pane.
3. Choose **Create Snapshot**.
4. For **Select resource type**, choose **Instance**.
5. Select the instance ID for which you want to create simultaneous backups for all of the attached EBS volumes. Multi-volume snapshots support up to 40 EBS volumes per instance.
6. (Optional) Set **Exclude root volume**.
7. (Optional) Set **Copy tags from volume** flag to automatically copy tags from the source volume to the corresponding snapshots. This sets snapshot metadata—such as access policies, attachment information, and cost allocation—to match the source volume.
8. (Optional) Choose **Add Tag** to add tags to your snapshot. For each tag, provide a tag key and a tag value.
9. Choose **Create Snapshot**.

During snapshot creation, the snapshots are managed together. If one of the snapshots in the volume set fails, the other snapshots are moved to error status for the volume set. You can monitor the progress of your snapshots using [CloudWatch Events](#). After the snapshot creation process completes, CloudWatch generates an event that contains the status and all of the relevant snapshots details for the affected instance.

To create multi-volume snapshots using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [create-snapshots](#) (AWS CLI)
- [New-EC2SnapshotBatch](#) (AWS Tools for Windows PowerShell)

Working with EBS snapshots

You can copy snapshots, share snapshots, and create volumes from snapshots. For more information, see the following:

- [Copying an Amazon EBS snapshot \(p. 1068\)](#)
- [Sharing an Amazon EBS snapshot \(p. 1073\)](#)
- [Creating a volume from a snapshot \(p. 1041\)](#)

Deleting an Amazon EBS snapshot

After you no longer need an Amazon EBS snapshot of a volume, you can delete it. Deleting a snapshot has no effect on the volume. Deleting a volume has no effect on the snapshots made from it.

Incremental snapshot deletion

If you make periodic snapshots of a volume, the snapshots are *incremental*. This means that only the blocks on the device that have changed after your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to create volumes. Data that was present on a volume, held in an earlier snapshot or series of snapshots, that is subsequently deleted from that volume at a later time, is still considered unique data of the earlier snapshots. This unique data is not deleted from the sequence of snapshots unless all snapshots that reference the unique data are deleted.

When you delete a snapshot, only the data referenced exclusively by that snapshot is removed. Unique data will not be deleted unless all of the snapshots that reference that data are deleted. Deleting previous snapshots of a volume does not affect your ability to create volumes from later snapshots of that volume.

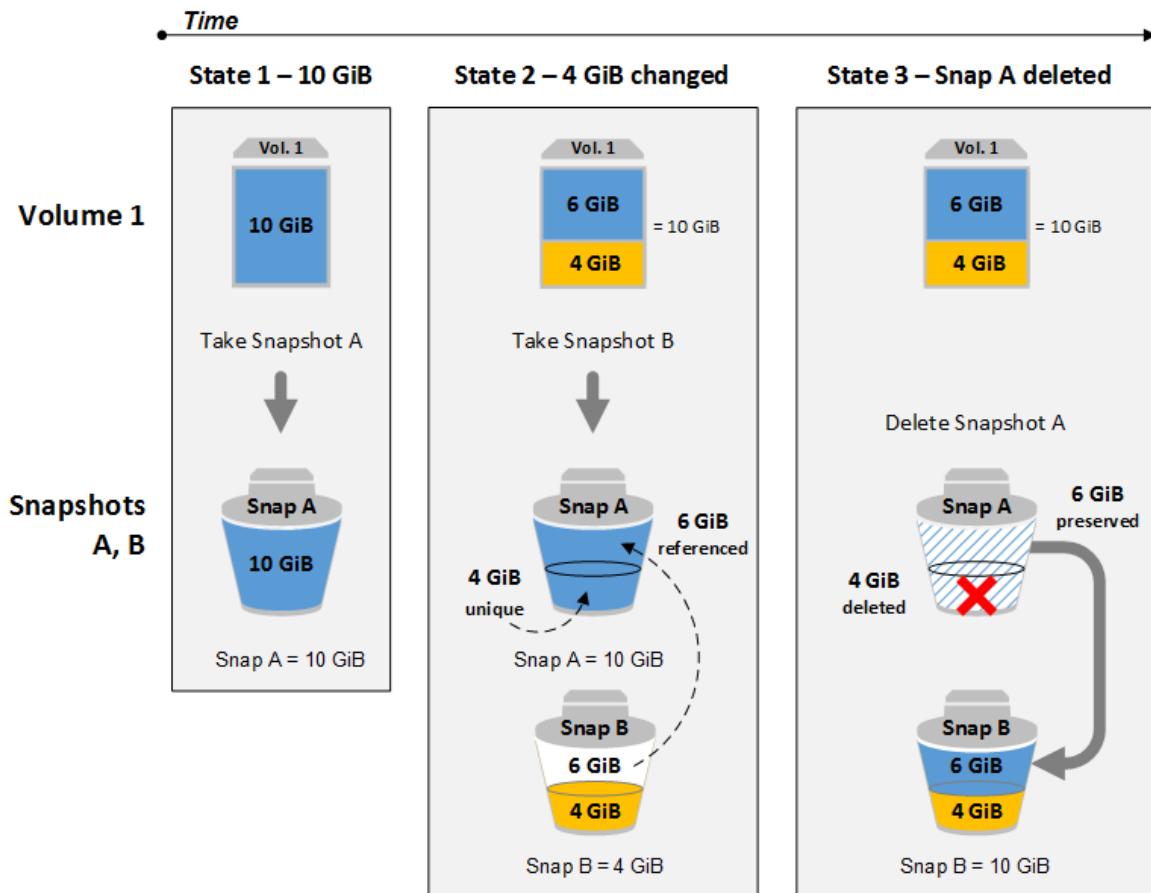
Deleting a snapshot might not reduce your organization's data storage costs. Other snapshots might reference that snapshot's data, and referenced data is always preserved. If you delete a snapshot containing data being used by a later snapshot, costs associated with the referenced data are allocated to the later snapshot. For more information about how snapshots store data, see [How incremental snapshots work \(p. 1061\)](#) and the following example.

In the following diagram, Volume 1 is shown at three points in time. A snapshot has captured each of the first two states, and in the third, a snapshot has been deleted.

- In State 1, the volume has 10 GiB of data. Because Snap A is the first snapshot taken of the volume, the entire 10 GiB of data must be copied.

- In State 2, the volume still contains 10 GiB of data, but 4 GiB have changed. Snap B needs to copy and store only the 4 GiB that changed after Snap A was taken. The other 6 GiB of unchanged data, which are already copied and stored in Snap A, are referenced by Snap B rather than (again) copied. This is indicated by the dashed arrow.
- In state 3, the volume has not changed since State 2, but Snapshot A has been deleted. The 6 GiB of data stored in Snapshot A that were referenced by Snapshot B have now been moved to Snapshot B, as shown by the heavy arrow. As a result, you are still charged for storing 10 GiB of data; 6 GiB of unchanged data preserved from Snap A and 4 GiB of changed data from Snap B.

Deleting a snapshot with some of its data referenced by another snapshot



Considerations

The following considerations apply to deleting snapshots:

- You can't delete a snapshot of the root device of an EBS volume used by a registered AMI. You must first deregister the AMI before you can delete the snapshot. For more information, see [Deregistering your Linux AMI \(p. 172\)](#).
- You can't delete a snapshot that is managed by the AWS Backup service using Amazon EC2. Instead, use AWS Backup to delete the corresponding recovery points in the backup vault.
- You can create, retain, and delete snapshots manually, or you can use Amazon Data Lifecycle Manager to manage your snapshots for you. For more information, see [Automating snapshots \(p. 1097\)](#).
- Although you can delete a snapshot that is still in progress, the snapshot must complete before the deletion takes effect. This might take a long time. If you are also at your concurrent snapshot limit, and you attempt to take an additional snapshot, you might get a `ConcurrentSnapshotLimitExceeded`

error. For more information, see the [Service Quotas](#) for Amazon EBS in the *Amazon Web Services General Reference*.

Delete a snapshot

Use the following procedure to delete a snapshot.

To delete a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** in the navigation pane.
3. Select a snapshot and then choose **Delete** from the **Actions** list.
4. Choose **Yes, Delete**.

To delete a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [delete-snapshot](#) (AWS CLI)
- [Remove-EC2Snapshot](#) (AWS Tools for Windows PowerShell)

Delete a multi-volume snapshot

To delete multi-volume snapshots, retrieve all of the snapshots for your multi-volume group using the tag you applied to the group when you created the snapshots. Then, delete the snapshots individually. You will not be prevented from deleting individual snapshots in the multi-volume snapshots group.

Copying an Amazon EBS snapshot

With Amazon EBS, you can create point-in-time snapshots of volumes, which we store for you in Amazon S3. After you create a snapshot and it has finished copying to Amazon S3 (when the snapshot status is completed), you can copy it from one AWS Region to another, or within the same Region. Amazon S3 server-side encryption (256-bit AES) protects a snapshot's data in transit during a copy operation. The snapshot copy receives an ID that is different from the ID of the original snapshot.

To copy multi-volume snapshots to another AWS Region, retrieve the snapshots using the tag you applied to the multi-volume snapshots group when you created it. Then individually copy the snapshots to another Region.

For information about copying an Amazon RDS snapshot, see [Copying a DB Snapshot](#) in the *Amazon RDS User Guide*.

If you would like another account to be able to copy your snapshot, you must either modify the snapshot permissions to allow access to that account or make the snapshot public so that all AWS accounts can copy it. For more information, see [Sharing an Amazon EBS snapshot \(p. 1073\)](#).

For pricing information about copying snapshots across AWS Regions and accounts, see [Amazon EBS Pricing](#). Note that snapshot copy operations within a single account and Region do not copy any actual data and therefore are cost-free as long as the encryption status of the snapshot copy does not change.

Note

If you copy a snapshot to a new Region, a complete (non-incremental) copy is always created, resulting in additional delay and storage costs.

Note

If you copy a snapshot and encrypt it to a new CMK, a complete (non-incremental) copy is always created, resulting in additional delay and storage costs.

Use cases

- Geographic expansion: Launch your applications in a new AWS Region.
- Migration: Move an application to a new Region, to enable better availability and to minimize cost.
- Disaster recovery: Back up your data and logs across different geographical locations at regular intervals. In case of disaster, you can restore your applications using point-in-time backups stored in the secondary Region. This minimizes data loss and recovery time.
- Encryption: Encrypt a previously unencrypted snapshot, change the key with which the snapshot is encrypted, or create a copy that you own in order to create a volume from it (for encrypted snapshots that have been shared with you).
- Data retention and auditing requirements: Copy your encrypted EBS snapshots from one AWS account to another to preserve data logs or other files for auditing or data retention. Using a different account helps prevent accidental snapshot deletions, and protects you if your main AWS account is compromised.

Prerequisites

- You can copy any accessible snapshots that have a completed status, including shared snapshots and snapshots that you have created.
- You can copy AWS Marketplace, VM Import/Export, and AWS Storage Gateway snapshots, but you must verify that the snapshot is supported in the destination Region.

Limits

- Each account can have up to twenty concurrent snapshot copy requests to a single destination Region.
- User-defined tags are not copied from the source snapshot to the new snapshot. You can add user-defined tags during or after the copy operation. For more information, see [Tagging your Amazon EC2 resources \(p. 1229\)](#).
- Snapshots created by the `CopySnapshot` action have an arbitrary volume ID that should not be used for any purpose.

Incremental snapshot copying

Whether a snapshot copy is incremental is determined by the most recently completed snapshot copy. When you copy a snapshot across Regions or accounts, the copy is an incremental copy if the following conditions are met:

- The snapshot was copied to the destination Region or account previously.
- The most recent snapshot copy still exists in the destination Region or account.
- All copies of the snapshot in the destination Region or account are either unencrypted or were encrypted using the same CMK.

If the most recent snapshot copy was deleted, the next copy is a full copy, not an incremental copy. If a copy is still pending when you start another copy, the second copy starts only after the first copy finishes.

We recommend that you tag your snapshots with the volume ID and creation time so that you can keep track of the most recent snapshot copy of a volume in the destination Region or account.

To see whether your snapshot copies are incremental, check the [copySnapshot \(p. 1183\)](#) CloudWatch event.

Encryption and snapshot copying

When you copy a snapshot, you can encrypt the copy or you can specify a CMK different from the original one, and the resulting copied snapshot uses the new CMK. However, changing the encryption status of a snapshot during a copy operation results in a full (not incremental) copy, which might incur greater data transfer and storage charges.

To copy an encrypted snapshot shared from another AWS account, you must have permissions to use the snapshot and the customer master key (CMK) that was used to encrypt the snapshot. When using an encrypted snapshot that was shared with you, we recommend that you re-encrypt the snapshot by copying it using a CMK that you own. This protects you if the original CMK is compromised, or if the owner revokes it, which could cause you to lose access to any encrypted volumes that you created using the snapshot. For more information, see [Sharing an Amazon EBS snapshot \(p. 1073\)](#).

You apply encryption to EBS snapshot copies by setting the `Encrypted` parameter to `true`. (The `Encrypted` parameter is optional if [encryption by default \(p. 1124\)](#) is enabled).

Optionally, you can use `KmsKeyId` to specify a custom key to use to encrypt the snapshot copy. (The `Encrypted` parameter must also be set to `true`, even if encryption by default is enabled.) If `KmsKeyId` is not specified, the key that is used for encryption depends on the encryption state of the source snapshot and its ownership.

The following table describes the encryption outcome for each possible combination of settings.

Encryption outcomes: Copying a snapshot

Is <code>Encrypted</code> parameter set?	Is encryption by default set?	Source snapshot	Default (no <code>KmsKeyId</code> specified)	Custom (<code>KmsKeyId</code> specified)
No	No	Unencrypted snapshot that you own	Unencrypted	N/A
No	No	Encrypted snapshot that you own	Encrypted by same key	
No	No	Unencrypted snapshot that is shared with you	Unencrypted	
No	No	Encrypted snapshot that is shared with you	Encrypted by default CMK*	
Yes	No	Unencrypted snapshot that you own	Encrypted by default CMK	Encrypted by a specified CMK**
Yes	No	Encrypted snapshot that you own	Encrypted by same key	
Yes	No	Unencrypted snapshot that is shared with you	Encrypted by default CMK	

Is Encrypted parameter set?	Is encryption by default set?	Source snapshot	Default (no KmsKeyId specified)	Custom (KmsKeyId specified)
Yes	No	Encrypted snapshot that is shared with you	Encrypted by default CMK	
No	Yes	Unencrypted snapshot that you own	Encrypted by default CMK	N/A
No	Yes	Encrypted snapshot that you own	Encrypted by same key	
No	Yes	Unencrypted snapshot that is shared with you	Encrypted by default CMK	
No	Yes	Encrypted snapshot that is shared with you	Encrypted by default CMK	
Yes	Yes	Unencrypted snapshot that you own	Encrypted by default CMK	Encrypted by a specified CMK
Yes	Yes	Encrypted snapshot that you own	Encrypted by same key	
Yes	Yes	Unencrypted snapshot that is shared with you	Encrypted by default CMK	
Yes	Yes	Encrypted snapshot that is shared with you	Encrypted by default CMK	

* This is the default CMK used for EBS encryption for the AWS account and Region. By default this is a unique AWS managed CMK for EBS, or you can specify a customer managed CMK. For more information, see [Default key for EBS encryption \(p. 1123\)](#).

** This is a customer managed CMK specified for the copy action. This CMK is used instead of the default CMK for the AWS account and Region.

Copy a snapshot

Use the following procedure to copy a snapshot using the Amazon EC2 console.

To copy a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Snapshots**.
3. Select the snapshot to copy, and then choose **Copy** from the **Actions** list.
4. In the **Copy Snapshot** dialog box, update the following as necessary:
 - **Destination region:** Select the Region where you want to write the copy of the snapshot.

- **Description:** By default, the description includes information about the source snapshot so that you can identify a copy from the original. You can change this description as necessary.
- **Encryption:** If the source snapshot is not encrypted, you can choose to encrypt the copy. If you have enabled [encryption by default \(p. 1124\)](#), the **Encryption** option is set and cannot be unset from the snapshot console. If the **Encryption** option is set, you can choose to encrypt it to a customer managed CMK by selecting one in the field, described below.

You cannot strip encryption from an encrypted snapshot.

Note

If you copy a snapshot and encrypt it to a new CMK, a complete (non-incremental) copy is always created, resulting in additional delay and storage costs.

- **Master Key:** The customer master key (CMK) to be used to encrypt this snapshot. The default key for your account is displayed initially, but you can optionally select from the master keys in your account or type/paste the ARN of a key from a different account. You can create new master encryption keys in the IAM console <https://console.aws.amazon.com/iam/>.

5. Choose **Copy**.
6. In the **Copy Snapshot** confirmation dialog box, choose **Snapshots** to go to the **Snapshots** page in the Region specified, or choose **Close**.

To view the progress of the copy process, switch to the destination Region, and then refresh the **Snapshots** page. Copies in progress are listed at the top of the page.

To check for failure

If you attempt to copy an encrypted snapshot without having permissions to use the encryption key, the operation fails silently. The error state is not displayed in the console until you refresh the page. You can also check the state of the snapshot from the command line, as in the following example.

```
aws ec2 describe-snapshots --snapshot-id snap-0123abcd
```

If the copy failed because of insufficient key permissions, you see the following message: "StateMessage": "Given key ID is not accessible".

When copying an encrypted snapshot, you must have `DescribeKey` permissions on the default CMK. Explicitly denying these permissions results in copy failure. For information about managing CMK keys, see [Controlling Access to Customer Master Keys](#).

To copy a snapshot using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [copy-snapshot \(AWS CLI\)](#)
- [Copy-EC2Snapshot \(AWS Tools for Windows PowerShell\)](#)

Viewing Amazon EBS snapshot information

You can view detailed information about your snapshots.

To view snapshot information using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** in the navigation pane.

3. To reduce the list, choose an option from the **Filter** list. For example, to view only your snapshots, choose **Owned By Me**. You can also filter your snapshots using tags and snapshot attributes. Choose the search bar to view the available tags and attributes.
4. To view more information about a snapshot, select it.

To view snapshot information using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-snapshots \(AWS CLI\)](#)
- [Get-EC2Snapshot \(AWS Tools for Windows PowerShell\)](#)

Example Example: Filter based on tags

The following command describes the snapshots with the tag Stack=production.

```
aws ec2 describe-snapshots --filters Name=tag:Stack,Values=production
```

Example Example: Filter based on volume

The following command describes the snapshots created from the specified volume.

```
aws ec2 describe-snapshots --filters Name=volume-id,Values=vol-049df61146c4d7901
```

Example Example: Filter based on snapshot age

With the AWS CLI, you can use JMESPath to filter results using expressions. For example, the following command displays the IDs of all snapshots created by your AWS account (represented by **123456789012**) before the specified date (represented by **2020-03-31**). If you do not specify the owner, the results include all public snapshots.

```
aws ec2 describe-snapshots --filters Name=owner-id,Values=123456789012 --query "Snapshots[?(StartTime<=`2020-03-31`)].[SnapshotId]" --output text
```

The following command displays the IDs of all snapshots created in the specified date range.

```
aws ec2 describe-snapshots --filters Name=owner-id,Values=123456789012 --query "Snapshots[?(StartTime>=`2019-01-01`)&&(StartTime<=`2019-12-31`)].[SnapshotId]" --output text
```

Sharing an Amazon EBS snapshot

By modifying the permissions of a snapshot, you can share it with the AWS accounts that you specify. Users that you have authorized can use the snapshots you share as the basis for creating their own EBS volumes, while your original snapshot remains unaffected.

If you choose, you can make your unencrypted snapshots available publicly to all AWS users. You can't make your encrypted snapshots available publicly.

When you share an encrypted snapshot, you must also share the customer managed CMK used to encrypt the snapshot. You can apply cross-account permissions to a customer managed CMK either when it is created or at a later time.

Important

When you share a snapshot, you are giving others access to all of the data on the snapshot. Share snapshots only with people with whom you want to share *all* of your snapshot data.

Considerations

The following considerations apply to sharing snapshots:

- Snapshots are constrained to the Region in which they were created. To share a snapshot with another Region, copy the snapshot to that Region. For more information, see [Copying an Amazon EBS snapshot \(p. 1068\)](#).
- AWS prevents you from sharing snapshots that were encrypted with your default CMK. Snapshots that you intend to share must instead be encrypted with a customer managed CMK. For more information, see [Creating Keys](#) in the [AWS Key Management Service Developer Guide](#).
- Users of your shared CMK who are accessing encrypted snapshots must be granted permissions to perform the following actions on the key: `kms:DescribeKey`, `kms>CreateGrant`, `GenerateDataKey`, and `kms:ReEncrypt`. For more information, see [Controlling Access to Customer Master Keys](#) in the [AWS Key Management Service Developer Guide](#).

Sharing an unencrypted snapshot using the console

To share a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** in the navigation pane.
3. Select the snapshot and then choose **Actions, Modify Permissions**.
4. Make the snapshot public or share it with specific AWS accounts as follows:
 - To make the snapshot public, choose **Public**.
This option is not valid for encrypted snapshots or snapshots with an AWS Marketplace product code.
 - To share the snapshot with one or more AWS accounts, choose **Private**, enter the AWS account ID (without hyphens) in **AWS Account Number**, and choose **Add Permission**. Repeat for any additional AWS accounts.
5. Choose **Save**.

To use an unencrypted snapshot that was privately shared with you

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** in the navigation pane.
3. Choose the **Private Snapshots** filter.
4. Locate the snapshot by ID or description. You can use this snapshot as you would any other; for example, you can create a volume from the snapshot or copy the snapshot to a different Region.

Sharing an encrypted snapshot using the console

To share an encrypted snapshot using the console

1. Open the AWS KMS console at <https://console.aws.amazon.com/kms>.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. Choose **Customer managed keys** in the navigation pane.
4. In the **Alias** column, choose the alias (text link) of the customer managed key that you used to encrypt the snapshot. The key details open in a new page.
5. In the **Key policy** section, you see either the *policy view* or the *default view*. The policy view displays the key policy document. The default view displays sections for **Key administrators**, **Key deletion**, **Key Use**, and **Other AWS accounts**. The default view displays if you created the policy in the console

and have not customized it. If the default view is not available, you'll need to manually edit the policy in the policy view. For more information, see [Viewing a Key Policy \(Console\)](#) in the *AWS Key Management Service Developer Guide*.

Use either the policy view or the default view, depending on which view you can access, to add one or more AWS account IDs to the policy, as follows:

- (Policy view) Choose **Edit**. Add one or more AWS account IDs to the following statements: "Allow use of the key" and "Allow attachment of persistent resources". Choose **Save changes**. In the following example, the AWS account ID 444455556666 is added to the policy.

```
{  
    "Sid": "Allow use of the key",  
    "Effect": "Allow",  
    "Principal": {"AWS": [  
        "arn:aws:iam::111122223333:user/CMKUser",  
        "arn:aws:iam::444455556666:root"  
    ]},  
    "Action": [  
        "kms:Encrypt",  
        "kms:Decrypt",  
        "kms:ReEncrypt*",  
        "kms:GenerateDataKey*",  
        "kms:DescribeKey"  
    ],  
    "Resource": "*"  
},  
{  
    "Sid": "Allow attachment of persistent resources",  
    "Effect": "Allow",  
    "Principal": {"AWS": [  
        "arn:aws:iam::111122223333:user/CMKUser",  
        "arn:aws:iam::444455556666:root"  
    ]},  
    "Action": [  
        "kms>CreateGrant",  
        "kms>ListGrants",  
        "kms:RevokeGrant"  
    ],  
    "Resource": "*",  
    "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}  
}
```

- (Default view) Scroll down to **Other AWS accounts**. Choose **Add other AWS accounts** and enter the AWS account ID as prompted. To add another account, choose **Add another AWS account** and enter the AWS account ID. When you have added all AWS accounts, choose **Save changes**.
6. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 7. Choose **Snapshots** in the navigation pane.
 8. Select the snapshot and then choose **Actions, Modify Permissions**.
 9. For each AWS account, enter the AWS account ID in **AWS Account Number** and choose **Add Permission**. When you have added all AWS accounts, choose **Save**.

To use an encrypted snapshot that was shared with you

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** in the navigation pane.
3. Choose the **Private Snapshots** filter. Optionally add the **Encrypted** filter.
4. Locate the snapshot by ID or description.

5. Select the snapshot and choose **Actions, Copy**.
6. (Optional) Select a destination Region.
7. The copy of the snapshot is encrypted by the key displayed in **Master Key**. By default, the selected key is your account's default CMK. To select a customer managed CMK, click inside the input box to see a list of available keys.
8. Choose **Copy**.

Sharing a snapshot using the command line

The permissions for a snapshot are specified using the `createVolumePermission` attribute of the snapshot. To make a snapshot public, set the group to `all`. To share a snapshot with a specific AWS account, set the user to the ID of the AWS account.

To modify snapshot permissions using the command line

Use one of the following commands:

- [modify-snapshot-attribute](#) (AWS CLI)
- [Edit-EC2SnapshotAttribute](#) (AWS Tools for Windows PowerShell)

To view snapshot permissions using the command line

Use one of the following commands:

- [describe-snapshot-attribute](#) (AWS CLI)
- [Get-EC2SnapshotAttribute](#) (AWS Tools for Windows PowerShell)

For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

Determining the use of shared snapshots

You can use AWS CloudTrail to monitor whether a snapshot that you have shared with others is copied or used to create a volume. The following events are logged in CloudTrail:

- **SharedSnapshotCopyInitiated** — A shared snapshot is being copied.
- **SharedSnapshotVolumeCreated** — A shared snapshot is being used to create a volume.

For more information about using CloudTrail, see [Logging Amazon EC2 and Amazon EBS API calls with AWS CloudTrail \(p. 755\)](#).

Using EBS direct APIs to access the contents of an EBS snapshot

You can use the Amazon Elastic Block Store (Amazon EBS) direct APIs to create EBS snapshots, write data directly to your snapshots, read data on your snapshots, and identify the differences or changes between two snapshots. If you're an independent software vendor (ISV) who offers backup services for Amazon EBS, the EBS direct APIs make it more efficient and cost-effective to track incremental changes on your EBS volumes through snapshots. This can be done without having to create new volumes from snapshots, and then use Amazon Elastic Compute Cloud (Amazon EC2) instances to compare the differences.

You can create incremental snapshots directly from data on-premises into EBS volumes and the cloud to use for quick disaster recovery. With the ability to write and read snapshots, you can write your on-premises data to an EBS snapshot during a disaster. Then after recovery, you can restore it back to AWS or on-premises from the snapshot. You no longer need to build and maintain complex mechanisms to copy data to and from Amazon EBS.

This user guide provides an overview of the elements that make up the EBS direct APIs, and examples of how to use them effectively. For more information about the actions, data types, parameters, and errors of the APIs, see the [EBS direct APIs reference](#). For more information about the supported AWS Regions, endpoints, and service quotas for the EBS direct APIs, see [Amazon EBS Endpoints and Quotas in the AWS General Reference](#).

Contents

- [Understanding the EBS direct APIs \(p. 1077\)](#)
- [Permissions for IAM users \(p. 1080\)](#)
- [Using encryption \(p. 1084\)](#)
- [Using Signature Version 4 signing \(p. 1084\)](#)
- [Using checksums \(p. 1084\)](#)
- [Working with the EBS direct APIs using the API or AWS SDKs \(p. 1085\)](#)
- [Working with the EBS direct APIs using the command line \(p. 1089\)](#)
- [Optimizing performance \(p. 1092\)](#)
- [Frequently asked questions \(p. 1093\)](#)
- [Logging API Calls for the EBS direct APIs with AWS CloudTrail \(p. 1094\)](#)
- [EBS direct APIs and interface VPC endpoints \(p. 1096\)](#)
- [Idempotency for StartSnapshot API \(p. 1097\)](#)

Understanding the EBS direct APIs

The following are the key elements that you should understand before getting started with the EBS direct APIs.

Pricing

The price that you pay to use the EBS direct APIs depends on the requests you make. For more information, see [Amazon EBS pricing](#).

Snapshots

Snapshots are the primary means to back up data from your EBS volumes. With the EBS direct APIs, you can also back up data from your on-premises disks to snapshots. To save storage costs, successive snapshots are incremental, containing only the volume data that changed since the previous snapshot. For more information, see [Amazon EBS snapshots \(p. 1060\)](#).

Note

Public snapshots are not supported by the EBS direct APIs.

Blocks

A block is a fragment of data within a snapshot. Each snapshot can contain thousands of blocks. All blocks in a snapshot are of a fixed size.

Block indexes

A block index is the offset position of a block within a snapshot, and it is used to identify the block. Multiply the BlockIndex value with the BlockSize value ($\text{BlockIndex} * \text{BlockSize}$) to identify the logical offset of the data in the logical volume.

Block tokens

A block token is the identifying hash of a block within a snapshot, and it is used to locate the block data. Block tokens returned by EBS direct APIs are temporary. They change on the expiry timestamp specified for them, or if you run another `ListSnapshotBlocks` or `ListChangedBlocks` request for the same snapshot.

Checksum

A checksum is a small-sized datum derived from a block of data for the purpose of detecting errors that were introduced during its transmission or storage. The EBS direct APIs use checksums to validate data integrity. When you read data from an EBS snapshot, the service provides Base64-encoded SHA256 checksums for each block of data transmitted, which you can use for validation. When you write data to an EBS snapshot, you must provide a Base64 encoded SHA256 checksum for each block of data transmitted. The service validates the data received using the checksum provided. For more information, see [Using checksums \(p. 1084\)](#) later in this guide.

Encryption

Encryption protects your data by converting it into unreadable code that can be deciphered only by people who have access to the key used to encrypt it. You can use the EBS direct APIs to read and write encrypted snapshots, but there are some limitations. For more information, see [Using encryption \(p. 1084\)](#) later in this guide.

API actions

The EBS direct APIs consists of six actions. There are three read actions and three write actions. The read actions are `ListSnapshotBlocks`, `ListChangedBlocks`, and `GetSnapshotBlock`. The write actions are `StartSnapshot`, `PutSnapshotBlock`, and `CompleteSnapshot`. These actions are described in the following sections.

List snapshot blocks

The `ListSnapshotBlocks` action returns the block indexes and block tokens of blocks in the specified snapshot.

List changed blocks

The `ListChangedBlocks` action returns the block indexes and block tokens of blocks that are different between two specified snapshots of the same volume and snapshot lineage.

Get snapshot block

The `GetSnapshotBlock` action returns the data in a block for the specified snapshot ID, block index, and block token.

Start snapshot

The `StartSnapshot` action starts a snapshot, either as an incremental snapshot of an existing one or as a new snapshot. The started snapshot remains in a pending state until it is completed using the `CompleteSnapshot` action.

Put snapshot block

The `PutSnapshotBlock` action adds data to a started snapshot in the form of individual blocks. You must specify a Base64-encoded SHA256 checksum for the block of data transmitted. The service validates the checksum after the transmission is completed. The request fails if the checksum computed by the service doesn't match what you specified.

Complete snapshot

The `CompleteSnapshot` action completes a started snapshot that is in a pending state. The snapshot is then changed to a completed state.

Using the EBS direct APIs to read snapshots

The following steps describe how to use the EBS direct APIs to read snapshots:

1. Use the `ListSnapshotBlocks` action to view all block indexes and block tokens of blocks in a snapshot. Or use the `ListChangedBlocks` action to view only the block indexes and block tokens of blocks that are different between two snapshots of the same volume and snapshot lineage. These actions help you identify the block tokens and block indexes of blocks for which you might want to get data.
2. Use the `GetSnapshotBlock` action, and specify the block index and block token of the block for which you want to get data.

For examples of how to run these actions, see the [Working with the EBS direct APIs using the API or AWS SDKs \(p. 1085\)](#) and [Working with the EBS direct APIs using the command line \(p. 1089\)](#) sections later in this guide.

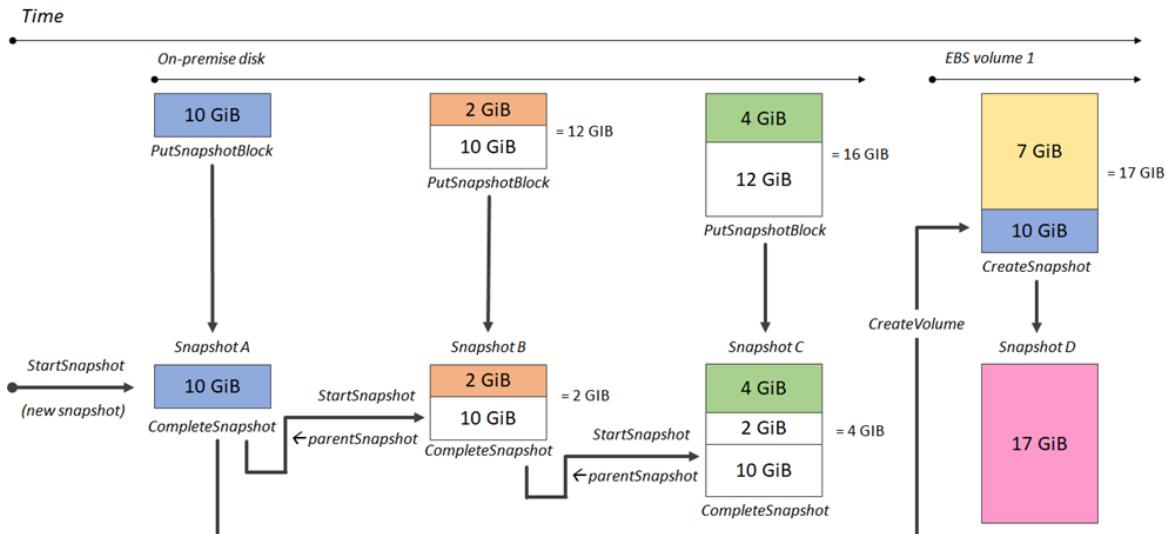
Using the EBS direct APIs to write incremental snapshots

The following steps describe how to use the EBS direct APIs to write incremental snapshots:

1. Use the `StartSnapshot` action and specify a parent snapshot ID to start a snapshot as an incremental snapshot of an existing one, or omit the parent snapshot ID to start a new snapshot. This action returns the new snapshot ID, which is in a pending state.
2. Use the `PutSnapshotBlock` action and specify the ID of the pending snapshot to add data to it in the form of individual blocks. You must specify a Base64-encoded SHA256 checksum for the block of data transmitted. The service computes the checksum of the data received and validates it with the checksum that you specified. The action fails if the checksums don't match.
3. When you're done adding data to the pending snapshot, use the `CompleteSnapshot` action to start an asynchronous workflow that seals the snapshot and moves it to a completed state.

Repeat these steps to create a new, incremental snapshot using the previously created snapshot as the parent.

For example, in the following diagram, snapshot A is the first new snapshot started. Snapshot A is used as the parent snapshot to start snapshot B. Snapshot B is used as the parent snapshot to start and create snapshot C. Snapshots A, B, and C are incremental snapshots. Snapshot A is used to create EBS volume 1. Snapshot D is created from EBS volume 1. Snapshot D is an incremental snapshot of A; it is not an incremental snapshot of B or C.



For examples of how to run these actions, see the [Working with the EBS direct APIs using the API or AWS SDKs \(p. 1085\)](#) and [Working with the EBS direct APIs using the command line \(p. 1089\)](#) sections later in this guide.

Permissions for IAM users

An AWS Identity and Access Management (IAM) user must have the following policies to use the EBS direct APIs. For more information, see [Changing Permissions for an IAM User](#).

Be cautious when assigning the following policies to IAM users. By assigning these policies, you might give access to a user who is denied access to the same resource through the Amazon EC2 APIs, such as the `CopySnapshot` or `CreateVolume` actions.

Permissions to read snapshots

The following policy allows the *read* EBS direct APIs to be used on all snapshots in a specific AWS Region. In the policy, replace `<Region>` with the Region of the snapshot.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ebs:ListSnapshotBlocks",  
                "ebs:ListChangedBlocks",  
                "ebs:GetSnapshotBlock"  
            ],  
            "Resource": "arn:aws:ec2:<Region>::snapshot/*"  
        }  
    ]  
}
```

The following policy allows the *read* EBS direct APIs to be used on snapshots with a specific key-value tag. In the policy, replace `<Key>` with the key value of the tag, and `<Value>` with the value of the tag.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ebs:ListSnapshotBlocks",  
                "ebs:ListChangedBlocks",  
                "ebs:GetSnapshotBlock"  
            ],  
            "Resource": "arn:aws:ec2::snapshot/*",  
            "Condition": {  
                "StringEqualsIgnoreCase": {  
                    "aws:ResourceTag/<Key>": "<Value>"  
                }  
            }  
        }  
    ]  
}
```

The following policy allows all of the *read* EBS direct APIs to be used on all snapshots in the account only within a specific time range. This policy authorizes use of the EBS direct APIs based on the `aws:CurrentTime` global condition key. In the policy, be sure to replace the date and time range shown with the date and time range for your policy.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ebs:ListSnapshotBlocks",  
                "ebs:ListChangedBlocks",  
                "ebs:GetSnapshotBlock"  
            ],  
            "Resource": "arn:aws:ec2::snapshot/*",  
            "Condition": {  
                "LessThan": {  
                    "aws:CurrentTime": "2012-10-17T12:00:00Z"  
                },  
                "LessThanEqual": {  
                    "aws:CurrentTime": "2012-10-17T12:05:00Z"  
                }  
            }  
        }  
    ]  
}
```

```
"Effect": "Allow",
"Action": [
    "ebs>ListSnapshotBlocks",
    "ebs>ListChangedBlocks",
    "ebs>GetSnapshotBlock"
],
"Resource": "arn:aws:ec2::snapshot/*",
"Condition": {
    "DateGreaterThan": {
        "aws:CurrentTime": "2018-05-29T00:00:00Z"
    },
    "DateLessThan": {
        "aws:CurrentTime": "2020-05-29T23:59:59Z"
    }
}
]
```

The following policy grants access to decrypt an encrypted snapshot using a specific key ID from the AWS Key Management Service (AWS KMS). It grants access to encrypt new snapshots using the default AWS KMS key ID for EBS snapshots. It also provides the ability to determine if encrypt by default is enabled on the account. In the policy, replace <Region> with the Region of the AWS KMS key, <AccountId> with the ID of the AWS account of the key, and <KeyId> with the ID of the key used to encrypt the snapshot that you want to read with the EBS direct APIs.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:ReEncrypt*",
                "kms>CreateGrant",
                "ec2>CreateTags",
                "kms:DescribeKey",
                "ec2:GetEbsDefaultKmsKeyId",
                "ec2:GetEbsEncryptionByDefault"
            ],
            "Resource": "arn:aws:kms:<Region>:<AccountId>:key/<KeyId>"
        }
    ]
}
```

For more information, see [Changing Permissions for an IAM User](#) in the *IAM User Guide*.

Permissions to write snapshots

The following policy allows the *write* EBS direct APIs to be used on all snapshots in a specific AWS Region. In the policy, replace <Region> with the Region of the snapshot.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:StartSnapshot",
                "ebs:DeleteSnapshot"
            ],
            "Resource": "*"
        }
    ]
}
```

```
        "ebs:PutSnapshotBlock",
        "ebs:CompleteSnapshot"
    ],
    "Resource": "arn:aws:ec2:<Region>::snapshot/*"
}
]
```

The following policy allows the *write* EBS direct APIs to be used on snapshots with a specific key-value tag. In the policy, replace **<Key>** with the key value of the tag, and **<Value>** with the value of the tag.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:StartSnapshot",
                "ebs:PutSnapshotBlock",
                "ebs:CompleteSnapshot"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*",
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "aws:ResourceTag/<Key>": "<Value>"
                }
            }
        }
    ]
}
```

The following policy allows all of the EBS direct APIs to be used. It also allows the `StartSnapshot` action only if a parent snapshot ID is specified. Therefore, this policy blocks the ability to start new snapshots without using a parent snapshot.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ebs:*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ebs:ParentSnapshot": "arn:aws:ec2::*:snapshot/*"
                }
            }
        }
    ]
}
```

The following policy allows all of the EBS direct APIs to be used. It also allows only the `user` tag key to be created for a new snapshot. This policy also ensures that the user has access to create tags. The `StartSnapshot` action is the only action that can specify tags.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ebs:*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ebs:TagUser": "user"
                }
            }
        }
    ]
}
```

```
"Condition": {
    "ForAllValues:StringEquals": {
        "aws:TagKeys": "user"
    }
},
{
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "*"
}
]
```

The following policy allows all of the *write* EBS direct APIs to be used on all snapshots in the account only within a specific time range. This policy authorizes use of the EBS direct APIs based on the `aws:CurrentTime` global condition key. In the policy, be sure to replace the date and time range shown with the date and time range for your policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:StartSnapshot",
                "ebs:PutSnapshotBlock",
                "ebs:CompleteSnapshot"
            ],
            "Resource": "arn:aws:ec2:::snapshot/*",
            "Condition": {
                "DateGreaterThan": {
                    "aws:CurrentTime": "2018-05-29T00:00:00Z"
                },
                "DateLessThan": {
                    "aws:CurrentTime": "2020-05-29T23:59:59Z"
                }
            }
        }
    ]
}
```

The following policy grants access to decrypt an encrypted snapshot using a specific key ID from the AWS Key Management Service (AWS KMS). It grants access to encrypt new snapshots using the default AWS KMS key ID for EBS snapshots. It also provides the ability to determine if encrypt by default is enabled on the account. In the policy, replace `<Region>` with the Region of the AWS KMS key, `<AccountId>` with the ID of the AWS account of the key, and `<KeyId>` with the ID of the key used to encrypt the snapshot that you want to read with the EBS direct APIs.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:ReEncrypt*",
                "kms>CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": [
                "arn:aws:kms:<Region>::<AccountId>/alias/<KeyId>"
            ]
        }
    ]
}
```

```
        "ec2:CreateTags",
        "kms:DescribeKey",
        "ec2:GetEbsDefaultKmsKeyId",
        "ec2:GetEbsEncryptionByDefault"
    ],
    "Resource": "arn:aws:kms:<Region>:<AccountId>:key/<KeyId>"
}
]
```

For more information, see [Changing Permissions for an IAM User](#) in the *IAM User Guide*.

Using encryption

If Amazon EBS encryption by default is enabled on your AWS account, you cannot start a new snapshot using an un-encrypted parent snapshot. You must first encrypt the parent snapshot by copying it. For more information, see [Copying an Amazon EBS snapshot \(p. 1068\)](#) and [Encryption by default \(p. 1124\)](#).

To start an encrypted snapshot, specify the Amazon Resource Name (ARN) of an AWS KMS key, or specify an encrypted parent snapshot in your StartSnapshot request. If neither are specified, and Amazon EBS encryption by default is enabled on the account, then the default CMK for the account is used. If no default CMK has been specified for the account, then the AWS managed CMK is used.

Important

By default, all principals in the account have access to the default AWS managed CMK, and they can use it for EBS encryption and decryption operations. For more information, see [Default key for EBS encryption \(p. 1123\)](#).

You might need additional IAM permissions to use the EBS direct APIs with encryption. For more information, see the [Permissions for IAM users \(p. 1080\)](#) section earlier in this guide.

Using Signature Version 4 signing

Signature Version 4 is the process to add authentication information to AWS requests sent by HTTP. For security, most requests to AWS must be signed with an access key, which consists of an access key ID and secret access key. These two keys are commonly referred to as your security credentials. For information about how to obtain credentials for your account, see [Understanding and getting your credentials](#).

If you intend to manually create HTTP requests, you must learn how to sign them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

For more information, see [Signing AWS requests with Signature Version 4](#) in the *AWS General Reference*.

Using checksums

The GetSnapshotBlock action returns data that is in a block of a snapshot, and the PutSnapshotBlock action adds data to a block in a snapshot. The block data that is transmitted is not signed as part of the Signature Version 4 signing process. As a result, checksums are used to validate the integrity of the data as follows:

- When you use the GetSnapshotBlock action, the response provides a Base64-encoded SHA256 checksum for the block data using the **x-amz-Checksum** header, and the checksum algorithm using the **x-amz-Checksum-Algorithm** header. Use the returned checksum to validate the integrity of the data. If the checksum that you generate doesn't match what Amazon EBS provided, you should consider the data not valid and retry your request.
- When you use the PutSnapshotBlock action, your request must provide a Base64-encoded SHA256 checksum for the block data using the **x-amz-Checksum** header, and the checksum algorithm using the **x-amz-Checksum-Algorithm** header. The checksum that you provide is validated against a

checksum generated by Amazon EBS to validate the integrity of the data. If the checksums do not correspond, the request fails.

- When you use the CompleteSnapshot action, your request can optionally provide an aggregate Base64-encoded SHA256 checksum for the complete set of data added to the snapshot. Provide the checksum using the **x-amz-Checksum** header, the checksum algorithm using the **x-amz-Checksum-Algorithm** header, and the checksum aggregation method using the **x-amz-Checksum-Aggregation-Method** header. To generate the aggregated checksum using the linear aggregation method, arrange the checksums for each written block in ascending order of their block index, concatenate them to form a single string, and then generate the checksum on the entire string using the SHA256 algorithm.

The checksums in these actions are part of the Signature Version 4 signing process.

Working with the EBS direct APIs using the API or AWS SDKs

The [EBS direct APIs Reference](#) provides descriptions and syntax for each of the service's actions and data types. You can also use one of the AWS SDKs to access an API that's tailored to the programming language or platform that you're using. For more information, see [AWS SDKs](#).

The EBS direct APIs require an AWS Signature Version 4 signature. For more information, see [Using Signature Version 4 signing \(p. 1084\)](#).

Using the API to read snapshots

List blocks in a snapshot

The following [ListChangedBlocks](#) example request returns the block indexes and block tokens of blocks that are in snapshot snap-0acEXAMPLEcf41648. The `startingBlockIndex` parameter limits the results to block indexes greater than 1000, and the `maxResults` parameter limits the results to the first 100 blocks.

```
GET /snapshots/snap-0acEXAMPLEcf41648/blocks?maxResults=100&startingBlockIndex=0 HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
User-Agent: <User agent parameter>
X-Amz-Date: 20200617T231953Z
Authorization: <Authentication parameter>
```

The following example response for the previous request lists the block indexes and block tokens in the snapshot. Use the GetSnapshotBlock action and specify the block index and block token of the block for which you want to get data. The block tokens are valid until the expiry time listed.

```
HTTP/1.1 200 OK
x-amzn-RequestId: d6e5017c-70a8-4539-8830-57f5557f3f27
Content-Type: application/json
Content-Length: 2472
Date: Wed, 17 Jun 2020 23:19:56 GMT
Connection: keep-alive

{
    "BlockSize": 524288,
    "Blocks": [
        {
            "BlockIndex": 0,
            "BlockToken": "AAUBAcuWqOCnDNuKle11s7IIIX6jp6FYcc/q8oT93913HhvLvA+3JRrSybp/0"
        },
        {
            "BlockIndex": 1536,
            "BlockToken": "AAUBAWudwfmofcrQhGV1LwuRKm2b8ZXPiyrgoykTRC6IU1NbxBWDY1pPjvnV"
        }
    ]
}
```

```
        "BlockIndex": 3072,
        "BlockToken": "AAUBAV7p6pC5fKAC7TokoNCtAnZhqq27u6YEXZ3MwRevBkDjmMx6iuA6tsBt"
    },
    {
        "BlockIndex": 3073,
        "BlockToken": "AAUBAbqt9zpqBUEvtO2HINAffaWToOwlPjbIsQ0lx6JUN/0+iMq10NtNbnX4"
    },
    ...
],
"ExpiryTime": 1.59298379649E9,
"VolumeSize": 3
}
```

List blocks that are different between two snapshots

The following [ListChangedBlocks](#) example request returns the block indexes and block tokens of blocks that are different between snapshots `snap-0acEXAMPLEcf41648` and `snap-0c9EXAMPLE1b30e2f`. The `startingBlockIndex` parameter limits the results to block indexes greater than 0, and the `maxResults` parameter limits the results to the first 500 blocks.

```
GET /snapshots/snap-0c9EXAMPLE1b30e2f/changedblocks?
firstSnapshotId=snap-0acEXAMPLEcf41648&maxResults=500&startingBlockIndex=0 HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
User-Agent: <User agent parameter>
X-Amz-Date: 20200617T232546Z
Authorization: <Authentication parameter>
```

The following example response for the previous request shows that block indexes 0, 3072, 6002, and 6003 are different between the two snapshots. Additionally, block indexes 6002, and 6003 exist only in the first snapshot ID specified, and not in the second snapshot ID because there is no second block token listed in the response.

Use the `GetSnapshotBlock` action and specify the block index and block token of the block for which you want to get data. The block tokens are valid until the expiry time listed.

```
HTTP/1.1 200 OK
x-amzn-RequestId: fb0f6743-6d81-4be8-afbe-db11a5bb8a1f
Content-Type: application/json
Content-Length: 1456
Date: Wed, 17 Jun 2020 23:25:47 GMT
Connection: keep-alive

{
    "BlockSize": 524288,
    "ChangedBlocks": [
        {
            "BlockIndex": 0,
            "FirstBlockToken": "AAUBAVaWqOCnDNuKle11s7IIX6jp6FYcC/tJuVT1GgP23AuLntwiMdJ+OJkL",
            "SecondBlockToken": "AAUBASxzy0Y0b33JVRLoYm3NOresCxn5R0+HVfzXW3Y/RwfFaPX2Edx8QHCh"
        },
        {
            "BlockIndex": 3072,
            "FirstBlockToken": "AAUBAcHp6pC5fKAC7TokoNCtAnZhqq27u6fxRfZOLEmeXLmHBf2R/Yb24MaS",
            "SecondBlockToken": "AAUBARGCaufCqBRZC8tEkPYGGkSv3vqv0jJ2xKDi3ljdFiytUxBLXYgTmkid"
        },
        {
            "BlockIndex": 6002,
            "FirstBlockToken": "AAUBABqf9zpqBUEvtO2HINAffaWToOwlPjbIsQ0lx6JUN/0+iMq10NtNbnX4"
        }
    ]
}
```

```
        "FirstBlockToken": "AAABASqX4/  
NWjvNceoyMULjcRd0DnwbSwNnes1UkoP62CrQXvn47BY5435aw"  
    },  
    {  
        "BlockIndex": 6003,  
        "FirstBlockToken":  
"AAABASmJ005JxAOce25rF4P1sdRtyIDsX12tFEDunnePYUKOf4PBROuICb2A"  
    },  
    ...  
],  
"ExpiryTime": 1.592976647009E9,  
"VolumeSize": 3  
}
```

Get block data from a snapshot

The following [GetSnapshotBlock](#) example request returns the data in the block index 3072 with block token AAUBARGCaufCqBRZC8tEkPYGGkSv3vqv0jJ2xKDj3ljdFiytUxBLXYgTmkid, in snapshot snap-0c9EXAMPLE1b30e2f.

```
GET /snapshots/snap-0c9EXAMPLE1b30e2f/blocks/3072?  
blockToken=AAUBARGCaufCqBRZC8tEkPYGGkSv3vqv0jJ2xKDj3ljdFiytUxBLXYgTmkid HTTP/1.1  
Host: ebs.us-east-2.amazonaws.com  
Accept-Encoding: identity  
User-Agent: <User agent parameter>  
X-Amz-Date: 20200617T232838Z  
Authorization: <Authentication parameter>
```

The following example response for the previous request shows the size of the data returned, the checksum to validate the data, and the algorithm used to generate the checksum. The binary data is transmitted in the body of the response and is represented as *BlockData* in the following example.

```
HTTP/1.1 200 OK  
x-amzn-RequestId: 2d0db2fb-bd88-474d-a137-81c4e57d7b9f  
x-amz-Data-Length: 524288  
x-amz-C checksum: Vc0yY2j3qg8bUL9I6GQuI2orTudrQRBDMIhc7bdEsw=  
x-amz-C checksum-Algorithm: SHA256  
Content-Type: application/octet-stream  
Content-Length: 524288  
Date: Wed, 17 Jun 2020 23:28:38 GMT  
Connection: keep-alive  
  
BlockData
```

Using the API to write incremental snapshots

Start a snapshot

The following [StartSnapshot](#) example request starts an 8 GiB snapshot, using snapshot snap-123EXAMPLE1234567 as the parent snapshot. The new snapshot will be an incremental snapshot of the parent snapshot. The snapshot moves to an error state if there are no put or complete requests made for the snapshot within the specified 60 minute timeout period. The 550e8400-e29b-41d4-a716-446655440000 client token ensures idempotency for the request. If the client token is omitted, the AWS SDK automatically generates one for you. For more information about idempotency, see [Idempotency for StartSnapshot API \(p. 1097\)](#).

```
POST /snapshots HTTP/1.1  
Host: ebs.us-east-2.amazonaws.com  
Accept-Encoding: identity  
User-Agent: <User agent parameter>  
X-Amz-Date: 20200618T040724Z
```

```
Authorization: <Authentication parameter>

{
    "VolumeSize": 8,
    "ParentSnapshot": "snap-123EXAMPLE1234567",
    "ClientToken": "550e8400-e29b-41d4-a716-446655440000",
    "Timeout": 60
}
```

The following example response for the previous request shows the snapshot ID, AWS account ID, status, volume size in GiB, and size of the blocks in the snapshot. The snapshot is started in a pending state. Specify the snapshot ID in a subsequent `PutSnapshotBlocks` request to write data to the snapshot.

```
HTTP/1.1 201 Created
x-amzn-RequestId: 929e6eb9-7183-405a-9502-5b7da37c1b18
Content-Type: application/json
Content-Length: 181
Date: Thu, 18 Jun 2020 04:07:29 GMT
Connection: keep-alive

{
    "BlockSize": 524288,
    "Description": null,
    "OwnerId": "138695307491",
    "Progress": null,
    "SnapshotId": "snap-052EXAMPLEc85d8dd",
    "StartTime": null,
    "Status": "pending",
    "Tags": null,
    "VolumeSize": 8
}
```

Put data into a snapshot

The following `PutSnapshot` example request writes 524288 Bytes of data to block index 1000 on snapshot snap-052EXAMPLEc85d8dd. The Base64 encoded QOD3gmEQQXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM= checksum was generated using the SHA256 algorithm. The data is transmitted in the body of the request and is represented as `BlockData` in the following example.

```
PUT /snapshots/snap-052EXAMPLEc85d8dd(blocks/1000 HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
x-amz-Data-Length: 524288
x-amz-C checksum: QOD3gmEQQXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM=
x-amz-C checksum-Algorithm: SHA256
User-Agent: <User agent parameter>
X-Amz-Date: 20200618T042215Z
X-Amz-Content-SHA256: UNSIGNED-PAYLOAD
Authorization: <Authentication parameter>

BlockData
```

The following example response for the previous request confirms the data length, checksum, and checksum algorithm for the data received by the service.

```
HTTP/1.1 201 Created
x-amzn-RequestId: 643ac797-7e0c-4ad0-8417-97b77b43c57b
x-amz-C checksum: QOD3gmEQQXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM=
x-amz-C checksum-Algorithm: SHA256
Content-Type: application/json
```

```
Content-Length: 2
Date: Thu, 18 Jun 2020 04:22:12 GMT
Connection: keep-alive

{}
```

Complete a snapshot

The following [CompleteSnapshot](#) example request completes snapshot snap-052EXAMPLEc85d8dd. The command specifies that 5 blocks were written to the snapshot. The 6D3nmwi5f2F0wlh7xX8QprJBFzDX8aacdOcA3KCM3c= checksum represents the checksum for the complete set of data written to a snapshot.

```
POST /snapshots/completion/snap-052EXAMPLEc85d8dd HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
x-amz-ChangedBlocksCount: 5
x-amz-C checksum: 6D3nmwi5f2F0wlh7xX8QprJBFzDX8aacdOcA3KCM3c=
x-amz-C checksum-Algorithm: SHA256
x-amz-C checksum-Aggregation-Method: LINEAR
User-Agent: <User agent parameter>
X-Amz-Date: 20200618T043158Z
Authorization: <Authentication parameter>
```

The following is an example response for the previous request.

```
HTTP/1.1 202 Accepted
x-amzn-RequestId: 06cba5b5-b731-49de-af40-80333ac3a117
Content-Type: application/json
Content-Length: 20
Date: Thu, 18 Jun 2020 04:31:50 GMT
Connection: keep-alive

{"Status": "pending"}
```

Working with the EBS direct APIs using the command line

The following examples show how to use the EBS direct APIs using the AWS Command Line Interface (AWS CLI). For more information about installing and configuring the AWS CLI, see [Installing the AWS CLI](#) and [Quickly Configuring the AWS CLI](#).

Using the AWS CLI to read snapshots

List blocks in a snapshot

The following [list-snapshot-blocks](#) example command returns the block indexes and block tokens of blocks that are in snapshot snap-0987654321. The --starting-block-index parameter limits the results to block indexes greater than 1000, and the --max-results parameter limits the results to the first 100 blocks.

```
aws ebs list-snapshot-blocks --snapshot-id snap-0987654321 --starting-block-index 1000 --max-results 100
```

The following example response for the previous command lists the block indexes and block tokens in the snapshot. Use the [get-snapshot-block](#) command and specify the block index and block token of the block for which you want to get data. The block tokens are valid until the expiry time listed.

```
{
    "Blocks": [
        {
```

```
        "BlockIndex": 1001,
        "BlockToken": "AAABAV3/PNhXOynVdMYHUpPsetaSvjLB1dtIGfbJv5OJ0sX855EzGTWos4a4"
    },
    {
        "BlockIndex": 1002,
        "BlockToken": "AAABATGQIgwr0WwIuqIMjCA/Sy7e/YoQFZsHejzGNvjKauzNgzeI13YHBfQB"
    },
    {
        "BlockIndex": 1007,
        "BlockToken": "AAABAZ9CTuQtUvp/dXqRWw4d07e0gTZ3jvn6hiW30W9duM8MiMw6yQayzF2c"
    },
    {
        "BlockIndex": 1012,
        "BlockToken": "AAABAQdzxhw0rVV6PNmsfo/YRIxo9JPR85XxPf1BLjg0Hec6pygYr6laE1p0"
    },
    {
        "BlockIndex": 1030,
        "BlockToken": "AAABAAaYvPax6mv+iGWLdTUjQtFWouQ7Dqz6nSD9L+CbXnvpkswA6iIDID523d"
    },
    {
        "BlockIndex": 1031,
        "BlockToken": "AAABATgWZC0XcFwUKvTJbUXMiSPg59KVxJGL+BWBCLkw6spzCxJVqDVaTskJ"
    },
    ...
],
"ExpiryTime": 1576287332.806,
"VolumeSize": 32212254720,
"BlockSize": 524288
}
```

List blocks that are different between two snapshots

The following [list-changed-blocks](#) example command returns the block indexes and block tokens of blocks that are different between snapshots snap-1234567890 and snap-0987654321. The --starting-block-index parameter limits the results to block indexes greater than 0, and the --max-results parameter limits the results to the first 500 blocks..

```
aws ebs list-changed-blocks --first-snapshot-id snap-1234567890 --second-snapshot-id snap-0987654321 --starting-block-index 0 --max-results 500
```

The following example response for the previous command shows that block indexes 0, 6000, 6001, 6002, and 6003 are different between the two snapshots. Additionally, block indexes 6001, 6002, and 6003 exist only in the first snapshot ID specified, and not in the second snapshot ID because there is no second block token listed in the response.

Use the [get-snapshot-block](#) command and specify the block index and block token of the block for which you want to get data. The block tokens are valid until the expiry time listed.

```
{
    "ChangedBlocks": [
        {
            "BlockIndex": 0,
            "FirstBlockToken": "AAABAVahm9S060Dyi0ORySzn2ZjGjw/KN3uygGlS0QOYWesbzBbDnX2dGpmC",
            "SecondBlockToken":
"AAABAf8o0o6UFi1rDbSZGIRaCEdDyBu9TlvtCQxxoKV8qrUPQP7vcM6iWGsr"
        },
        {
            "BlockIndex": 6000,
            "FirstBlockToken": "AAABAbYSiZvJ0/R9tz8suI8dSzecLjN4kkazK8inFXVintPkdaVFLfCMQsKe",
            "SecondBlockToken":
"AAABAZnqTdzFmKRpsaMASDxviVqEI/3jJzI2crq2eFDCgHmyNf777elD9oVR"
```

```
},
{
    "BlockIndex": 6001,
    "FirstBlockToken": "AAABASBpSJ2UAD3PLxJnCt6zun4/
T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR"
},
{
    "BlockIndex": 6002,
    "FirstBlockToken": "AAABASqX4/
NWjvNceoyMULjcRd0DnwbSwNnes1UkoP62CrQXvn47BY5435aw"
},
{
    "BlockIndex": 6003,
    "FirstBlockToken":
"AAABASmJ005JxAOce25rF4P1sdRtyIDsX12tFEDunnePYUKOf4PBROuICb2A"
},
...
],
"ExpiryTime": 1576308931.973,
"VolumeSize": 32212254720,
"BlockSize": 524288,
"NextToken": "AAADARqElNng/sV98CYk/bJDCXeLJmLJHnNSkHvLzVaO0zsPH/QM3Bi3zF//O6Mdi/
BbJarBnp8h"
}
```

Get block data from a snapshot

The following [get-snapshot-block](#) example command returns the data in the block index 6001 with block token AAABASBpSJ2UAD3PLxJnCt6zun4/T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR, in snapshot snap-1234567890. The binary data is output to the data file in the C:\Temp directory on a Windows computer. If you run the command on a Linux or Unix computer, replace the output path with /tmp/data to output the data to the data file in the /tmp directory.

```
aws ebs get-snapshot-block --snapshot-id snap-1234567890 --block-index 6001 --block-
token AAABASBpSJ2UAD3PLxJnCt6zun4/T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR C:/Temp/data
```

The following example response for the previous command shows the size of the data returned, the checksum to validate the data, and the algorithm of the checksum. The binary data is automatically saved to the directory and file you specified in the request command.

```
{
    "DataLength": "524288",
    "Checksum": "cf0Y6/Fn0oFa4VyjQPOa/iD0zhTf1PTKzxGv2OKowXc=",
    "ChecksumAlgorithm": "SHA256"
}
```

Using the AWS CLI to write incremental snapshots

Start a snapshot

The following [start-snapshot](#) example command starts an 8 GiB snapshot, using snapshot snap-123EXAMPLE1234567 as the parent snapshot. The new snapshot will be an incremental snapshot of the parent snapshot. The snapshot moves to an error state if there are no put or complete requests made for the snapshot within the specified 60 minute timeout period. The 550e8400-e29b-41d4-a716-446655440000 client token ensures idempotency for the request. If the client token is omitted, the AWS SDK automatically generates one for you. For more information about idempotency, see [Idempotency for StartSnapshot API \(p. 1097\)](#).

```
aws ebs start-snapshot --volume-size 8 --parent-snapshot snap-123EXAMPLE1234567 --
timeout 60 --client-token 550e8400-e29b-41d4-a716-446655440000
```

The following example response for the previous command shows the snapshot ID, AWS account ID, status, volume size in GiB, and size of the blocks in the snapshot. The snapshot is started in a pending state. Specify the snapshot ID in subsequent `put-snapshot-block` commands to write data to the snapshot, then use the `complete-snapshot` command to complete the snapshot and change its status to completed.

```
{  
    "SnapshotId": "snap-0aaEXAMPLEe306d62",  
    "OwnerId": "111122223333",  
    "Status": "pending",  
    "VolumeSize": 8,  
    "BlockSize": 524288  
}
```

Put data into a snapshot

The following `put-snapshot-block` example command writes 524288 Bytes of data to block index 1000 on snapshot `snap-0aaEXAMPLEe306d62`. The Base64 encoded `QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM=` checksum was generated using the SHA256 algorithm. The data that is transmitted is in the `/tmp/data` file.

```
aws ebs put-snapshot-block --snapshot-id snap-0aaEXAMPLEe306d62  
    --block-index 1 --data-length 524288 --block-data /tmp/data --  
    checksum QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM= --checksum-algorithm SHA256
```

The following example response for the previous command confirms the data length, checksum, and checksum algorithm for the data received by the service.

```
{  
    "DataLength": "524288",  
    "Checksum": "QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM=",  
    "ChecksumAlgorithm": "SHA256"  
}
```

Complete a snapshot

The following `complete-snapshot` example command completes snapshot `snap-0aaEXAMPLEe306d62`. The command specifies that 5 blocks were written to the snapshot. The `6D3nmwi5f2F0wlh7xx8QprrrJBFzDX8aacd0ca3KCM3c=` checksum represents the checksum for the complete set of data written to a snapshot. For more information about checksums, see [Using checksums \(p. 1084\)](#) earlier in this guide.

```
aws ebs complete-snapshot --snapshot-id snap-0aaEXAMPLEe306d62 --changed-blocks-count 5  
    --checksum 6D3nmwi5f2F0wlh7xx8QprrrJBFzDX8aacd0ca3KCM3c= --checksum-algorithm SHA256 --  
    checksum-aggregation-method LINEAR
```

The following is an example response for the previous command.

```
{  
    "Status": "pending"  
}
```

Optimizing performance

You can run API requests concurrently. Assuming `PutSnapshotBlock` latency is 100ms, then a thread can process 10 requests in one second. Furthermore, assuming your client application creates multiple

threads and connections (for example, 100 connections), it can make 1000 ($10 * 100$) requests per second in total. This will correspond to a throughput of around 500 MB per second.

The following list contains few things to look for in your application:

- Is each thread using a separate connection? If the connections are limited on the application then multiple threads will wait for the connection to be available and you will notice lower throughput.
- Is there any wait time in the application between two put requests? This will reduce the effective throughput of a thread.
- The bandwidth limit on the instance – If bandwidth on the instance is shared by other applications, it could limit the available throughput for PutSnapshotBlock requests.

Be sure to take note of other workloads that might be running in the account to avoid bottlenecks. You should also build retry mechanisms into your EBS direct APIs workflows to handle throttling, timeouts, and service unavailability.

Review the EBS direct APIs service quotas to determine the maximum API requests that you can run per second. For more information, see [Amazon Elastic Block Store Endpoints and Quotas](#) in the *AWS General Reference*.

Frequently asked questions

Can a snapshot be accessed using the EBS direct APIs if it has a pending status?

No. The snapshot can be accessed only if it has a completed status.

Are the block indexes returned by the EBS direct APIs in numerical order?

Yes. The block indexes returned are unique, and in numerical order.

Can I submit a request with a MaxResults parameter value of under 100?

No. The minimum MaxResult parameter value you can use is 100. If you submit a request with a MaxResult parameter value of under 100, and there are more than 100 blocks in the snapshot, then the API will return at least 100 results.

Can I run API requests concurrently?

You can run API requests concurrently. Be sure to take note of other workloads that might be running in the account to avoid bottlenecks. You should also build retry mechanisms into your EBS direct APIs workflows to handle throttling, timeouts, and service unavailability. For more information, see [Optimizing performance \(p. 1092\)](#).

Review the EBS direct APIs service quotas to determine the API requests that you can run per second. For more information, see [Amazon Elastic Block Store Endpoints and Quotas](#) in the *AWS General Reference*.

When running the ListChangedBlocks action, is it possible to get an empty response even though there are blocks in the snapshot?

Yes. If the changed blocks are scarce in the snapshot, the response may be empty but the API will return a next page token value. Use the next page token value to continue to the next page of results. You can confirm that you have reached the last page of results when the API returns a next page token value of null.

If the NextToken parameter is specified together with a StartingBlockIndex parameter, which of the two is used?

The NextToken is used, and the StartingBlockIndex is ignored.

How long are the block tokens and next tokens valid?

Block tokens are valid for seven days, and next tokens are valid for 60 minutes.

Are encrypted snapshots supported?

Yes. Encrypted snapshots can be accessed using the EBS direct APIs.

To access an encrypted snapshot, the user must have access to the key used to encrypt the snapshot, and the AWS KMS decrypt action. See the [Permissions for IAM users \(p. 1080\)](#) section earlier in this guide for the AWS KMS policy to assign to a user.

Are public snapshots supported?

Public snapshots are not supported.

Does list snapshot block return all block indexes and block tokens in a snapshot, or only those that have data written to them?

It returns only block indexes and tokens that have data written to them.

Can I get a history of the API calls made by the EBS direct APIs on my account for security analysis and operational troubleshooting purposes?

Yes. To receive a history of EBS direct APIs API calls made on your account, turn on AWS CloudTrail in the AWS Management Console. For more information, see [Logging API Calls for the EBS direct APIs with AWS CloudTrail \(p. 1094\)](#).

Logging API Calls for the EBS direct APIs with AWS CloudTrail

The EBS direct APIs service is integrated with AWS CloudTrail. CloudTrail is a service that provides a record of actions taken by a user, role, or an AWS service in the EBS direct APIs. CloudTrail captures [StartSnapshot](#) and [CompleteSnapshot](#) API calls for the EBS direct APIs as events. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon Simple Storage Service (Amazon S3) bucket, including events for the EBS direct APIs. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in [Event history](#). You can use the information collected by CloudTrail to determine the request that was made to the EBS direct APIs, the IP address from which the request was made, who made the request, when it was made, and additional details.

For more information about CloudTrail, see the [AWS CloudTrail User Guide](#).

EBS direct APIs Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When supported event activity occurs in the EBS direct APIs, that activity is recorded in a CloudTrail event along with other AWS service events in [Event history](#). You can view, search, and download recent events in your AWS account. For more information, see [Viewing Events with CloudTrail Event History](#).

For an ongoing record of events in your AWS account, including events for the EBS direct APIs, create a trail. A *trail* enables CloudTrail to deliver log files to an S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- [Overview for Creating a Trail](#)
- [CloudTrail Supported Services and Integrations](#)
- [Configuring Amazon SNS Notifications for CloudTrail](#)
- [Receiving CloudTrail Log Files from Multiple Regions](#) and [Receiving CloudTrail Log Files from Multiple Accounts](#)

Supported API actions

The following API actions support logging as events in CloudTrail log files:

- [StartSnapshot](#)
- [CompleteSnapshot](#)

Identity information

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the [CloudTrail userIdentityElement](#).

Understanding EBS direct APIs Log File Entries

A trail is a configuration that enables delivery of events as log files to an S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following examples show CloudTrail log entries that demonstrates the `StartSnapshot` and `CompleteSnapshot` actions.

`StartSnapshot` example:

```
{  
    "eventVersion": "1.05",  
    "userIdentity": {  
        "type": "IAMUser",  
        "principalId": "123456789012",  
        "arn": "arn:aws:iam::123456789012:root",  
        "accountId": "123456789012",  
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",  
        "userName": "user"  
    },  
    "eventTime": "2020-07-03T23:27:26Z",  
    "eventSource": "ebs.amazonaws.com",  
    "eventName": "StartSnapshot",  
    "awsRegion": "eu-west-1",  
    "sourceIPAddress": "192.0.2.0",  
    "userAgent": "PostmanRuntime/7.25.0",  
    "requestParameters": {  
        "volumeSize": 8,  
        "clientToken": "token",  
        "encrypted": true  
    },  
    "responseElements": {  
        "snapshotId": "snap-123456789012",  
        "ownerId": "123456789012",  
        "status": "pending",  
        "startTime": "Jul 3, 2020 11:27:26 PM",  
        "volumeSize": 8,  
        "blockSize": 524288,  
        "kmsKeyArn": "HIDDEN_DUE_TO_SECURITY_REASONS"  
    },  
    "requestID": "be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",  
    "eventId": "6e12345-2a4e-417c-aa78-7594fEXAMPLE",  
}
```

```
        "eventType": "AwsApiCall",
        "recipientAccountId": "123456789012"
    }
```

CompleteSnapshot example:

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
    "eventTime": "2020-07-03T23:28:24Z",
    "eventSource": "ebs.amazonaws.com",
    "eventName": "CompleteSnapshot",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "PostmanRuntime/7.25.0",
    "requestParameters": {
        "snapshotId": "snap-123456789012",
        "changedBlocksCount": 5
    },
    "responseElements": {
        "status": "completed"
    },
    "requestID": "be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
    "eventId": "6e12345-2a4e-417c-aa78-7594fEXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

EBS direct APIs and interface VPC endpoints

You can establish a private connection between your VPC and EBS direct APIs by creating an *interface VPC endpoint*. Interface endpoints are powered by [AWS PrivateLink](#), a technology that enables you to privately access EBS direct APIs without an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to communicate with EBS direct APIs. Traffic between your VPC and EBS direct APIs does not leave the Amazon network.

Each interface endpoint is represented by one or more [Elastic Network Interfaces](#) in your subnets.

For more information, see [Interface VPC endpoints \(AWS PrivateLink\)](#) in the *Amazon VPC User Guide*.

Considerations for EBS direct APIs VPC endpoints

Before you set up an interface VPC endpoint for EBS direct APIs, ensure that you review [Interface endpoint properties and limitations](#) in the *Amazon VPC User Guide*.

VPC endpoint policies are not supported for EBS direct APIs. By default, full access to EBS direct APIs is allowed through the endpoint. However, you can control access to the interface endpoint using security groups. For more information, see [Controlling access to services with VPC endpoints](#) in the *Amazon VPC User Guide*.

Creating an interface VPC endpoint for EBS direct APIs

You can create a VPC endpoint for the EBS direct APIs service using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see [Creating an interface endpoint](#) in the *Amazon VPC User Guide*.

Create a VPC endpoint for EBS direct APIs using the following service name:

- com.amazonaws.*region*.ebs

If you enable private DNS for the endpoint, you can make API requests to EBS direct APIs using its default DNS name for the Region, for example, ebs.us-east-1.amazonaws.com. For more information, see [Accessing a service through an interface endpoint](#) in the *Amazon VPC User Guide*.

Idempotency for StartSnapshot API

Idempotency ensures that an API request completes only once. With an idempotent request, if the original request completes successfully. The subsequent retries return the result from the original successful request and they have no additional effect.

The [StartSnapshot](#) API supports idempotency using a *client token*. A client token is a unique string that you specify when you make an API request. If you retry an API request with the same client token and the same request parameters after it has completed successfully, the result of the original request is returned. If you retry a request with the same client token, but change one or more of the request parameters, the `ConflictException` error is returned.

If you do not specify your own client token, the AWS SDKs automatically generates a client token for the request to ensure that it is idempotent.

A client token can be any string that includes up to up to 64 ASCII characters. You should not reuse the same client tokens for different requests.

To make an idempotent StartSnapshot request with your own client token using the API

Specify the ClientToken request parameter.

```
POST /snapshots HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
User-Agent: <User agent parameter>
X-Amz-Date: 20200618T040724Z
Authorization: <Authentication parameter>

{
    "VolumeSize": 8,
    "ParentSnapshot": snap-123EXAMPLE1234567,
    "ClientToken": "550e8400-e29b-41d4-a716-446655440000",
    "Timeout": 60
}
```

To make an idempotent StartSnapshot request with your own client token using the AWS CLI

Specify the `--client-token` request parameter.

```
$ aws ebs start-snapshot --region us-east-2 --volume-size 8 --parent-snapshot
snap-123EXAMPLE1234567 --timeout 60 --client-token 550e8400-e29b-41d4-a716-446655440000
```

Automating the Amazon EBS snapshot lifecycle

You can use Amazon Data Lifecycle Manager to automate the creation, retention, and deletion of snapshots that you use to back up your Amazon EBS volumes. When you automate snapshot management, it helps you to:

- Protect valuable data by enforcing a regular backup schedule.

- Retain backups as required by auditors or internal compliance.
- Reduce storage costs by deleting outdated backups.

When combined with the monitoring features of Amazon CloudWatch Events and AWS CloudTrail, Amazon Data Lifecycle Manager provides a complete backup solution for EBS volumes at no additional cost.

Contents

- [How Amazon Data Lifecycle Manager works \(p. 1098\)](#)
- [Policy schedules \(p. 1099\)](#)
- [Considerations for Amazon Data Lifecycle Manager \(p. 1100\)](#)
- [Prerequisites \(p. 1101\)](#)
- [Manage backups using the console \(p. 1103\)](#)
- [Manage backups using the AWS CLI \(p. 1105\)](#)
- [Manage backups using the API \(p. 1108\)](#)
- [Monitor the snapshot lifecycle \(p. 1108\)](#)

How Amazon Data Lifecycle Manager works

The following are the key elements of Amazon Data Lifecycle Manager.

Elements

- [Snapshots \(p. 1098\)](#)
- [Target resource tags \(p. 1098\)](#)
- [Snapshot tags \(p. 1098\)](#)
- [Lifecycle policies \(p. 1099\)](#)

Snapshots

Snapshots are the primary means to back up data from your EBS volumes. To save storage costs, successive snapshots are incremental, containing only the volume data that changed since the previous snapshot. When you delete one snapshot in a series of snapshots for a volume, only the data unique to that snapshot is removed. The rest of the captured history of the volume is preserved.

For more information, see [Amazon EBS snapshots \(p. 1060\)](#).

Target resource tags

Amazon Data Lifecycle Manager uses resource tags to identify the EBS volumes to back up. Tags are customizable metadata that you can assign to your AWS resources (including EBS volumes and snapshots). An Amazon Data Lifecycle Manager policy (described later) targets a volume for backup using a single tag. Multiple tags can be assigned to a volume if you want to run multiple policies on it.

You can't use a '\' or '=' character in a tag key.

For more information, see [Tagging your Amazon EC2 resources \(p. 1229\)](#).

Snapshot tags

Amazon Data Lifecycle Manager applies the following tags to all snapshots created by a policy, to distinguish them from snapshots created by any other means:

- `aws:dlm:lifecycle-policy-id`
- `aws:dlm:lifecycle-schedule-name`

- `aws:dlm:expirationTime`
- `dlm:managed`

You can also specify custom tags to be applied to snapshots on creation.

You can't use a '\' or '=' character in a tag key.

The target tags that Amazon Data Lifecycle Manager uses to associate volumes with a policy can optionally be applied to snapshots created by the policy.

Lifecycle policies

A lifecycle policy consists of these core settings:

- Resource type—Defines the type of AWS resource managed by the policy. Use `VOLUME` to create snapshots of individual volumes or use `INSTANCE` to create multi-volume snapshots from the volumes for an instance. For more information, see [Multi-volume snapshots \(p. 1064\)](#).
- Target tags—Specifies the tags that must be associated with an EBS volume or an Amazon EC2 instance for it to be managed by the policy.
- Schedules—The start times and intervals for creating snapshots. The first snapshot is created by a policy within one hour after the specified start time. Subsequent snapshots are created within one hour of their scheduled time. A policy can have up to four schedules; one mandatory schedule and up to three optional schedules. For more information, see [Policy schedules \(p. 1099\)](#).
- Retention—Specifies how snapshots are retained. You can retain snapshots based on either the total count of snapshots or the age of each snapshot.

For example, you could create a policy that manages all EBS volumes that have a tag with a key of `account` and a value of `finance`, creates snapshots every 24 hours at 0900 UTC, and retains the five most recent snapshots. Snapshot creation would start by 0959 each day.

Policy schedules

Policy schedules define when snapshots are created by the policy. Policies can have up to four schedules—one mandatory schedule and up to three optional schedules.

Adding multiple schedules to a single policy lets you create snapshots at different frequencies using the same policy. For example, you can create a single policy that creates daily, weekly, monthly, and yearly snapshots. This eliminates the need to manage multiple policies.

For each schedule, you can define the frequency, fast snapshot restore settings, cross-Region copy rules, and tags. The tags that are assigned to a schedule are automatically assigned to the snapshots that are created when the schedule is triggered. In addition, Amazon Data Lifecycle Manager automatically assigns a system-generated tag to each snapshot based on the schedule's frequency.

Each schedule is triggered individually based on its frequency. If multiple schedules are triggered at the same time, Amazon Data Lifecycle Manager creates only one snapshot and uses the snapshot retention settings of the schedule that has the highest retention period. The tags of all of the triggered schedules are applied to the snapshot.

- If more than one of the triggered schedules is enabled for fast snapshot restore, then the snapshot is enabled for fast snapshot restore in all of the Availability Zones specified across all of the triggered schedules, and the highest retention settings of the triggered schedules is used for each Availability Zone.
- If more than one of the triggered schedules is enabled for cross-Region copy, the snapshot is copied to all Regions specified across the triggered schedules, and the highest retention period of the triggered schedules is used.

Considerations for Amazon Data Lifecycle Manager

Your AWS account has the following quotas related to Amazon Data Lifecycle Manager:

- You can create up to 100 lifecycle policies per Region.
- You can add up to 45 tags per resource.
- You can create up to four schedules per lifecycle policy.

The following considerations apply to lifecycle policies:

- A policy does not begin creating snapshots until you set its activation status to *enabled*. You can configure a policy to be enabled upon creation.
- The first snapshot is created by a policy within one hour after the specified start time. Subsequent snapshots are created within one hour of their scheduled time.
- If you modify a policy by removing or changing its target tag, the EBS volumes with that tag are no longer affected by the policy.
- If you modify the schedule name for a policy, the snapshots created under the old schedule name are no longer affected by the policy.
- If you modify a retention schedule based on time to use a new time interval, the new interval is used only for new snapshots. The new schedule does not affect the retention schedule of existing snapshots created by this policy.
- You cannot change the retention schedule of a policy from the count of snapshots to the age of each snapshot. To make this change, you must create a new policy.
- If you disable a policy with a retention schedule based on the age of each snapshot, the snapshots whose retention periods expire while the policy is disabled are retained indefinitely. You must delete these snapshots manually. When you enable the policy again, Amazon Data Lifecycle Manager resumes deleting snapshots as their retention periods expire.
- If you delete the resource to which a policy with count-based retention applies, the policy no longer manages the previously created snapshots. You must manually delete the snapshots if they are no longer needed.
- If you delete the resource to which a policy with age-based retention applies, the policy continues to delete snapshots on the defined schedule, up to the last snapshot. You must manually delete the last snapshot if it is no longer needed.
- You can create multiple policies to back up an EBS volume or an Amazon EC2 instance. For example, if an EBS volume has two tags, where tag A is the target for policy A to create a snapshot every 12 hours, and tag B is the target for policy B to create a snapshot every 24 hours, Amazon Data Lifecycle Manager creates snapshots according to the schedules for both policies.

The following considerations apply to lifecycle policies and [fast snapshot restore \(p. 1132\)](#):

- A snapshot that is enabled for fast snapshot restore remains enabled even if you delete or disable the lifecycle policy, disable fast snapshot restore for the lifecycle policy, or disable fast snapshot restore for the Availability Zone. You can disable fast snapshot restore for these snapshots manually.
- If you enable fast snapshot restore and you exceed the maximum number of snapshots that can be enabled for fast snapshot restore, Amazon Data Lifecycle Manager creates snapshots as scheduled but does not enable them for fast snapshot restore. After a snapshot that is enabled for fast snapshot restore is deleted, the next snapshot that Amazon Data Lifecycle Manager creates is enabled for fast snapshot restore.
- When you enable fast snapshot restore for a snapshot, it takes 60 minutes per TiB to optimize the snapshot. We recommend that you create a schedule that ensures that each snapshot is fully optimized before Amazon Data Lifecycle Manager creates the next snapshot.

- You are billed for each minute that fast snapshot restore is enabled for a snapshot in a particular Availability Zone. Charges are pro-rated with a minimum of one hour. For more information, see [Pricing and Billing \(p. 1135\)](#).

Note

Depending on the configuration of your lifecycle policies, you could have multiple snapshots enabled for fast snapshot restore simultaneously.

The following considerations apply to lifecycle policies and [Multi-Attach \(p. 1043\)](#) enabled volumes:

- When creating a lifecycle policy based on instance tags for Multi-Volume snapshots, Amazon Data Lifecycle Manager initiates a snapshot of the volume for each attached instance. Use the *timestamp* tag to identify the set of time-consistent snapshots that are created from the attached instances.

Prerequisites

The following prerequisites are required by Amazon Data Lifecycle Manager.

Prerequisites

- [Permissions for Amazon Data Lifecycle Manager \(p. 1101\)](#)
- [Permissions for IAM users \(p. 1102\)](#)
- [Permissions for encrypted snapshots \(p. 1102\)](#)

Permissions for Amazon Data Lifecycle Manager

Amazon Data Lifecycle Manager uses an IAM role to get the permissions that are required to manage snapshots on your behalf. Amazon Data Lifecycle Manager creates **AWSDataLifecycleManagerDefaultRole** the first time that you create a lifecycle policy using the AWS Management Console. You can also create this role using the following `create-default-role` command.

```
aws dlm create-default-role
```

Alternatively, you can create a custom IAM role with the required permissions and select it when you create a lifecycle policy.

To create a custom IAM role

1. Create a role with the following permissions.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateSnapshot",  
                "ec2:CreateSnapshots",  
                "ec2:DeleteSnapshot",  
                "ec2:DescribeVolumes",  
                "ec2:DescribeInstances",  
                "ec2:DescribeSnapshots"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iam:PassRole"  
            ]  
        }  
    ]  
}
```

```
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:::snapshot/*"
}
]
```

For more information, see [Creating a Role](#) in the *IAM User Guide*.

2. Add a trust relationship to the role.
 - a. In the IAM console, choose **Roles**.
 - b. Select the role you created and then choose **Trust relationships**.
 - c. Choose **Edit Trust Relationship**, add the following policy, and then choose **Update Trust Policy**.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "dlm.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Permissions for IAM users

An IAM user must have the following permissions to use Amazon Data Lifecycle Manager.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["iam:PassRole", "iam>ListRoles"],
            "Resource": "arn:aws:iam::123456789012:role/AWSDataLifecycleManagerDefaultRole"
        },
        {
            "Effect": "Allow",
            "Action": "dlm:*",
            "Resource": "*"
        }
    ]
}
```

For more information, see [Changing Permissions for an IAM User](#) in the *IAM User Guide*.

Permissions for encrypted snapshots

To copy an encrypted snapshot between Regions, you must have access to both the source and destination customer master key (CMK) from AWS Key Management Service (AWS KMS).

If the source volume is encrypted, ensure that **AWSDataLifecycleManagerDefaultRole** has permission to use the CMK used to encrypt the volume. If you enable **Cross Region copy** and choose to encrypt the copied snapshot, ensure that **AWSDataLifecycleManagerDefaultRole** has permission to use the CMK needed to encrypt the snapshot in the destination Region. For more information, see [Managing access to AWS KMS CMKs](#) in the *AWS Key Management Service Developer Guide*.

Manage backups using the console

The following examples show how to use Amazon Data Lifecycle Manager to manage the backups of your EBS volumes using the AWS Management Console.

Tasks

- [Create a lifecycle policy \(p. 1103\)](#)
- [View a lifecycle policy \(p. 1104\)](#)
- [Modify a lifecycle policy \(p. 1104\)](#)
- [Delete a lifecycle policy \(p. 1104\)](#)

Create a lifecycle policy

Use the following procedure to create a lifecycle policy.

To create a lifecycle policy

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic Block Store, Lifecycle Manager**, and then choose **Create snapshot lifecycle policy**.
3. Provide the following information for your policy as needed:
 - **Description**—A description of the policy.
 - **Resource type**—The type of resource to back up. Use **Volume** to create snapshots of individual volumes or use **Instance** to create multi-volume snapshots from the volumes for an instance.
 - **Target with these tags**—The resource tags that identify the volumes or instances to back up.
 - **Lifecycle policy tags**—The tags for the lifecycle policy.
4. For **IAM role**, choose the IAM role that has permissions to create, delete, and describe snapshots, and to describe volumes. AWS provides a default role, **AWSDataLifecycleManagerDefaultRole**, or you can create a custom IAM role.
5. Add the policy schedules. Schedule 1 is mandatory. Schedules 2, 3, and 4 are optional. For each policy schedule, specify the following information:
 - **Schedule name**—A name for the schedule.
 - **Frequency**—The interval between policy runs. You can configure policy runs on a daily, weekly, monthly, or yearly schedule. Alternatively, choose **Custom cron expression** to specify an interval of up to 1 year. For more information, see [Cron expressions](#) in the *Amazon CloudWatch Events User Guide*.
 - **Starting at hh:mm UTC**—The time at which the policy runs are scheduled to start. The first policy run starts within an hour after the scheduled time.
 - **Retention type**—You can retain snapshots based on either the total count of snapshots or the age of each snapshot. For retention based on the count, the range is 1 to 1000. After the maximum count is reached, the oldest snapshot is deleted when a new one is created. For age-based retention, the range is 1 day to 100 years. After the retention period of each snapshot expires, it is deleted. The retention period should be greater than or equal to the creation interval.

Note

All schedules must have the same retention type. You can specify the retention type for Schedule 1 only. Schedules 2, 3, and 4 inherit the retention type from Schedule 1. Each schedule can have its own retention count or period.

- **Tagging information**—Choose whether to copy all user-defined tags on a source volume to the snapshots that are created by this policy. You can also specify additional tags for the snapshots in addition to the tags applied by Amazon Data Lifecycle Manager. If the resource type is **INSTANCE**,

you can choose to automatically tag your snapshots with the following variable tags: `instance-id` and `timestamp`. The values of the variable tags are determined when the tags are added.

- **Fast snapshot restore**—Choose whether to enable fast snapshot restore for all snapshots that are created by this policy. If you enable fast snapshot restore, you must choose the Availability Zones in which to enable it. You are billed for each minute that fast snapshot restore is enabled for a snapshot in a particular Availability Zone. Charges are pro-rated with a minimum of one hour. You can also specify the maximum number of snapshots that can be enabled for fast snapshot restore.
 - **Enable cross Region copy**—You can copy each snapshot to up to three additional Regions. You must ensure that you do not exceed the number of concurrent snapshot copies per Region. For each Region, you can choose different retention policies and whether to copy all tags or no tags. If the source snapshot is encrypted or if encryption is enabled by default, the snapshot copies are encrypted. If the source snapshot is unencrypted, you can enable encryption. If you do not specify a CMK, the snapshots are encrypted using the default key for EBS encryption in each destination Region. If you specify a CMK for the destination Region, you must have access to the CMK.
6. For **Policy status after creation**, choose **Enable policy** to start the policy runs at the next scheduled time, or **Disable policy** to prevent the policy from running.
 7. Choose **Create Policy**.

[View a lifecycle policy](#)

Use the following procedure to view a lifecycle policy.

To view a lifecycle policy

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic Block Store, Lifecycle Manager**.
3. Select a lifecycle policy from the list. The **Details** tab displays information about the policy.

[Modify a lifecycle policy](#)

Use the following procedure to modify a lifecycle policy.

To modify a lifecycle policy

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic Block Store, Lifecycle Manager**.
3. Select a lifecycle policy from the list.
4. Choose **Actions, Modify Snapshot Lifecycle Policy**.
5. Modify the policy settings as needed. For example, you can modify the schedule, add or remove tags, or enable or disable the policy.
6. Choose **Update policy**.

[Delete a lifecycle policy](#)

Use the following procedure to delete a lifecycle policy.

Note

You can delete snapshots created only by Amazon Data Lifecycle Manager.

To delete a lifecycle policy

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic Block Store, Lifecycle Manager**.
3. Select a lifecycle policy from the list.

4. Choose **Actions, Delete Snapshot Lifecycle Policy**.
5. When prompted for confirmation, choose **Delete Snapshot Lifecycle Policy**.

Manage backups using the AWS CLI

The following examples show how to use Amazon Data Lifecycle Manager to manage the backups of your EBS volumes using the AWS CLI.

Examples

- [Create a lifecycle policy \(p. 1105\)](#)
- [Display a lifecycle policy \(p. 1106\)](#)
- [Modify a lifecycle policy \(p. 1107\)](#)
- [Delete a lifecycle policy \(p. 1108\)](#)

Create a lifecycle policy

Use the `create-lifecycle-policy` command to create a lifecycle policy. To simplify the syntax, the example uses a JSON file, `policyDetails.json`, that includes the policy details.

This example uses a resource type of `VOLUME` to create snapshots of all volumes with the specified target tags. To create snapshots of all volumes for all instances with the specified target tags, use a resource type of `INSTANCE` instead. The policy includes two schedules. The first schedule creates a snapshot every day at 03:00 UTC. The second schedule creates a weekly snapshot every Friday at 17:00 UTC.

```
aws dlm create-lifecycle-policy --description "My volume policy" --state ENABLED --  
execution-role-arn arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole --  
policy-details file:///policyDetails.json
```

The following is an example of the `policyDetails.json` file.

```
{  
    "ResourceTypes": [  
        "VOLUME"  
    ],  
    "TargetTags": [{  
        "Key": "costcenter",  
        "Value": "115"  
    }],  
    "Schedules": [{  
        "Name": "DailySnapshots",  
        "TagsToAdd": [{  
            "Key": "type",  
            "Value": "myDailySnapshot"  
        }],  
        "CreateRule": {  
            "Interval": 24,  
            "IntervalUnit": "HOURS",  
            "Times": [  
                "03:00"  
            ]  
        },  
        "RetainRule": {  
            "Count": 5  
        },  
        "CopyTags": false  
    },  
    {  
        "Name": "WeeklySnapshots",  
        "TagsToAdd": [{  
            "Key": "type",  
            "Value": "myWeeklySnapshot"  
        }],  
        "CreateRule": {  
            "Interval": 168,  
            "IntervalUnit": "WEEKS",  
            "Times": [  
                "17:00"  
            ]  
        },  
        "RetainRule": {  
            "Count": 1  
        },  
        "CopyTags": false  
    }]
```

```
"TagsToAdd": [
    "Key": "type",
    "Value": "myWeeklySnapshot"
],
"CreateRule": {
    "CronExpression": "cron(0 0 17 ? * FRI *)"
},
"RetainRule": {
    "Count": 5
},
"CopyTags": false
]
}
```

Upon success, the command returns the ID of the newly created policy. The following is example output.

```
{
    "PolicyId": "policy-0123456789abcdef0"
}
```

Display a lifecycle policy

Use the [get-lifecycle-policy](#) command to display information about a lifecycle policy.

```
aws dlm get-lifecycle-policy --policy-id policy-0123456789abcdef0
```

The following is example output. It includes the information that you specified, plus metadata inserted by AWS.

```
{
    "Policy": {
        "Description": "My first policy",
        "DateCreated": "2018-05-15T00:16:21+0000",
        "State": "ENABLED",
        "ExecutionRoleArn": "arn:aws:iam::210774411744:role/AWSDataLifecycleManagerDefaultRole",
        "PolicyId": "policy-0123456789abcdef0",
        "DateModified": "2018-05-15T00:16:22+0000",
        "PolicyDetails": {
            "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
            "ResourceTypes": [
                "VOLUME"
            ],
            "TargetTags": [
                {
                    "Value": "115",
                    "Key": "costcenter"
                }
            ],
            "Schedules": [
                {
                    "TagsToAdd": [
                        {
                            "Value": "myDailySnapshot",
                            "Key": "type"
                        }
                    ],
                    "RetainRule": {
                        "Count": 5
                    },
                    "CopyTags": false,
                    "CronExpression": "cron(0 0 17 ? * FRI *)"
                }
            ]
        }
    }
}
```

```
        "CreateRule": {
            "Interval": 24,
            "IntervalUnit": "HOURS",
            "Times": [
                "03:00"
            ]
        },
        "Name": "DailySnapshots"
    }
}
}
```

Modify a lifecycle policy

Use the [update-lifecycle-policy](#) command to modify the information in a lifecycle policy. To simplify the syntax, this example references a JSON file, `policyDetailsUpdated.json`, that includes the policy details.

```
aws dlm update-lifecycle-policy --state DISABLED --execution-role-arn
arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole" --policy-details
file://policyDetailsUpdated.json
```

The following is an example of the `policyDetailsUpdated.json` file.

```
{
    "ResourceTypes": [
        "VOLUME"
    ],
    "TargetTags": [
        {
            "Key": "costcenter",
            "Value": "120"
        }
    ],
    "Schedules": [
        {
            "Name": "DailySnapshots",
            "TagsToAdd": [
                {
                    "Key": "type",
                    "Value": "myDailySnapshot"
                }
            ],
            "CreateRule": {
                "Interval": 12,
                "IntervalUnit": "HOURS",
                "Times": [
                    "15:00"
                ]
            },
            "RetainRule": {
                "Count": 5
            },
            "CopyTags": false
        }
    ]
}
```

To view the updated policy, use the [get-lifecycle-policy](#) command. You can see that the state, the value of the tag, the snapshot interval, and the snapshot start time were changed.

Delete a lifecycle policy

Use the [delete-lifecycle-policy](#) command to delete a lifecycle policy and free up the target tags specified in the policy for reuse.

Note

You can delete snapshots created only by Amazon Data Lifecycle Manager.

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

Manage backups using the API

The [Amazon Data Lifecycle Manager API Reference](#) provides descriptions and syntax for each of the actions and data types for the Amazon Data Lifecycle Manager Query API.

Alternatively, you can use one of the AWS SDKs to access the API in a way that's tailored to the programming language or platform that you're using. For more information, see [AWS SDKs](#).

Monitor the snapshot lifecycle

You can use the following features to monitor the lifecycle of your snapshots.

Features

- [Console and AWS CLI \(p. 1108\)](#)
- [CloudWatch Events \(p. 1108\)](#)
- [AWS CloudTrail \(p. 1109\)](#)

Console and AWS CLI

You can view your lifecycle policies using the Amazon EC2 console or the AWS CLI. Each snapshot created by a policy has a timestamp and policy-related tags. You can filter snapshots using tags to verify that your backups are being created as you intend. For information about viewing lifecycle policies using the console, see [View a lifecycle policy \(p. 1104\)](#). For information about displaying information about lifecycle policies using the CLI, see [Display a lifecycle policy \(p. 1106\)](#).

CloudWatch Events

Amazon EBS and Amazon Data Lifecycle Manager emit events related to lifecycle policy actions. You can use AWS Lambda and Amazon CloudWatch Events to handle event notifications programmatically. For more information, see the [Amazon CloudWatch Events User Guide](#).

The following events are available:

- **createSnapshot**—An Amazon EBS event emitted when a `CreateSnapshot` action succeeds or fails. For more information, see [Amazon CloudWatch Events for Amazon EBS \(p. 1178\)](#).
- **DLM Policy State Change**—An Amazon Data Lifecycle Manager event emitted when a lifecycle policy enters an error state. The event contains a description of what caused the error. The following is an example of an event when the permissions granted by the IAM role are insufficient.

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-0123456789ab",  
    "detail-type": "DLM Policy State Change",  
    "source": "aws.dlm",  
    "account": "123456789012",  
    "time": "2018-05-25T13:12:22Z",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"
```

```
],
  "detail": {
    "state": "ERROR",
    "cause": "Role provided does not have sufficient permissions",
    "policy_id": "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"
  }
}
```

The following is an example of an event when a limit is exceeded.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "DLM Policy State Change",
  "source": "aws.dlm",
  "account": "123456789012",
  "time": "2018-05-25T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"
  ],
  "detail":{
    "state": "ERROR",
    "cause": "Maximum allowed active snapshot limit exceeded",
    "policy_id": "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"
  }
}
```

AWS CloudTrail

With AWS CloudTrail, you can track user activity and API usage to demonstrate compliance with internal policies and regulatory standards. For more information, see the [AWS CloudTrail User Guide](#).

Amazon EBS data services

Amazon EBS provides the following data services.

Data services

- [Amazon EBS Elastic Volumes \(p. 1109\)](#)
- [Amazon EBS encryption \(p. 1121\)](#)
- [Amazon EBS fast snapshot restore \(p. 1132\)](#)

Amazon EBS Elastic Volumes

With Amazon EBS Elastic Volumes, you can increase the volume size, change the volume type, or adjust the performance of your EBS volumes. If your instance supports Elastic Volumes, you can do so without detaching the volume or restarting the instance. This enables you to continue using your application while the changes take effect.

There is no charge to modify the configuration of a volume. You are charged for the new volume configuration after volume modification starts. For more information, see the [Amazon EBS Pricing](#) page.

Contents

- [Requirements when modifying volumes \(p. 1110\)](#)
- [Requesting modifications to your EBS Volumes \(p. 1111\)](#)
- [Monitoring the progress of volume modifications \(p. 1114\)](#)

- [Extending a Linux file system after resizing a volume \(p. 1117\)](#)

Requirements when modifying volumes

The following requirements and limitations apply when you modify an Amazon EBS volume. To learn more about the general requirements for EBS volumes, see [Constraints on the size and configuration of an EBS volume \(p. 1037\)](#).

Supported instance types

Elastic Volumes are supported on the following instances:

- All [current-generation instances \(p. 201\)](#)
- The following previous-generation instances: C1, C3, CC2, CR1, G2, I2, M1, M3, and R3

If your instance type does not support Elastic Volumes, see [Modifying an EBS volume if Elastic Volumes is not supported \(p. 1114\)](#).

Requirements for Linux volumes

Linux AMIs require a GUID partition table (GPT) and GRUB 2 for boot volumes that are 2 TiB (2,048 GiB) or larger. Many Linux AMIs today still use the MBR partitioning scheme, which only supports boot volume sizes up to 2 TiB. If your instance does not boot with a boot volume larger than 2 TiB, the AMI you are using may be limited to a boot volume size of less than 2 TiB. Non-boot volumes do not have this limitation on Linux instances. For requirements affecting Windows volumes, see [Requirements for Windows Volumes in the Amazon EC2 User Guide for Windows Instances](#).

Before attempting to resize a boot volume beyond 2 TiB, you can determine whether the volume is using MBR or GPT partitioning by running the following command on your instance:

```
[ec2-user ~]$ sudo gdisk -l /dev/xvda
```

An Amazon Linux instance with GPT partitioning returns the following information:

```
GPT fdisk (gdisk) version 0.8.10

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.
```

A SUSE instance with MBR partitioning returns the following information:

```
GPT fdisk (gdisk) version 0.8.8

Partition table scan:
  MBR: MBR only
  BSD: not present
  APM: not present
  GPT: not present
```

Limitations

- Elastic Volume operations are not supported on Multi-Attach enabled Amazon EBS volumes.

- The new volume size cannot exceed the supported volume capacity. For more information, see [Constraints on the size and configuration of an EBS volume \(p. 1037\)](#).
- If the volume was attached before November 3, 2016 23:40 UTC, you must initialize Elastic Volumes support. For more information, see [Initializing Elastic Volumes Support \(p. 1113\)](#).
- If you are using an unsupported previous-generation instance type, or if you encounter an error while attempting a volume modification, see [Modifying an EBS volume if Elastic Volumes is not supported \(p. 1114\)](#).
- A gp2 volume that is attached to an instance as a root volume cannot be modified to an st1 or sc1 volume. If detached and modified to st1 or sc1, it cannot be attached to an instance as the root volume.
- A gp2 volume cannot be modified to an st1 or sc1 volume if the requested volume size is below the minimum size for st1 and sc1 volumes.
- In some cases, you must detach the volume or stop the instance for modification to proceed. If you encounter an error message while attempting to modify an EBS volume, or if you are modifying an EBS volume attached to a previous-generation instance type, take one of the following steps:
 - For a non-root volume, detach the volume from the instance, apply the modifications, and then re-attach the volume.
 - For a root (boot) volume, stop the instance, apply the modifications, and then restart the instance.
- After provisioning over 32,000 IOPS on an existing io1 or io2 volume, you may need to do one of the following to see the full performance improvements:
 - Detach and attach the volume.
 - Restart the instance.
- Decreasing the size of an EBS volume is not supported. However, you can create a smaller volume and then migrate your data to it using an application-level tool such as rsync.
- Modification time is increased if you modify a volume that has not been fully initialized. For more information see [Initializing Amazon EBS volumes \(p. 1161\)](#).
- After modifying a volume, wait at least six hours and ensure that the volume is in the `in-use` or `available` state before making additional modifications to the same volume.
- While m3.medium instances fully support volume modification, m3.large, m3.xlarge, and m3.2xlarge instances might not support all volume modification features.

Requesting modifications to your EBS Volumes

With Elastic Volumes, you can dynamically modify the size, performance, and volume type of your Amazon EBS volumes without detaching them.

Use the following process when modifying a volume:

1. (Optional) Before modifying a volume that contains valuable data, it is a best practice to create a snapshot of the volume in case you need to roll back your changes. For more information, see [Creating Amazon EBS snapshots \(p. 1063\)](#).
2. Request the volume modification.
3. Monitor the progress of the volume modification. For more information, see [Monitoring the progress of volume modifications \(p. 1114\)](#).
4. If the size of the volume was modified, extend the volume's file system to take advantage of the increased storage capacity. For more information, see [Extending a Linux file system after resizing a volume \(p. 1117\)](#).

Contents

- [Modifying an EBS volume using Elastic Volumes \(console\) \(p. 1112\)](#)

- [Modifying an EBS volume using Elastic Volumes \(AWS CLI\) \(p. 1112\)](#)
- [Initializing Elastic Volumes support \(if needed\) \(p. 1113\)](#)
- [Modifying an EBS volume if Elastic Volumes is not supported \(p. 1114\)](#)

Modifying an EBS volume using Elastic Volumes (console)

Use the following procedure to modify an EBS volume.

To modify an EBS volume using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Volumes**, select the volume to modify, and then choose **Actions, Modify Volume**.
3. The **Modify Volume** window displays the volume ID and the volume's current configuration, including type, size, and IOPS. You can change any or all of these settings in a single action. Set new configuration values as follows:
 - To modify the type, choose a value for **Volume Type**.
 - To modify the size, enter an allowed integer value for **Size**.
 - If you chose **Provisioned IOPS SSD (io1)** or **Provisioned IOPS SSD (io2)** as the volume type, enter an allowed integer value for **IOPS**.
4. After you have finished changing the volume settings, choose **Modify**. When prompted for confirmation, choose **Yes**.
5. Modifying volume size has no practical effect until you also extend the volume's file system to make use of the new storage capacity. For more information, see [Extending a Linux file system after resizing a volume \(p. 1117\)](#).

Modifying an EBS volume using Elastic Volumes (AWS CLI)

Use the `modify-volume` command to modify one or more configuration settings for a volume. For example, if you have a volume of type gp2 with a size of 100 GiB, the following command changes its configuration to a volume of type io1 with 10,000 IOPS and a size of 200 GiB.

```
aws ec2 modify-volume --volume-type io1 --iops 10000 --size 200 --volume-id vol-1111111111111111
```

The following is example output:

```
{  
    "VolumeModification": {  
        "TargetSize": 200,  
        "TargetVolumeType": "io1",  
        "ModificationState": "modifying",  
        "VolumeId": "vol-1111111111111111",  
        "TargetIops": 10000,  
        "StartTime": "2017-01-19T22:21:02.959Z",  
        "Progress": 0,  
        "OriginalVolumeType": "gp2",  
        "OriginalIops": 300,  
        "OriginalSize": 100  
    }  
}
```

Modifying volume size has no practical effect until you also extend the volume's file system to make use of the new storage capacity. For more information, see [Extending a Linux file system after resizing a volume \(p. 1117\)](#).

Initializing Elastic Volumes support (if needed)

Before you can modify a volume that was attached to an instance before November 3, 2016 23:40 UTC, you must initialize volume modification support using one of the following actions:

- Detach and attach the volume
- Stop and start the instance

Use one of the following procedures to determine whether your instances are ready for volume modification.

New console

To determine whether your instances are ready using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, choose **Instances**.
3. Choose the **Show/Hide Columns** icon (the gear). Select the **Launch time** attribute column and then choose **Confirm**.
4. Sort the list of instances by the **Launch Time** column. For each instance that was started before the cutoff date, choose the **Storage** tab and check the **Attachment time** column to see when its volumes were attached.

Old console

To determine whether your instances are ready using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, choose **Instances**.
3. Choose the **Show/Hide Columns** icon (the gear). Select the **Launch Time** and **Block Devices** attributes and then choose **Close**.
4. Sort the list of instances by the **Launch Time** column. For instances that were started before the cutoff date, check when the devices were attached. In the following example, you must initialize volume modification for the first instance because it was started before the cutoff date and its root volume was attached before the cutoff date. The other instances are ready because they were started after the cutoff date.

Instance ID	Launch Time	Block Devices
i-e905622e	February 25, 2016 at 1:49:35 PM UTC-8	/dev/xvda=vol-e6b46410 attached:2016-02-25T21:49:35.000Z:true
i-719f99a8	December 8, 2016 at 2:21:51 PM UTC-8	/dev/xvda=vol-bad60e7a attached:2016-01-15T18:36:12.000Z:true
i-006b02c1b78381e57	May 17, 2017 at 1:52:52 PM UTC-7	/dev/sda1=vol-0de9250441c73024c attached:2017-05-17T20:52:53.000Z:true, xvdb=vol-0863a86c393496d3d attached:2017-05-17T20:52:53.000Z:false
i-e3d172ed	May 17, 2017 at 2:48:54 PM UTC-7	/dev/sda1=vol-04c34d0b attached:2015-01-21T21:19:46.000Z:true

To determine whether your instances are ready using the CLI

Use the following [describe-instances](#) command to determine whether the volume was attached before November 3, 2016 23:40 UTC.

```
aws ec2 describe-instances --query "Reservations[*].Instances[*].[InstanceId,LaunchTime<='2016-11-01',BlockDeviceMappings[*][Ebs.AttachTime<='2016-11-01']]"
--output text
```

The first line of the output for each instance shows its ID and whether it was started before the cutoff date (True or False). The first line is followed by one or more lines that show whether each EBS volume was attached before the cutoff date (True or False). In the following example output, you must initialize

volume modification for the first instance because it was started before the cutoff date and its root volume was attached before the cutoff date. The other instances are ready because they were started after the cutoff date.

```
i-e905622e      True
True
i-719f99a8      False
True
i-006b02c1b78381e57  False
False
False
i-e3d172ed      False
True
```

Modifying an EBS volume if Elastic Volumes is not supported

If you are using a supported instance type, you can use Elastic Volumes to dynamically modify the size, performance, and volume type of your Amazon EBS volumes without detaching them.

If you cannot use Elastic Volumes but you need to modify the root (boot) volume, you must stop the instance, modify the volume, and then restart the instance.

After the instance has started, you can check the file system size to see if your instance recognizes the larger volume space. On Linux, use the **df -h** command to check the file system size.

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      7.9G  943M  6.9G  12% /
tmpfs           1.9G     0  1.9G   0% /dev/shm
```

If the size does not reflect your newly expanded volume, you must extend the file system of your device so that your instance can use the new space. For more information, see [Extending a Linux file system after resizing a volume \(p. 1117\)](#).

Monitoring the progress of volume modifications

When you modify an EBS volume, it goes through a sequence of states. The volume enters the modifying state, the optimizing state, and finally the completed state. At this point, the volume is ready to be further modified.

Note

Rarely, a transient AWS fault can result in a failed state. This is not an indication of volume health; it merely indicates that the modification to the volume failed. If this occurs, retry the volume modification.

While the volume is in the optimizing state, your volume performance is in between the source and target configuration specifications. Transitional volume performance will be no less than the source volume performance. If you are downgrading IOPS, transitional volume performance is no less than the target volume performance.

Volume modification changes take effect as follows:

- Size changes usually take a few seconds to complete and take effect after a volume is in the Optimizing state.
- Performance (IOPS) changes can take from a few minutes to a few hours to complete and are dependent on the configuration change being made.
- It might take up to 24 hours for a new configuration to take effect, and in some cases more, such as when the volume has not been fully initialized. Typically, a fully used 1-TiB volume takes about 6 hours to migrate to a new performance configuration.

Use one of the following methods to monitor the progress of a volume modification.

Contents

- [Monitoring the progress of a volume modification \(console\) \(p. 1115\)](#)
- [Monitoring the progress of a volume modification \(AWS CLI\) \(p. 1115\)](#)
- [Monitoring the progress of a volume modification \(CloudWatch Events\) \(p. 1116\)](#)

Monitoring the progress of a volume modification (console)

Use the following procedure to view the progress of one or more volume modifications.

To monitor progress of a modification using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. Select the volume.
4. The **State** column and the **State** field in the details pane contain information in the following format: *volume-state - modification-state (progress%)*. The possible volume states are **creating**, **available**, **in-use**, **deleting**, **deleted**, and **error**. The possible modification states are **modifying**, **optimizing**, and **completed**. Shortly after the volume modification is completed, we remove the modification state and progress, leaving only the volume state.

In this example, the modification state of the selected volume is **optimizing**. The modification state of the next volume is **modifying**.

Name	Volume ID	Size	Volume Type	IOPS	Snapshot	Created	Availability Zone	State
vol-0ddaa54cd90f15...	vol-0ddaa54cd90f15...	8 GiB	gp2	100	snap-09aa45c...	January 9, 2020 at ...	eu-west-1b	in-use
Windows-ins...	vol-02940f6ee433f...	16 GiB	gp2	100	snap-076d641...	January 9, 2020 at ...	eu-west-1c	in-use - optimizing (1%)
attach-vol-te...	vol-0b01f92a8e62...	8 GiB	gp2	100		October 11, 2019 at ...	eu-west-1a	available - modifying (0%)
	attach-vol-te...	100 GiB	gp2	300		January 30, 2019 at ...	eu-west-1b	available

5. Choose the text in the **State** field in the details pane to display information about the most recent modification action, as shown in the previous step.

Monitoring the progress of a volume modification (AWS CLI)

Use the `describe-volumes-modifications` command to view the progress of one or more volume modifications. The following example describes the volume modifications for two volumes.

```
aws ec2 describe-volumes-modifications --volume-id vol-1111111111111111 vol-2222222222222222
```

In the following example output, the volume modifications are still in the modifying state. Progress is reported as a percentage.

```
{  
    "VolumesModifications": [  
        {  
            "TargetSize": 200,  
            "TargetVolumeType": "io1",  
            "ModificationState": "modifying",  
            "VolumeId": "vol-1111111111111111",  
            "TargetIops": 10000,  
            "StartTime": "2017-01-19T22:21:02.959Z",  
            "Progress": 0,  
            "OriginalVolumeType": "gp2",  
            "OriginalIops": 300,  
            "OriginalSize": 100  
        },  
        {  
            "TargetSize": 2000,  
            "TargetVolumeType": "sc1",  
            "ModificationState": "modifying",  
            "VolumeId": "vol-2222222222222222",  
            "StartTime": "2017-01-19T22:23:22.158Z",  
            "Progress": 0,  
            "OriginalVolumeType": "gp2",  
            "OriginalIops": 300,  
            "OriginalSize": 1000  
        }  
    ]  
}
```

The next example describes all volumes with a modification state of either optimizing or completed, and then filters and formats the results to show only modifications that were initiated on or after February 1, 2017:

```
aws ec2 describe-volumes-modifications --filters Name=modification-state,Values="optimizing","completed" --query "VolumesModifications[?StartTime>='2017-02-01'].{ID:VolumeId,STATE:ModificationState}"
```

The following is example output with information about two volumes:

```
[  
    {  
        "STATE": "optimizing",  
        "ID": "vol-06397e7a0eEXAMPLE"  
    },  
    {  
        "STATE": "completed",  
        "ID": "vol-ba74e18c2aEXAMPLE"  
    }  
]
```

Monitoring the progress of a volume modification (CloudWatch Events)

With CloudWatch Events, you can create a notification rule for volume modification events. You can use your rule to generate a notification message using [Amazon SNS](#) or to invoke a [Lambda function](#) in response to matching events.

To monitor progress of a modification using CloudWatch Events

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.

2. Choose **Events, Create rule**.
3. For **Build event pattern to match events by service**, choose **Custom event pattern**.
4. For **Build custom event pattern**, replace the contents with the following and choose **Save**.

```
{  
    "source": [  
        "aws.ec2"  
    ],  
    "detail-type": [  
        "EBS Volume Notification"  
    ],  
    "detail": {  
        "event": [  
            "modifyVolume"  
        ]  
    }  
}
```

The following is example event data:

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "2017-01-12T21:09:07Z",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-east-1:012345678901:volume/vol-03a55cf56513fa1b6"  
    ],  
    "detail": {  
        "result": "optimizing",  
        "cause": "",  
        "event": "modifyVolume",  
        "request-id": "01234567-0123-0123-0123-0123456789ab"  
    }  
}
```

Extending a Linux file system after resizing a volume

After you [increase the size of an EBS volume \(p. 1111\)](#), you must use file system-specific commands to extend the file system to the larger size. You can resize the file system as soon as the volume enters the optimizing state.

Important

Before extending a file system that contains valuable data, it is best practice to create a snapshot of the volume, in case you need to roll back your changes. For more information, see [Creating Amazon EBS snapshots \(p. 1063\)](#). If your Linux AMI uses the MBR partitioning scheme, you are limited to a boot volume size of up to 2 TiB. For more information, see [Requirements for Linux volumes \(p. 1110\)](#) and [Constraints on the size and configuration of an EBS volume \(p. 1037\)](#).

The process for extending a file system on Linux is as follows:

1. Your EBS volume might have a partition that contains the file system and data. Increasing the size of a volume does not increase the size of the partition. Before you extend the file system on a resized volume, check whether the volume has a partition that must be extended to the new size of the volume.

2. Use a file system-specific command to resize each file system to the new volume capacity.

For information about extending a Windows file system, see [Extending a Windows File System after Resizing a Volume](#) in the *Amazon EC2 User Guide for Windows Instances*.

The following examples walk you through the process of extending a Linux file system. For file systems and partitioning schemes other than the ones shown here, refer to the documentation for those file systems and partitioning schemes for instructions.

Examples

- [Example: Extending the file system of NVMe EBS volumes \(p. 1118\)](#)
- [Example: Extending the file system of EBS volumes \(p. 1119\)](#)

Example: Extending the file system of NVMe EBS volumes

For this example, suppose that you have an instance built on the [Nitro System \(p. 205\)](#), such as an M5 instance. You resized the boot volume from 8 GB to 16 GB and an additional volume from 8 GB to 30 GB. Use the following procedure to extend the file system of the resized volumes.

To extend the file system of NVMe EBS volumes

1. [Connect to your instance \(p. 563\)](#).
2. To verify the file system for each volume, use the **df -hT** command.

```
[ec2-user ~]$ df -hT
```

The following is example output for an instance that has a boot volume with an XFS file system and an additional volume with an XFS file system. The naming convention `/dev/nvme[0-26]n1` indicates that the volumes are exposed as NVMe block devices.

```
[ec2-user ~]$ df -hT
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/nvme0n1p1  xfs   8.0G  1.6G  6.5G  20% /
/dev/nvme1n1    xfs   8.0G   33M  8.0G   1% /data
...
```

3. To check whether the volume has a partition that must be extended, use the **lsblk** command to display information about the NVMe block devices attached to your instance.

```
[ec2-user ~]$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme1n1    259:0    0  30G  0 disk /data
nvme0n1    259:1    0  16G  0 disk
##nvme0n1p1 259:2    0   8G  0 part /
##nvme0n1p128 259:3   0   1M  0 part
```

This example output shows the following:

- The root volume, `/dev/nvme0n1`, has a partition, `/dev/nvme0n1p1`. While the size of the root volume reflects the new size, 16 GB, the size of the partition reflects the original size, 8 GB, and must be extended before you can extend the file system.
 - The volume `/dev/nvme1n1` has no partitions. The size of the volume reflects the new size, 30 GB.
4. To extend the partition on the root volume, use the following **growpart** command. Notice that there is a space between the device name and the partition number.

```
[ec2-user ~]$ sudo growpart /dev/nvme0n1 1
```

5. (Optional) Use the **lsblk** command again to verify that the partition reflects the increased volume size.

```
[ec2-user ~]$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme1n1    259:0   0   30G  0 disk /
nvme0n1    259:1   0   16G  0 disk
##nvme0n1p1 259:2   0   16G  0 part /
##nvme0n1p128 259:3  0    1M  0 part
```

6. Use the **df -h** command to verify the size of the file system for each volume. In this example output, both file systems reflect the original volume size, 8 GB.

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme0n1p1  8.0G  1.6G  6.5G  20% /
/dev/nvme1n1    8.0G  33M  8.0G   1% /data
...
```

7. [XFS file system] Use the **xfs_growfs** command to extend the file system on each volume. In this example, / and /data are the volume mount points shown in the output for **df -h**.

```
[ec2-user ~]$ sudo xfs_growfs -d /
[ec2-user ~]$ sudo xfs_growfs -d /data
```

If the XFS tools are not already installed, you can install them as follows.

```
[ec2-user ~]$ sudo yum install xfsprogs
```

8. [ext4 file system] Use the **resize2fs** command to extend the file system on each volume.

```
[ec2-user ~]$ sudo resize2fs /dev/nvme0n1p1
[ec2-user ~]$ sudo resize2fs /dev/nvme1n1
```

9. (Optional) Use the **df -h** command again to verify that each file system reflects the increased volume size.

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme0n1p1  16G  1.6G  15G  10% /
/dev/nvme1n1    30G  33M  30G   1% /data
...
```

Example: Extending the file system of EBS volumes

For this example, suppose that you have resized the boot volume of an instance, such as a T2 instance, from 8 GB to 16 GB and an additional volume from 8 GB to 30 GB. Use the following procedure to extend the file system of the resized volumes.

To extend the file system of EBS volumes

1. [Connect to your instance \(p. 563\)](#).
2. To verify the file system in use for each volume, use the **df -hT** command.

```
[ec2-user ~]$ df -hT
```

The following is example output for an instance that has a boot volume with an ext4 file system and an additional volume with an XFS file system.

```
[ec2-user ~]$ df -hT
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/xvda1      ext4  8.0G  1.9G  6.2G  24% /
/dev/xvdf1      xfs   8.0G   45M  8.0G   1% /data
...
```

3. To check whether the volume has a partition that must be extended, use the **lsblk** command to display information about the block devices attached to your instance.

```
[ec2-user ~]$ lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda   202:0    0 16G  0 disk
##xvda1 202:1    0   8G  0 part /
xvdf   202:80   0 30G  0 disk
##xvdf1 202:81   0   8G  0 part /data
```

This example output shows the following:

- The root volume, `/dev/xvda`, has a partition, `/dev/xvda1`. While the size of the volume is 16 GB, the size of the partition is still 8 GB and must be extended.
 - The volume `/dev/xvdf` has a partition, `/dev/xvdf1`. While the size of the volume is 30G, the size of the partition is still 8 GB and must be extended.
4. To extend the partition on each volume, use the following **growpart** commands. Notice that there is a space between the device name and the partition number.

```
[ec2-user ~]$ sudo growpart /dev/xvda 1
[ec2-user ~]$ sudo growpart /dev/xvdf 1
```

5. (Optional) Use the **lsblk** command again to verify that the partitions reflect the increased volume size.

```
[ec2-user ~]$ lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda   202:0    0 16G  0 disk
##xvda1 202:1    0 16G  0 part /
xvdf   202:80   0 30G  0 disk
##xvdf1 202:81   0 30G  0 part /data
```

6. Use the **df -h** command to verify the size of the file system for each volume. In this example output, both file systems reflect the original volume size, 8 GB.

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      8.0G  1.9G  6.2G  24% /
/dev/xvdf1      8.0G   45M  8.0G   1% /data
...
```

7. [XFS volumes] Use the **xfs_growfs** command to extend the file system on each volume. In this example, `/` and `/data` are the volume mount points shown in the output for **df -h**.

```
[ec2-user ~]$ sudo xfs_growfs -d /
```

```
[ec2-user ~]$ sudo xfs_growfs -d /data
```

If the XFS tools are not already installed, you can install them as follows.

```
[ec2-user ~]$ sudo yum install xfsprogs
```

8. [ext4 volumes] Use the **resize2fs** command to extend the file system on each volume.

```
[ec2-user ~]$ sudo resize2fs /dev/xvda1
[ec2-user ~]$ sudo resize2fs /dev/xvdf1
```

9. (Optional) Use the **df -h** command again to verify that each file system reflects the increased volume size.

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1       16G  1.9G  14G  12% /
/dev/xvdf1       30G   45M  30G   1% /data
...
```

Amazon EBS encryption

Use Amazon EBS encryption as a straight-forward encryption solution for your EBS resources associated with your EC2 instances. With Amazon EBS encryption, you aren't required to build, maintain, and secure your own key management infrastructure. Amazon EBS encryption uses AWS Key Management Service (AWS KMS) customer master keys (CMK) when creating encrypted volumes and snapshots.

Encryption operations occur on the servers that host EC2 instances, ensuring the security of both data-at-rest and data-in-transit between an instance and its attached EBS storage.

You can attach both encrypted and unencrypted volumes to an instance simultaneously

Contents

- [How EBS encryption works \(p. 1121\)](#)
- [Requirements \(p. 1122\)](#)
- [Default key for EBS encryption \(p. 1123\)](#)
- [Encryption by default \(p. 1124\)](#)
- [Encrypting EBS resources \(p. 1124\)](#)
- [Encryption scenarios \(p. 1125\)](#)
- [Setting encryption defaults using the API and CLI \(p. 1131\)](#)

How EBS encryption works

You can encrypt both the boot and data volumes of an EC2 instance. When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

- Data at rest inside the volume
- All data moving between the volume and the instance
- All snapshots created from the volume
- All volumes created from those snapshots

EBS encrypts your volume with a data key using the industry-standard AES-256 algorithm. Your data key is stored on-disk with your encrypted data, but not before EBS encrypts it with your CMK. Your data

key never appears on disk in plaintext. The same data key is shared by snapshots of the volume and any subsequent volumes created from those snapshots. For more information, see [Data keys](#) in the *AWS Key Management Service Developer Guide*.

Amazon EBS works with AWS KMS to encrypt and decrypt your EBS volumes as follows:

1. Amazon EBS sends a [GenerateDataKeyWithoutPlaintext](#) request to AWS KMS, specifying the CMK that you chose for volume encryption.
2. AWS KMS generates a new data key, encrypts it under the CMK that you chose for volume encryption, and sends the encrypted data key to Amazon EBS to be stored with the volume metadata.
3. When you attach an encrypted volume to an instance, Amazon EC2 sends a [Decrypt](#) request to AWS KMS, specifying the encrypted data key.
4. Amazon EBS sends a [CreateGrant](#) request to AWS KMS, so that it can decrypt the data key.
5. AWS KMS decrypts the encrypted data key and sends the decrypted data key to Amazon EC2.
6. Amazon EC2 uses the plaintext data key in hypervisor memory to encrypt disk I/O to the volume. The plaintext data key persists in memory as long as the volume is attached to the instance.

For more information, see [How Amazon Elastic Block Store \(Amazon EBS\) uses AWS KMS](#) and [Amazon EC2 example two](#) in the *AWS Key Management Service Developer Guide*.

Requirements

Before you begin, verify that the following requirements are met.

Supported volume types

Encryption is supported by all EBS volume types. You can expect the same IOPS performance on encrypted volumes as on unencrypted volumes, with a minimal effect on latency. You can access encrypted volumes the same way that you access unencrypted volumes. Encryption and decryption are handled transparently, and they require no additional action from you or your applications.

Supported instance types

Amazon EBS encryption is available on all [current generation \(p. 201\)](#) instance types and the following [previous generation \(p. 204\)](#) instance types: C3, cr1.8xlarge, G2, I2, M3, and R3.

Permissions for IAM users

When you configure a CMK as the default key for EBS encryption, the default key policy allows any IAM user with access to the required KMS actions to use this key to encrypt or decrypt EBS resources. You must grant IAM users permission to call the following actions in order to use EBS encryption:

- `kms:CreateGrant`
- `kms:Decrypt`
- `kms:DescribeKey`
- `kms:GenerateDataKeyWithoutPlainText`
- `kms:ReEncrypt`

To follow the principle of least privilege, do not allow full access to `kms:CreateGrant`. Instead, allow the user to create grants on the CMK only when the grant is created on the user's behalf by an AWS service, as shown in the following example:

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": "kms>CreateGrant",
        "Resource": [
            "arn:aws:kms:us-east-2:123456789012:key/abcd1234-a123-456d-a12b-
a123b4cd56ef"
        ],
        "Condition": {
            "Bool": {
                "kms:GrantIsForAWSResource": true
            }
        }
    }
]
```

For more information, see [Allows access to the AWS account and enables IAM policies](#) in the **Default key policy** section in the *AWS Key Management Service Developer Guide*.

Default key for EBS encryption

Amazon EBS automatically creates a unique AWS managed CMK in each Region where you store AWS resources. This key has the alias `alias/aws/ebs`. By default, Amazon EBS uses this key for encryption. Alternatively, you can specify a symmetric customer managed CMK that you created as the default key for EBS encryption. Using your own CMK gives you more flexibility, including the ability to create, rotate, and disable keys.

Important

Amazon EBS does not support asymmetric CMKs. For more information, see [Using symmetric and asymmetric keys](#) in the *AWS Key Management Service Developer Guide*.

New console

To configure the default key for EBS encryption for a Region

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region.
3. From the navigation pane, select **EC2 Dashboard**.
4. In the upper-right corner of the page, choose **Account Attributes, EBS encryption**.
5. Choose **Manage**.
6. For **Default encryption key**, choose a symmetric customer managed CMK.
7. Choose **Update EBS encryption**.

Old console

To configure the default key for EBS encryption for a Region

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region.
3. From the navigation pane, select **EC2 Dashboard**.
4. In the upper-right corner of the page, choose **Account Attributes, Settings**.
5. Choose **Change the default key** and then choose an available key.
6. Choose **Save settings**.

Encryption by default

You can configure your AWS account to enforce the encryption of the new EBS volumes and snapshot copies that you create. For example, Amazon EBS encrypts the EBS volumes created when you launch an instance and the snapshots that you copy from an unencrypted snapshot. For examples of transitioning from unencrypted to encrypted EBS resources, see [Encrypting unencrypted resources \(p. 1125\)](#).

Encryption by default has no effect on existing EBS volumes or snapshots.

Considerations

- Encryption by default is a Region-specific setting. If you enable it for a Region, you cannot disable it for individual volumes or snapshots in that Region.
- When you enable encryption by default, you can launch an instance only if the instance type supports EBS encryption. For more information, see [Supported instance types \(p. 1122\)](#).
- When migrating servers using AWS Server Migration Service (SMS), do not turn on encryption by default. If encryption by default is already on and you are experiencing delta replication failures, turn off encryption by default. Instead, enable AMI encryption when you create the replication job.

New console

To enable encryption by default for a Region

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region.
3. From the navigation pane, select **EC2 Dashboard**.
4. In the upper-right corner of the page, choose **Account Attributes, EBS encryption**.
5. Choose **Manage**.
6. Select **Enable**. You keep the AWS managed CMK with the alias `alias/aws/ebs` created on your behalf as the default encryption key, or choose a symmetric customer managed CMK.
7. Choose **Update EBS encryption**.

Old console

To enable encryption by default for a Region

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region.
3. From the navigation pane, select **EC2 Dashboard**.
4. In the upper-right corner of the page, choose **Account Attributes, Settings**.
5. Under **EBS Storage**, select **Always encrypt new EBS volumes**.
6. Choose **Save settings**.

You cannot change the CMK that is associated with an existing snapshot or encrypted volume. However, you can associate a different CMK during a snapshot copy operation so that the resulting copied snapshot is encrypted by the new CMK.

Encrypting EBS resources

You encrypt EBS volumes by enabling encryption, either using [encryption by default \(p. 1124\)](#) or by enabling encryption when you create a volume that you want to encrypt.

When you encrypt a volume, you can specify the symmetric CMK to use to encrypt the volume. If you do not specify a CMK, the key that is used for encryption depends on the encryption state of the source snapshot and its ownership. For more information, see the [encryption outcomes table \(p. 1130\)](#).

Note

If you are using the API or AWS CLI to specify a CMK, be aware that AWS authenticates the CMK asynchronously. If you specify a key ID, an alias, or an ARN that is not valid, the action can appear to complete, but it eventually fails.

You cannot change the CMK that is associated with an existing snapshot or volume. However, you can associate a different CMK during a snapshot copy operation so that the resulting copied snapshot is encrypted by the new CMK.

[Encrypting an empty volume on creation](#)

When you create a new, empty EBS volume, you can encrypt it by enabling encryption for the specific volume creation operation. If you enabled EBS encryption by default, the volume is automatically encrypted. By default, the volume is encrypted to your default key for EBS encryption. Alternatively, you can specify a different symmetric CMK for the specific volume creation operation. The volume is encrypted by the time it is first available, so your data is always secured. For detailed procedures, see [Creating an Amazon EBS volume \(p. 1040\)](#).

By default, the CMK that you selected when creating a volume encrypts the snapshots that you make from the volume and the volumes that you restore from those encrypted snapshots. You cannot remove encryption from an encrypted volume or snapshot, which means that a volume restored from an encrypted snapshot, or a copy of an encrypted snapshot, is always encrypted.

Public snapshots of encrypted volumes are not supported, but you can share an encrypted snapshot with specific accounts. For detailed directions, see [Sharing an Amazon EBS snapshot \(p. 1073\)](#).

[Encrypting unencrypted resources](#)

Although there is no direct way to encrypt an existing unencrypted volume or snapshot, you can encrypt them by creating either a volume or a snapshot. If you enabled encryption by default, Amazon EBS encrypts the resulting new volume or snapshot using your default key for EBS encryption. Even if you have not enabled encryption by default, you can enable encryption when you create an individual volume or snapshot. Whether you enable encryption by default or in individual creation operations, you can override the default key for EBS encryption and select a symmetric customer managed CMK. For more information, see [Creating an Amazon EBS volume \(p. 1040\)](#) and [Copying an Amazon EBS snapshot \(p. 1068\)](#).

To encrypt the snapshot copy to a customer managed CMK, you must both enable encryption and specify the key, as shown in [Copy an unencrypted snapshot \(encryption by default not enabled\) \(p. 1127\)](#).

Important

Amazon EBS does not support asymmetric CMKs. For more information, see [Using Symmetric and Asymmetric Keys](#) in the *AWS Key Management Service Developer Guide*.

You can also apply new encryption states when launching an instance from an EBS-backed AMI. This is because EBS-backed AMIs include snapshots of EBS volumes that can be encrypted as described. For more information, see [Using encryption with EBS-backed AMIs \(p. 157\)](#).

[Encryption scenarios](#)

When you create an encrypted EBS resource, it is encrypted by your account's default key for EBS encryption unless you specify a different customer managed CMK in the volume creation parameters or the block device mapping for the AMI or instance. For more information, see [Default key for EBS encryption \(p. 1123\)](#).

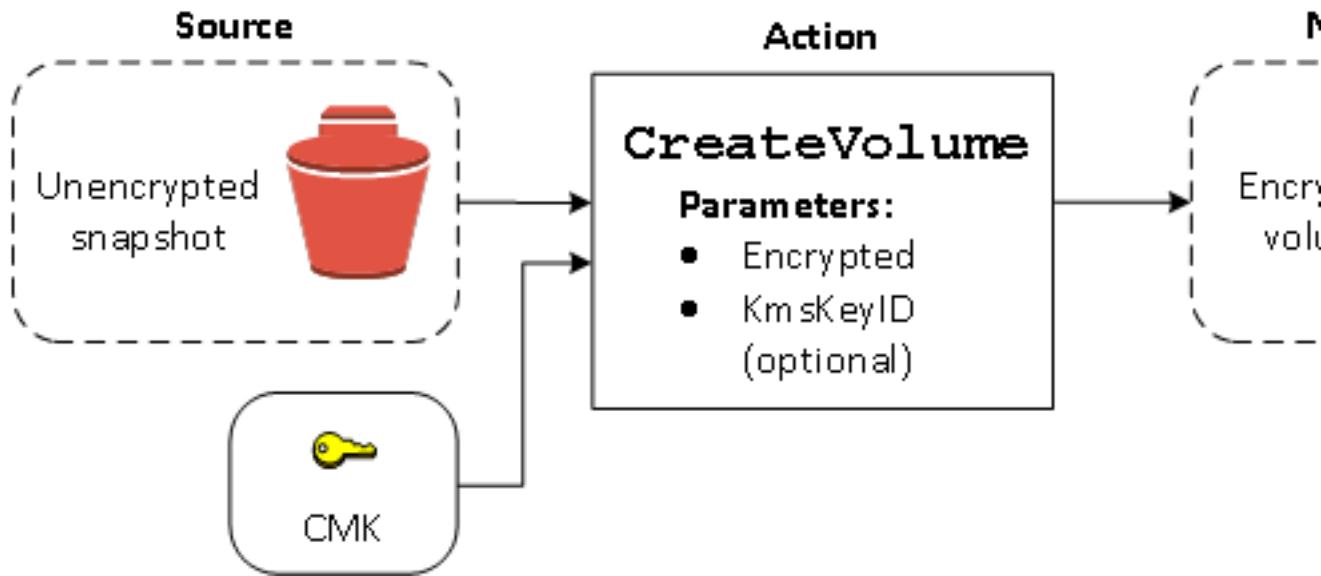
The following examples illustrate how you can manage the encryption state of your volumes and snapshots. For a full list of encryption cases, see the [encryption outcomes table \(p. 1130\)](#).

Examples

- [Restore an unencrypted volume \(encryption by default not enabled\) \(p. 1126\)](#)
- [Restore an unencrypted volume \(encryption by default enabled\) \(p. 1126\)](#)
- [Copy an unencrypted snapshot \(encryption by default not enabled\) \(p. 1127\)](#)
- [Copy an unencrypted snapshot \(encryption by default enabled\) \(p. 1128\)](#)
- [Re-encrypt an encrypted volume \(p. 1128\)](#)
- [Re-encrypt an encrypted snapshot \(p. 1129\)](#)
- [Migrate data between encrypted and unencrypted volumes \(p. 1130\)](#)
- [Encryption outcomes \(p. 1130\)](#)

[Restore an unencrypted volume \(encryption by default not enabled\)](#)

Without encryption by default enabled, a volume restored from an unencrypted snapshot is unencrypted by default. However, you can encrypt the resulting volume by setting the `Encrypted` parameter and, optionally, the `KmsKeyId` parameter. The following diagram illustrates the process.

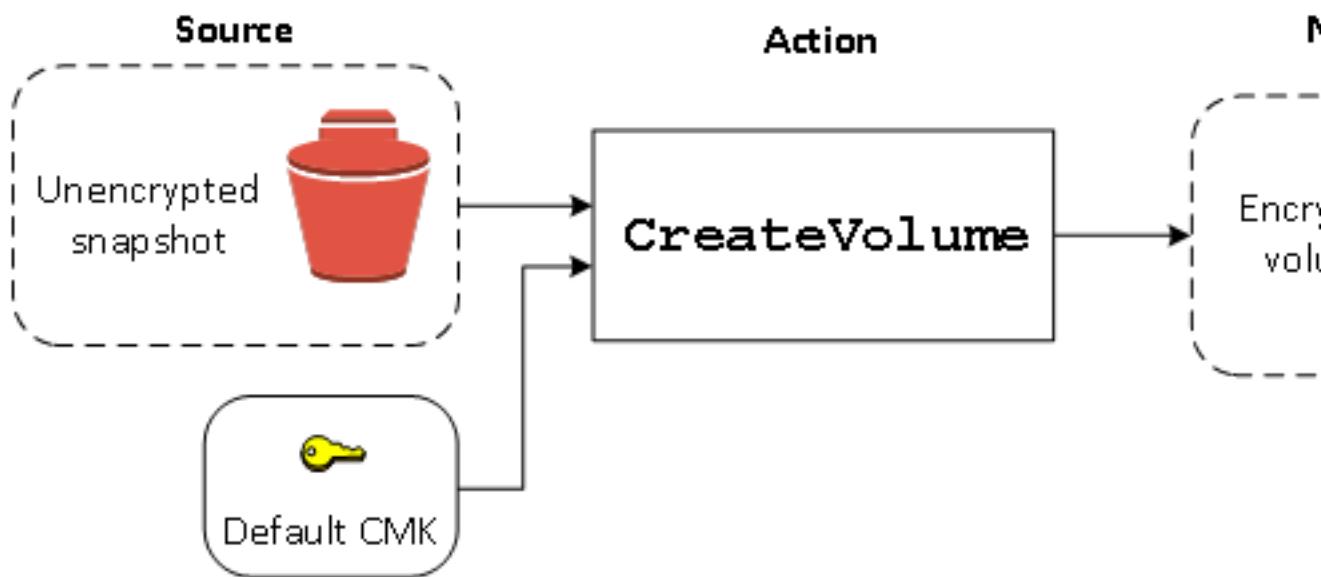


If you leave out the `KmsKeyId` parameter, the resulting volume is encrypted using your default key for EBS encryption. You must specify a key ID to encrypt the volume to a different CMK.

For more information, see [Creating a volume from a snapshot \(p. 1041\)](#).

[Restore an unencrypted volume \(encryption by default enabled\)](#)

When you have enabled encryption by default, encryption is mandatory for volumes restored from unencrypted snapshots, and no encryption parameters are required for your default CMK to be used. The following diagram shows this simple default case:

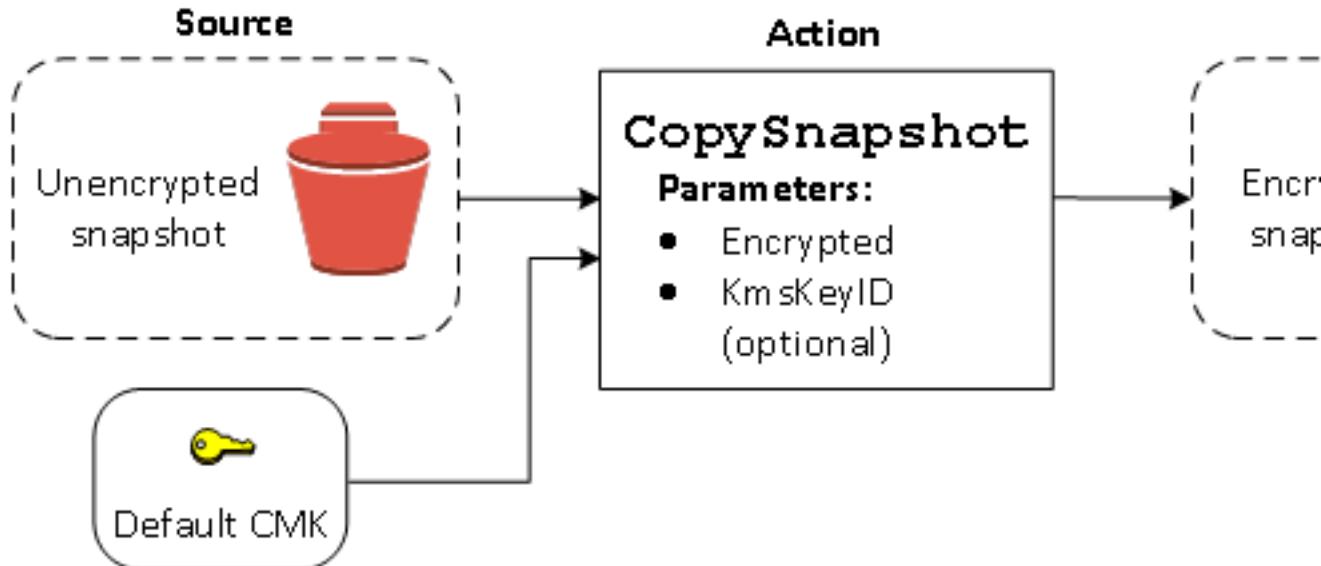


If you want to encrypt the restored volume to a symmetric customer managed CMK, you must supply both the `Encrypted` and `KmsKeyId` parameters as shown in [Restore an unencrypted volume \(encryption by default not enabled\) \(p. 1126\)](#).

[Copy an unencrypted snapshot \(encryption by default not enabled\)](#)

Without encryption by default enabled, a copy of an unencrypted snapshot is unencrypted by default. However, you can encrypt the resulting snapshot by setting the `Encrypted` parameter and, optionally, the `KmsKeyId` parameter. If you omit `KmsKeyId`, the resulting snapshot is encrypted by your default CMK. You must specify a key ID to encrypt the volume to a different symmetric CMK.

The following diagram illustrates the process.



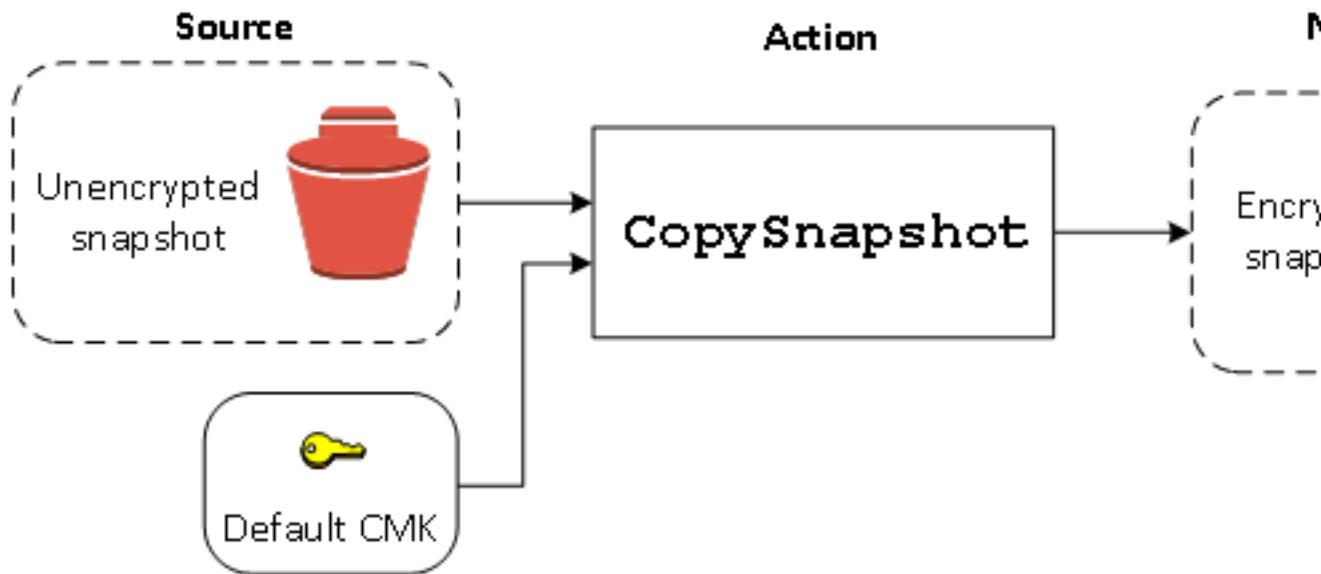
Note

If you copy a snapshot and encrypt it to a new CMK, a complete (non-incremental) copy is always created, resulting in additional delay and storage costs.

You can encrypt an EBS volume by copying an unencrypted snapshot to an encrypted snapshot and then creating a volume from the encrypted snapshot. For more information, see [Copying an Amazon EBS snapshot \(p. 1068\)](#).

[Copy an unencrypted snapshot \(encryption by default enabled\)](#)

When you have enabled encryption by default, encryption is mandatory for copies of unencrypted snapshots, and no encryption parameters are required if your default CMK is used. The following diagram illustrates this default case:

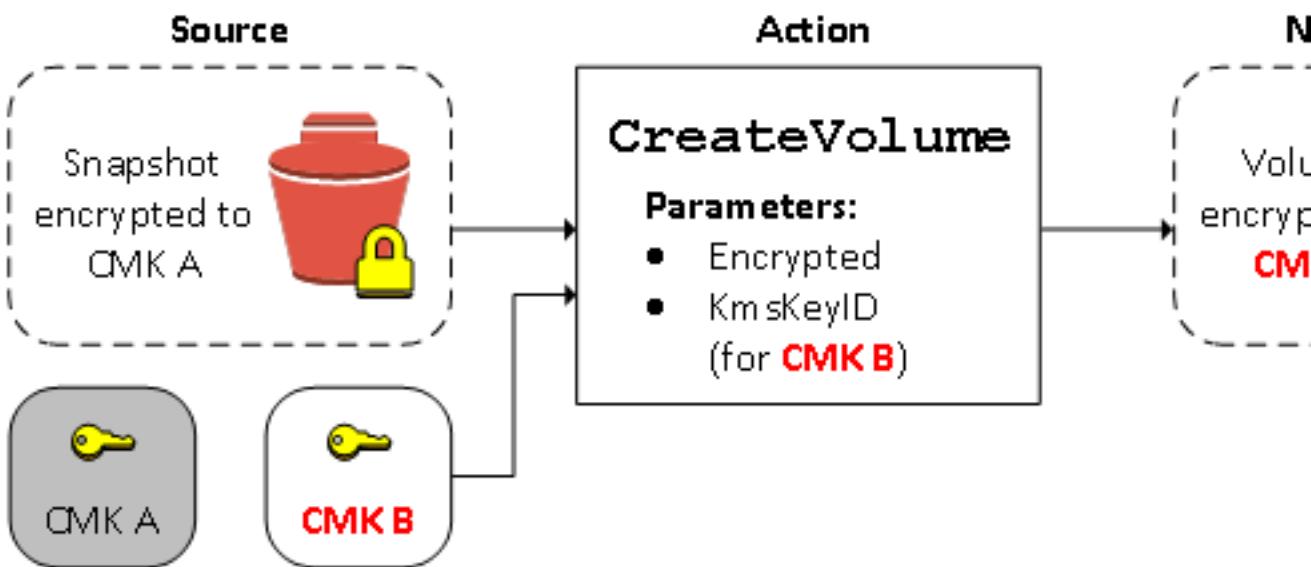


Note

If you copy a snapshot and encrypt it to a new CMK, a complete (non-incremental) copy is always created, resulting in additional delay and storage costs.

[Re-encrypt an encrypted volume](#)

When the `createVolume` action operates on an encrypted snapshot, you have the option of re-encrypting it with a different CMK. The following diagram illustrates the process. In this example, you own two CMKs, CMK A and CMK B. The source snapshot is encrypted by CMK A. During volume creation, with the key ID of CMK B specified as a parameter, the source data is automatically decrypted, then re-encrypted by CMK B.



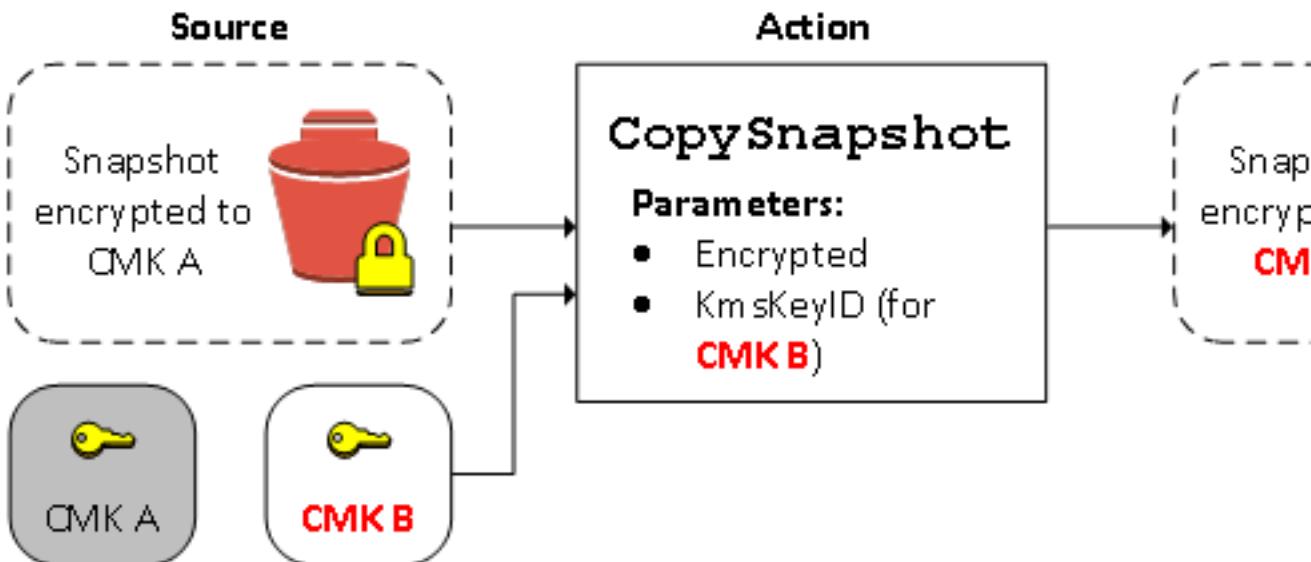
Note

If you copy a snapshot and encrypt it to a new CMK, a complete (non-incremental) copy is always created, resulting in additional delay and storage costs.

For more information, see [Creating a volume from a snapshot \(p. 1041\)](#).

Re-encrypt an encrypted snapshot

The ability to encrypt a snapshot during copying allows you to apply a new symmetric CMK to an already-encrypted snapshot that you own. Volumes restored from the resulting copy are only accessible using the new CMK. The following diagram illustrates the process. In this example, you own two CMKs, CMK A and CMK B. The source snapshot is encrypted by CMK A. During copy, with the key ID of CMK B specified as a parameter, the source data is automatically re-encrypted by CMK B.



Note

If you copy a snapshot and encrypt it to a new CMK, a complete (non-incremental) copy is always created, resulting in additional delay and storage costs.

In a related scenario, you can choose to apply new encryption parameters to a copy of a snapshot that has been shared with you. By default, the copy is encrypted with a CMK shared by the snapshot's owner. However, we recommend that you create a copy of the shared snapshot using a different CMK that you control. This protects your access to the volume if the original CMK is compromised, or if the owner revokes the CMK for any reason. For more information, see [Encryption and snapshot copying \(p. 1070\)](#).

Migrate data between encrypted and unencrypted volumes

When you have access to both an encrypted and unencrypted volume, you can freely transfer data between them. EC2 carries out the encryption and decryption operations transparently.

For example, use the **rsync** command to copy the data. In the following command, the source data is located in `/mnt/source` and the destination volume is mounted at `/mnt/destination`.

```
[ec2-user ~]$ sudo rsync -avh --progress /mnt/source/ /mnt/destination/
```

Encryption outcomes

The following table describes the encryption outcome for each possible combination of settings.

Is encryption enabled?	Is encryption by default enabled?	Source of volume	Default (no CMK specified)	Custom (CMK specified)
No	No	New (empty) volume	Unencrypted	N/A
No	No	Unencrypted snapshot that you own	Unencrypted	
No	No	Encrypted snapshot that you own	Encrypted by same key	
No	No	Unencrypted snapshot that is shared with you	Unencrypted	
No	No	Encrypted snapshot that is shared with you	Encrypted by default CMK*	
Yes	No	New volume	Encrypted by default CMK	Encrypted by a specified CMK**
Yes	No	Unencrypted snapshot that you own	Encrypted by default CMK	
Yes	No	Encrypted snapshot that you own	Encrypted by same key	
Yes	No	Unencrypted snapshot that is shared with you	Encrypted by default CMK	
Yes	No	Encrypted snapshot that is shared with you	Encrypted by default CMK	
No	Yes	New (empty) volume	Encrypted by default CMK	N/A
No	Yes	Unencrypted snapshot that you own	Encrypted by default CMK	

Is encryption enabled?	Is encryption by default enabled?	Source of volume	Default (no CMK specified)	Custom (CMK specified)
No	Yes	Encrypted snapshot that you own	Encrypted by same key	
No	Yes	Unencrypted snapshot that is shared with you	Encrypted by default CMK	
No	Yes	Encrypted snapshot that is shared with you	Encrypted by default CMK	
Yes	Yes	New volume	Encrypted by default CMK	Encrypted by a specified CMK
Yes	Yes	Unencrypted snapshot that you own	Encrypted by default CMK	
Yes	Yes	Encrypted snapshot that you own	Encrypted by same key	
Yes	Yes	Unencrypted snapshot that is shared with you	Encrypted by default CMK	
Yes	Yes	Encrypted snapshot that is shared with you	Encrypted by default CMK	

* This is the default CMK used for EBS encryption for the AWS account and Region. By default this is a unique AWS managed CMK for EBS, or you can specify a customer managed CMK. For more information, see [Default key for EBS encryption \(p. 1123\)](#).

** This is a customer managed CMK specified for the volume at launch time. This CMK is used instead of the default CMK for the AWS account and Region.

Setting encryption defaults using the API and CLI

You can manage encryption by default and the default customer master key (CMK) using the following API actions and CLI commands.

API action	CLI command	Description
DisableEbsEncryptionByDefault	<code>disable-ebs-encryption-by-default</code>	Disables encryption by default.
EnableEbsEncryptionByDefault	<code>enable-ebs-encryption-by-default</code>	Enables encryption by default.
GetEbsDefaultKmsKeyId	<code>get-ebs-default-kms-key-id</code>	Describes the default CMK.
GetEbsEncryptionByDefault	<code>get-ebs-encryption-by-default</code>	Indicates whether encryption by default is enabled.
ModifyEbsDefaultKmsKeyId	<code>modify-ebs-default-kms-key-id</code>	Changes the default CMK used to encrypt EBS volumes.

API action	CLI command	Description
ResetEbsDefaultKmsKeyId	<code>reset-ebs-default-kms-key-id</code>	Resets the AWS managed default CMK as the default CMK used to encrypt EBS volumes.

Amazon EBS fast snapshot restore

Amazon EBS fast snapshot restore enables you to create a volume from a snapshot that is fully initialized at creation. This eliminates the latency of I/O operations on a block when it is accessed for the first time. Volumes that are created using fast snapshot restore instantly deliver all of their provisioned performance.

To get started, enable fast snapshot restore for specific snapshots in specific Availability Zones. Each snapshot and Availability Zone pair refers to one fast snapshot restore. When you create a volume from one of these snapshots in one of its enabled Availability Zones, the volume is restored using fast snapshot restore.

You can enable fast snapshot restore for snapshots that you own and for public and private snapshots that are shared with you.

Contents

- [Fast snapshot restore quotas \(p. 1132\)](#)
- [Fast snapshot restore states \(p. 1132\)](#)
- [Volume creation credits \(p. 1133\)](#)
- [Managing fast snapshot restore \(p. 1133\)](#)
- [View snapshots with fast snapshot restore enabled \(p. 1134\)](#)
- [View volumes restored using fast snapshot restore \(p. 1135\)](#)
- [Monitoring fast snapshot restore \(p. 1135\)](#)
- [Pricing and Billing \(p. 1135\)](#)

Fast snapshot restore quotas

You can enable up to 50 snapshots for fast snapshot restore per Region. The quota applies to snapshots that you own and snapshots that are shared with you. If you enable fast snapshot restore for a snapshot that is shared with you, it counts towards your fast snapshot restore quota. It does not count towards the snapshot owner's fast snapshot restore quota.

Fast snapshot restore states

After you enable fast snapshot restore for a snapshot, it can be in one of the following states.

- **enabling** — A request was made to enable fast snapshot restore.
- **optimizing** — Fast snapshot restore is being enabled. It takes 60 minutes per TiB to optimize a snapshot.
- **enabled** — Fast snapshot restore is enabled.
- **disabling** — A request was made to disable fast snapshot restore, or a request to enable fast snapshot restore failed.
- **disabled** — Fast snapshot restore is disabled. You can enable fast snapshot restore again as needed.

Volume creation credits

The number of volumes that receive the full performance benefit of fast snapshot restore is determined by the volume creation credits for the snapshot. There is one credit bucket per snapshot per Availability Zone. Each volume that you create from a snapshot with fast snapshot restore enabled consumes one credit from the credit bucket.

When you enable fast snapshot restore for a snapshot that is shared with you, you get a separate credit bucket for the shared snapshot in your account. If you create volumes from the shared snapshot, the credits are consumed from your credit bucket; they are not consumed from the snapshot owner's credit bucket.

The size of a credit bucket depends on the size of the snapshot, not the size of the volumes created from the snapshot. The size of the credit bucket for each snapshot is calculated as follows:

```
MAX (1, MIN (10, FLOOR(1024/snapshot_size_gib)))
```

As you consume credits, the credit bucket is refilled over time. The refill rate for each credit bucket is calculated as follows:

```
MIN (10, 1024/snapshot_size_gib)
```

For example, if you enable fast snapshot restore for a snapshot with a size of 100 GiB, the maximum size of its credit bucket is 10 credits and the refill rate is 10 credits per hour. When the credit bucket is full, you can create 10 initialized volumes from this snapshot simultaneously.

You can use Cloudwatch metrics to monitor the size of your credit buckets and the number of credits available in each bucket. For more information, see [Fast snapshot restore metrics \(p. 1176\)](#).

After you create a volume from a snapshot with fast snapshot restore enabled, you can describe the volume using [describe-volumes](#) and check the `fastRestored` field in the output to determine whether the volume was created as an initialized volume using fast snapshot restore.

Managing fast snapshot restore

Fast snapshot restore is disabled for a snapshot by default. You can enable or disable fast snapshot restore for snapshots that you own and for snapshots that are shared with you. When you enable or disable fast snapshot restore for a snapshot, the changes apply to your account only.

Note

When you enable fast snapshot restore for a snapshot, your account is billed for each minute that fast snapshot restore is enabled in a particular Availability Zone. Charges are pro-rated and have a minimum of one hour.

When you delete a snapshot that you own, fast snapshot restore is automatically disabled for that snapshot in your account. If you enabled fast snapshot restore for a snapshot that is shared with you, and the snapshot owner deletes or unshares it, fast snapshot restore is automatically disabled for the shared snapshot in your account.

If you enabled fast snapshot restore for a snapshot that is shared with you, and it's encrypted using a custom CMK, fast snapshot restore is not automatically disabled for the snapshot when the snapshot owner revokes your access to the custom CMK. You must manually disable fast snapshot restore for that snapshot.

Use the following procedure to enable or disable fast snapshot restore for a snapshot that you own or for a snapshot that is shared with you.

To enable or disable fast snapshot restore

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Snapshots**.
3. Select the snapshot.
4. Choose **Actions, Manage Fast Snapshot Restore**.
5. Select or deselect Availability Zones, and then choose **Save**.
6. To track the state of fast snapshot restore as it is enabled, see **Fast Snapshot Restore** on the **Description** tab.

To manage fast snapshot restore using the AWS CLI

- [enable-fast-snapshot-restores](#)
- [disable-fast-snapshot-restores](#)
- [describe-fast-snapshot-restores](#)

View snapshots with fast snapshot restore enabled

Use the following procedure to view the state of fast snapshot restore for a snapshot that you own or for a snapshot that is shared with you.

To view the state of fast snapshot restore using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Snapshots**.
3. Select the snapshot.
4. On the **Description** tab, see **Fast Snapshot Restore**, which indicates the state of fast snapshot restore. For example, it might show a state of "2 Availability Zones optimizing" or "2 Availability Zones enabled".

To view snapshots with fast snapshot restore enabled using the AWS CLI

Use the [describe-fast-snapshot-restores](#) command to describe the snapshots that are enabled for fast snapshot restore.

```
aws ec2 describe-fast-snapshot-restores --filters Name=state,Values=enabled
```

The following is example output.

```
{
  "FastSnapshotRestores": [
    {
      "SnapshotId": "snap-0e946653493cb0447",
      "AvailabilityZone": "us-east-2a",
      "State": "enabled",
      "StateTransitionReason": "Client.UserInitiated - Lifecycle state transition",
      "OwnerId": "123456789012",
      "EnablingTime": "2020-01-25T23:57:49.596Z",
      "OptimizingTime": "2020-01-25T23:58:25.573Z",
      "EnabledTime": "2020-01-25T23:59:29.852Z"
    },
    {
      "SnapshotId": "snap-0e946653493cb0447",
      "AvailabilityZone": "us-east-2b",
      "State": "enabled",
    }
  ]
}
```

```
        "StateTransitionReason": "Client.UserInitiated - Lifecycle state transition",
        "OwnerId": "123456789012",
        "EnablingTime": "2020-01-25T23:57:49.596Z",
        "OptimizingTime": "2020-01-25T23:58:25.573Z",
        "EnabledTime": "2020-01-25T23:59:29.852Z"
    }
}
```

View volumes restored using fast snapshot restore

When you create a volume from a snapshot that is enabled for fast snapshot restore in the Availability Zone for the volume, it is restored using fast snapshot restore.

Use the [describe-volumes](#) command to view volumes that were created from a snapshot that is enabled for fast snapshot restore.

```
aws ec2 describe-volumes --filters Name=fast-restored,Values=true
```

The following is example output.

```
{
    "Volumes": [
        {
            "Attachments": [],
            "AvailabilityZone": "us-east-2a",
            "CreateTime": "2020-01-26T00:34:11.093Z",
            "Encrypted": true,
            "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/8c5b2c63-b9bc-45a3-
a87a-5513e232e843",
            "Size": 20,
            "SnapshotId": "snap-0e946653493cb0447",
            "State": "available",
            "VolumeId": "vol-0d371921d4ca797b0",
            "Iops": 100,
            "VolumeType": "gp2",
            "FastRestored": true
        }
    ]
}
```

Monitoring fast snapshot restore

Amazon EBS emits Amazon CloudWatch events when the fast snapshot restore state for a snapshot changes. For more information, see [EBS fast snapshot restore events \(p. 1185\)](#).

Pricing and Billing

You are billed for each minute that fast snapshot restore is enabled for a snapshot in a particular Availability Zone. Charges are pro-rated with a minimum of one hour.

For example, if you enable fast snapshot restore for one snapshot in us-East-1a for one month (30 days), you are billed **\$540** (1 snapshot x 1 AZ x 720 hours x \$0.75 per hour). If you enable fast snapshot restore for two snapshots in us-east-1a, us-east-1b, and us-east-1c for the same period, you are billed **\$3240** (2 snapshot x 3 AZs x 720 hours x \$0.75 per hour).

If you enable fast snapshot restore for a public or private snapshot that is shared with you, your account is billed; the snapshot owner is not billed. When a snapshot that is shared with you is deleted or unshared by the snapshot owner, fast snapshot restore is disabled for the snapshot in your account and billing is stopped.

For more information, see [Amazon EBS pricing](#).

Amazon EBS and NVMe on Linux instances

EBS volumes are exposed as NVMe block devices on instances built on the [Nitro System \(p. 205\)](#). The device names are `/dev/nvme0n1`, `/dev/nvme1n1`, and so on. The device names that you specify in a block device mapping are renamed using NVMe device names (`/dev/nvme[0-26]n1`). The block device driver can assign NVMe device names in a different order than you specified for the volumes in the block device mapping.

The EBS performance guarantees stated in [Amazon EBS Product Details](#) are valid regardless of the block-device interface.

Contents

- [Install or upgrade the NVMe driver \(p. 1136\)](#)
- [Identifying the EBS device \(p. 1137\)](#)
- [Working with NVMe EBS volumes \(p. 1138\)](#)
- [I/O operation timeout \(p. 1139\)](#)

Install or upgrade the NVMe driver

To access NVMe volumes, the NVMe drivers must be installed. Instances can support NVMe EBS volumes, NVMe instance store volumes, both types of NVMe volumes, or no NVMe volumes. For more information, see [Summary of networking and storage features \(p. 206\)](#).

The following AMIs include the required NVMe drivers:

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (with `linux-aws` kernel) or later
- Red Hat Enterprise Linux 7.4 or later
- SUSE Linux Enterprise Server 12 SP2 or later
- CentOS 7.4.1708 or later
- FreeBSD 11.1 or later
- Debian GNU/Linux 9 or later

For more information about NVMe drivers on Windows instances, see [Amazon EBS and NVMe on Windows Instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

To confirm that your instance has the NVMe driver

You can confirm that your instance has the NVMe driver and check the driver version using the following command. If the instance has the NVMe driver, the command returns information about the driver.

```
$ modinfo nvme
```

To update the NVMe driver

If your instance has the NVMe driver, you can update the driver to the latest version using the following procedure.

1. Connect to your instance.

2. Update your package cache to get necessary package updates as follows.
 - For Amazon Linux 2, Amazon Linux, CentOS, and Red Hat Enterprise Linux:

```
[ec2-user ~]$ sudo yum update -y
```

- For Ubuntu and Debian:

```
[ec2-user ~]$ sudo apt-get update -y
```

3. Ubuntu 16.04 and later include the `linux-aws` package, which contains the NVMe and ENA drivers required by Nitro-based instances. Upgrade the `linux-aws` package to receive the latest version as follows:

```
[ec2-user ~]$ sudo apt-get install --only-upgrade -y linux-aws
```

For Ubuntu 14.04, you can install the latest `linux-aws` package as follows:

```
[ec2-user ~]$ sudo apt-get install linux-aws
```

4. Reboot your instance to load the latest kernel version.

```
sudo reboot
```

5. Reconnect to your instance after it has rebooted.

Identifying the EBS device

EBS uses single-root I/O virtualization (SR-IOV) to provide volume attachments on Nitro-based instances using the NVMe specification. These devices rely on standard NVMe drivers on the operating system. These drivers typically discover attached devices by scanning the PCI bus during instance boot, and create device nodes based on the order in which the devices respond, not on how the devices are specified in the block device mapping. In Linux, NVMe device names follow the pattern `/dev/nvme<x>n<y>`, where `<x>` is the enumeration order, and, for EBS, `<y>` is 1. Occasionally, devices can respond to discovery in a different order in subsequent instance starts, which causes the device name to change.

We recommend that you use stable identifiers for your EBS volumes within your instance, such as one of the following:

- For Nitro-based instances, the block device mappings that are specified in the Amazon EC2 console when you are attaching an EBS volume or during `AttachVolume` or `RunInstances` API calls are captured in the vendor-specific data field of the NVMe controller identification. With Amazon Linux AMIs later than version 2017.09.01, we provide a udev rule that reads this data and creates a symbolic link to the block-device mapping.
- NVMe EBS volumes have the EBS volume ID set as the serial number in the device identification. Use the `lsblk -o +SERIAL` command to list the serial number.
- When a device is formatted, a UUID is generated that persists for the life of the filesystem. A device label can be specified at the same time. For more information, see [Making an Amazon EBS volume available for use on Linux \(p. 1046\)](#) and [Booting from the wrong volume \(p. 1280\)](#).

Amazon Linux AMIs

With Amazon Linux AMI 2017.09.01 or later (including Amazon Linux 2), you can run the `ebsnvme-id` command as follows to map the NVMe device name to a volume ID and device name:

```
[ec2-user ~]$ sudo /sbin/ebsnvme-id /dev/nvme1n1
Volume ID: vol-01324f611e2463981
/dev/sdf
```

Amazon Linux also creates a symbolic link from the device name in the block device mapping (for example, `/dev/sdf`), to the NVMe device name.

FreeBSD AMIs

Starting with FreeBSD 12.2-RELEASE, you can run the **ebsnvme-id** command as shown above. Pass either the name of the NVMe device (for example, `nvme0`) or the disk device (for example, `nvd0` or `nda0`). FreeBSD also creates symbolic links to the disk devices (for example, `/dev/aws/disk/ebs/volume_id`).

Other Linux AMIs

With a kernel version of 4.2 or later, you can run the **nvme id-ctrl** command as follows to map an NVMe device to a volume ID. First, install the NVMe command line package, `nvme-cli`, using the package management tools for your Linux distribution. For download and installation instructions for other distributions, refer to the documentation specific to your distribution.

The following example gets the volume ID and device name. The device name is available through the NVMe controller vendor-specific extension (bytes 384:4095 of the controller identification):

```
[ec2-user ~]$ sudo nvme id-ctrl -v /dev/nvme1n1
NVME Identify Controller:
vid      : 0x1d0f
ssvid    : 0x1d0f
sn      : vol01234567890abcdef
mn      : Amazon Elastic Block Store
...
0000: 2f 64 65 76 2f 73 64 6a 20 20 20 20 20 20 20 20 "/dev/sdf..."
```

The **lsblk** command lists available devices and their mount points (if applicable). This helps you determine the correct device name to use. In this example, `/dev/nvme0n1p1` is mounted as the root device and `/dev/nvme1n1` is attached but not mounted.

```
[ec2-user ~]$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme1n1   259:3    0  100G  0 disk
nvme0n1   259:0    0     8G  0 disk
  nvme0n1p1 259:1    0     8G  0 part /
  nvme0n1p128 259:2   0     1M  0 part
```

Working with NVMe EBS volumes

To format and mount an NVMe EBS volume, see [Making an Amazon EBS volume available for use on Linux \(p. 1046\)](#).

If you are using Linux kernel 4.2 or later, any change you make to the volume size of an NVMe EBS volume is automatically reflected in the instance. For older Linux kernels, you might need to detach and attach the EBS volume or reboot the instance for the size change to be reflected. With Linux kernel 3.19 or later, you can use the **hdparm** command as follows to force a rescan of the NVMe device:

```
[ec2-user ~]$ sudo hdparm -z /dev/nvme1n1
```

When you detach an NVMe EBS volume, the instance does not have an opportunity to flush the file system caches or metadata before detaching the volume. Therefore, before you detach an NVMe EBS volume, you should first sync and unmount it. If the volume fails to detach, you can attempt a force-detach command as described in [Detaching an Amazon EBS volume from a Linux instance \(p. 1058\)](#).

I/O operation timeout

EBS volumes attached to Nitro-based instances use the default NVMe driver provided by the operating system. Most operating systems specify a timeout for I/O operations submitted to NVMe devices. The default timeout is 30 seconds and can be changed using the `nvme_core.io_timeout` boot parameter. For most Linux kernels earlier than version 4.6, this parameter is `nvme.io_timeout`.

If I/O latency exceeds the value of this timeout parameter, the Linux NVMe driver fails the I/O and returns an error to the filesystem or application. Depending on the I/O operation, your filesystem or application can retry the error. In some cases, your filesystem might be remounted as read-only.

For an experience similar to EBS volumes attached to Xen instances, we recommend setting `nvme_core.io_timeout` to the highest value possible. For current kernels, the maximum is 4294967295, while for earlier kernels the maximum is 255. Depending on the version of Linux, the timeout might already be set to the supported maximum value. For example, the timeout is set to 4294967295 by default for Amazon Linux AMI 2017.09.01 and later.

You can verify the maximum value for your Linux distribution by writing a value higher than the suggested maximum to `/sys/module/nvme_core/parameters/io_timeout` and checking for the Numerical result out of range error when attempting to save the file.

Amazon EBS-optimized instances

An Amazon EBS-optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

EBS-optimized instances deliver dedicated bandwidth to Amazon EBS. When attached to an EBS-optimized instance, General Purpose SSD (`gp2`) volumes are designed to deliver their baseline and burst performance 99% of the time, and Provisioned IOPS SSD (`io1` and `io2`) volumes are designed to deliver their provisioned performance 99.9% of the time. Both Throughput Optimized HDD (`st1`) and Cold HDD (`sc1`) guarantee performance consistency of 90% of burst throughput 99% of the time. Non-compliant periods are approximately uniformly distributed, targeting 99% of expected total throughput each hour. For more information, see [Amazon EBS volume types \(p. 1023\)](#).

Contents

- [Supported instance types \(p. 1139\)](#)
- [Getting maximum performance \(p. 1155\)](#)
- [Enabling EBS optimization at launch \(p. 1155\)](#)
- [Enable EBS optimization for an existing instance \(p. 1156\)](#)

Supported instance types

The following tables show which instance types support EBS optimization. They include the dedicated bandwidth to Amazon EBS, the typical maximum aggregate throughput that can be achieved on that connection with a streaming read workload and 128 KiB I/O size, and the maximum IOPS the instance can support if you are using a 16 KiB I/O size. Choose an EBS-optimized instance that provides more dedicated Amazon EBS throughput than your application needs; otherwise, the connection between Amazon EBS and Amazon EC2 can become a performance bottleneck.

EBS optimized by default

The following table lists the instance types that support EBS optimization and EBS optimization is enabled by default. There is no need to enable EBS optimization and no effect if you disable EBS optimization.

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
a1.medium *	3,500	437.5	20,000
a1.large *	3,500	437.5	20,000
a1.xlarge *	3,500	437.5	20,000
a1.2xlarge *	3,500	437.5	20,000
a1.4xlarge	3,500	437.5	20,000
a1.metal	3,500	437.5	20,000
c4.large	500	62.5	4,000
c4.xlarge	750	93.75	6,000
c4.2xlarge	1,000	125	8,000
c4.4xlarge	2,000	250	16,000
c4.8xlarge	4,000	500	32,000
c5.large *	4,750	593.75	20,000
c5.xlarge *	4,750	593.75	20,000
c5.2xlarge *	4,750	593.75	20,000
c5.4xlarge	4,750	593.75	20,000
c5.9xlarge	9,500	1,187.5	40,000
c5.12xlarge	9,500	1,187.5	40,000
c5.18xlarge	19,000	2,375	80,000
c5.24xlarge	19,000	2,375	80,000
c5.metal	19,000	2,375	80,000
c5a.large *	3,170	396	13,300
c5a.xlarge *	3,170	396	13,300
c5a.2xlarge *	3,170	396	13,300
c5a.4xlarge *	3,170	396	13,300
c5a.8xlarge	3,170	396	13,300
c5a.12xlarge	4,750	594	20,000
c5a.16xlarge	6,300	788	26,700

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
c5a.24xlarge	9,500	1,188	40,000
c5ad.large *	3,170	396	13,300
c5ad.xlarge *	3,170	396	13,300
c5ad.2xlarge *	3,170	396	13,300
c5ad.4xlarge *	3,170	396	13,300
c5ad.8xlarge	3,170	396	13,300
c5ad.12xlarge	4,750	594	20,000
c5ad.16xlarge	6,300	788	26,700
c5ad.24xlarge	9,500	1,188	40,000
c5d.large *	4,750	593.75	20,000
c5d.xlarge *	4,750	593.75	20,000
c5d.2xlarge *	4,750	593.75	20,000
c5d.4xlarge	4,750	593.75	20,000
c5d.9xlarge	9,500	1,187.5	40,000
c5d.12xlarge	9,500	1,187.5	40,000
c5d.18xlarge	19,000	2,375	80,000
c5d.24xlarge	19,000	2,375	80,000
c5d.metal	19,000	2,375	80,000
c5n.large *	4,750	593.75	20,000
c5n.xlarge *	4,750	593.75	20,000
c5n.2xlarge *	4,750	593.75	20,000
c5n.4xlarge	4,750	593.75	20,000
c5n.9xlarge	9,500	1,187.5	40,000
c5n.18xlarge	19,000	2,375	80,000
c5n.metal	19,000	2,375	80,000
c6g.medium *	4,750	593.75	20,000
c6g.large *	4,750	593.75	20,000
c6g.xlarge *	4,750	593.75	20,000
c6g.2xlarge *	4,750	593.75	20,000
c6g.4xlarge	4,750	593.75	20,000

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
c6g.8xlarge	9,500	1,187.5	40,000
c6g.12xlarge	14,250	1,781.25	50,000
c6g.16xlarge	19,000	2,375	80,000
c6g.metal	19,000	2,375	80,000
c6gd.medium *	4,750	593.75	20,000
c6gd.large *	4,750	593.75	20,000
c6gd.xlarge *	4,750	593.75	20,000
c6gd.2xlarge *	4,750	593.75	20,000
c6gd.4xlarge	4,750	593.75	20,000
c6gd.8xlarge	9,500	1,187.5	40,000
c6gd.12xlarge	14,250	1,781.25	50,000
c6gd.16xlarge	19,000	2,375	80,000
c6gd.metal	19,000	2,375	80,000
d2.xlarge	750	93.75	6,000
d2.2xlarge	1,000	125	8,000
d2.4xlarge	2,000	250	16,000
d2.8xlarge	4,000	500	32,000
f1.2xlarge	1,700	212.5	12,000
f1.4xlarge	3,500	437.5	44,000
f1.16xlarge	14,000	1,750	75,000
g3s.xlarge	850	106.25	5,000
g3.4xlarge	3,500	437.5	20,000
g3.8xlarge	7,000	875	40,000
g3.16xlarge	14,000	1,750	80,000
g4dn.xlarge *	3,500	437.5	20,000
g4dn.2xlarge *	3,500	437.5	20,000
g4dn.4xlarge	4,750	593.75	20,000
g4dn.8xlarge	9,500	1,187.5	40,000
g4dn.12xlarge	9,500	1,187.5	40,000
g4dn.16xlarge	9,500	1,187.5	40,000

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
g4dn.metal	19,000	2,375	80,000
h1.2xlarge	1,750	218.75	12,000
h1.4xlarge	3,500	437.5	20,000
h1.8xlarge	7,000	875	40,000
h1.16xlarge	14,000	1,750	80,000
i3.large	425	53.13	3000
i3.xlarge	850	106.25	6000
i3.2xlarge	1,700	212.5	12,000
i3.4xlarge	3,500	437.5	16,000
i3.8xlarge	7,000	875	32,500
i3.16xlarge	14,000	1,750	65,000
i3.metal	19,000	2,375	80,000
i3en.large *	4,750	593.75	20,000
i3en.xlarge *	4,750	593.75	20,000
i3en.2xlarge *	4,750	593.75	20,000
i3en.3xlarge *	4,750	593.75	20,000
i3en.6xlarge	4,750	593.75	20,000
i3en.12xlarge	9,500	1,187.5	40,000
i3en.24xlarge	19,000	2,375	80,000
i3en.metal	19,000	2,375	80,000
inf1.xlarge *	4,750	593.75	20,000
inf1.2xlarge *	4,750	593.75	20,000
inf1.6xlarge	4,750	593.75	20,000
inf1.24xlarge	19,000	2,375	80,000
m4.large	450	56.25	3,600
m4.xlarge	750	93.75	6,000
m4.2xlarge	1,000	125	8,000
m4.4xlarge	2,000	250	16,000
m4.10xlarge	4,000	500	32,000
m4.16xlarge	10,000	1,250	65,000

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
m5.large *	4,750	593.75	18,750
m5.xlarge *	4,750	593.75	18,750
m5.2xlarge *	4,750	593.75	18,750
m5.4xlarge	4,750	593.75	18,750
m5.8xlarge	6,800	850	30,000
m5.12xlarge	9,500	1,187.5	40,000
m5.16xlarge	13,600	1,700	60,000
m5.24xlarge	19,000	2,375	80,000
m5.metal	19,000	2,375	80,000
m5a.large *	2,880	360	16,000
m5a.xlarge *	2,880	360	16,000
m5a.2xlarge *	2,880	360	16,000
m5a.4xlarge	2,880	360	16,000
m5a.8xlarge	4,750	593.75	20,000
m5a.12xlarge	6,780	847.5	30,000
m5a.16xlarge	9,500	1,187.50	40,000
m5a.24xlarge	13,570	1,696.25	60,000
m5ad.large *	2,880	360	16,000
m5ad.xlarge *	2,880	360	16,000
m5ad.2xlarge *	2,880	360	16,000
m5ad.4xlarge	2,880	360	16,000
m5ad.8xlarge	4,750	593.75	20,000
m5ad.12xlarge	6,780	847.5	30,000
m5ad.16xlarge	9,500	1,187.5	40,000
m5ad.24xlarge	13,570	1,696.25	60,000
m5d.large *	4,750	593.75	18,750
m5d.xlarge *	4,750	593.75	18,750
m5d.2xlarge *	4,750	593.75	18,750
m5d.4xlarge	4,750	593.75	18,750
m5d.8xlarge	6,800	850	30,000

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
m5d.12xlarge	9,500	1,187.5	40,000
m5d.16xlarge	13,600	1,700	60,000
m5d.24xlarge	19,000	2,375	80,000
m5d.metal	19,000	2,375	80,000
m5dn.large *	4,750	593.75	18,750
m5dn.xlarge *	4,750	593.75	18,750
m5dn.2xlarge *	4,750	593.75	18,750
m5dn.4xlarge	4,750	593.75	18,750
m5dn.8xlarge	6,800	850	30,000
m5dn.12xlarge	9,500	1,187.5	40,000
m5dn.16xlarge	13,600	1,700	60,000
m5dn.24xlarge	19,000	2,375	80,000
m5n.large *	4,750	593.75	18,750
m5n.xlarge *	4,750	593.75	18,750
m5n.2xlarge *	4,750	593.75	18,750
m5n.4xlarge	4,750	593.75	18,750
m5n.8xlarge	6,800	850	30,000
m5n.12xlarge	9,500	1,187.5	40,000
m5n.16xlarge	13,600	1,700	60,000
m5n.24xlarge	19,000	2,375	80,000
m6g.medium *	4,750	593.75	20,000
m6g.large *	4,750	593.75	20,000
m6g.xlarge *	4,750	593.75	20,000
m6g.2xlarge *	4,750	593.75	20,000
m6g.4xlarge	4,750	593.75	20,000
m6g.8xlarge	9,500	1,187.5	40,000
m6g.12xlarge	14,250	1,781.25	50,000
m6g.16xlarge	19,000	2,375	80,000
m6g.metal	19,000	2,375	80,000
m6gd.medium *	4,750	593.75	20,000

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
m6gd.large *	4,750	593.75	20,000
m6gd.xlarge *	4,750	593.75	20,000
m6gd.2xlarge *	4,750	593.75	20,000
m6gd.4xlarge	4,750	593.75	20,000
m6gd.8xlarge	9,500	1,187.5	40,000
m6gd.12xlarge	14,250	1,781.25	50,000
m6gd.16xlarge	19,000	2,375	80,000
m6gd.metal	19,000	2,375	80,000
p2.xlarge	750	93.75	6,000
p2.8xlarge	5,000	625	32,500
p2.16xlarge	10,000	1,250	65,000
p3.2xlarge	1,750	218.75	10,000
p3.8xlarge	7,000	875	40,000
p3.16xlarge	14,000	1,750	80,000
p3dn.24xlarge	19,000	2,375	80,000
r4.large	425	53.13	3,000
r4.xlarge	850	106.25	6,000
r4.2xlarge	1,700	212.5	12,000
r4.4xlarge	3,500	437.5	18,750
r4.8xlarge	7,000	875	37,500
r4.16xlarge	14,000	1,750	75,000
r5.large *	4,750	593.75	18,750
r5.xlarge *	4,750	593.75	18,750
r5.2xlarge *	4,750	593.75	18,750
r5.4xlarge	4,750	593.75	18,750
r5.8xlarge	6,800	850	30,000
r5.12xlarge	9,500	1,187.5	40,000
r5.16xlarge	13,600	1,700	60,000
r5.24xlarge	19,000	2,375	80,000
r5.metal	19,000	2,375	80,000

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
r5a.large *	2,880	360	16,000
r5a.xlarge *	2,880	360	16,000
r5a.2xlarge *	2,880	360	16,000
r5a.4xlarge	2,880	360	16,000
r5a.8xlarge	4,750	593.75	20,000
r5a.12xlarge	6,780	847.5	30,000
r5a.16xlarge	9,500	1,187.5	40,000
r5a.24xlarge	13,570	1,696.25	60,000
r5ad.large *	2,880	360	16,000
r5ad.xlarge *	2,880	360	16,000
r5ad.2xlarge *	2,880	360	16,000
r5ad.4xlarge	2,880	360	16,000
r5ad.8xlarge	4,750	593.75	20,000
r5ad.12xlarge	6,780	847.5	30,000
r5ad.16xlarge	9,500	1,187.5	40,000
r5ad.24xlarge	13,570	1,696.25	60,000
r5d.large *	4,750	593.75	18,750
r5d.xlarge *	4,750	593.75	18,750
r5d.2xlarge *	4,750	593.75	18,750
r5d.4xlarge	4,750	593.75	18,750
r5d.8xlarge	6,800	850	30,000
r5d.12xlarge	9,500	1,187.5	40,000
r5d.16xlarge	13,600	1,700	60,000
r5d.24xlarge	19,000	2,375	80,000
r5d.metal	19,000	2,375	80,000
r5dn.large *	4,750	593.75	18,750
r5dn.xlarge *	4,750	593.75	18,750
r5dn.2xlarge *	4,750	593.75	18,750
r5dn.4xlarge	4,750	593.75	18,750
r5dn.8xlarge	6,800	850	30,000

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
r5dn.12xlarge	9,500	1,187.5	40,000
r5dn.16xlarge	13,600	1,700	60,000
r5dn.24xlarge	19,000	2,375	80,000
r5n.large *	4,750	593.75	18,750
r5n.xlarge *	4,750	593.75	18,750
r5n.2xlarge *	4,750	593.75	18,750
r5n.4xlarge	4,750	593.75	18,750
r5n.8xlarge	6,800	850	30,000
r5n.12xlarge	9,500	1,187.5	40,000
r5n.16xlarge	13,600	1,700	60,000
r5n.24xlarge	19,000	2,375	80,000
r6g.medium *	4,750	593.75	20,000
r6g.large *	4,750	593.75	20,000
r6g.xlarge *	4,750	593.75	20,000
r6g.2xlarge *	4,750	593.75	20,000
r6g.4xlarge	4,750	593.75	20,000
r6g.8xlarge	9,500	1,187.5	40,000
r6g.12xlarge	14,250	1,781.25	50,000
r6g.16xlarge	19,000	2,375	80,000
r6g.metal	19,000	2,375	80,000
r6gd.medium *	4,750	593.75	20,000
r6gd.large *	4,750	593.75	20,000
r6gd.xlarge *	4,750	593.75	20,000
r6gd.2xlarge *	4,750	593.75	20,000
r6gd.4xlarge	4,750	593.75	20,000
r6gd.8xlarge	9,500	1,187.5	40,000
r6gd.12xlarge	14,250	1,781.25	50,000
r6gd.16xlarge	19,000	2,375	80,000
r6gd.metal	19,000	2,375	80,000
t3.nano *	2,085	260.57	11,800

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
t3.micro *	2,085	260.57	11,800
t3.small *	2,085	260.57	11,800
t3.medium *	2,085	260.57	11,800
t3.large *	2,780	347.5	15,700
t3.xlarge *	2,780	347.5	15,700
t3.2xlarge *	2,780	347.5	15,700
t3a.nano *	2,085	260.57	11,800
t3a.micro *	2,085	260.57	11,800
t3a.small *	2,085	260.57	11,800
t3a.medium *	2,085	260.57	11,800
t3a.large *	2,780	347.5	15,700
t3a.xlarge *	2,780	347.5	15,700
t3a.2xlarge *	2,780	347.5	15,700
t4g.nano *	2,606	325.75	11,800
t4g.micro *	2,606	325.75	11,800
t4g.small *	2,606	325.75	11,800
t4g.medium *	2,606	325.75	11,800
t4g.large *	3,475	434.37	15,700
t4g.xlarge *	3,475	434.37	15,700
t4g.2xlarge *	3,475	434.37	15,700
u-6tb1.metal	38,000	4,750	160,000
u-9tb1.metal	38,000	4,750	160,000
u-12tb1.metal	38,000	4,750	160,000
u-18tb1.metal	38,000	4,750	160,000
u-24tb1.metal	38,000	4,750	160,000
x1.16xlarge	7,000	875	40,000
x1.32xlarge	14,000	1,750	80,000
x1e.xlarge	500	62.5	3,700
x1e.2xlarge	1,000	125	7,400
x1e.4xlarge	1,750	218.75	10,000

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
x1e.8xlarge	3,500	437.5	20,000
x1e.16xlarge	7,000	875	40,000
x1e.32xlarge	14,000	1,750	80,000
z1d.large *	3,170	396.25	13,333
z1d.xlarge *	3,170	396.25	13,333
z1d.2xlarge	3,170	396.25	13,333
z1d.3xlarge	4,750	593.75	20,000
z1d.6xlarge	9,500	1,187.5	40,000
z1d.12xlarge	19,000	2,375	80,000
z1d.metal	19,000	2,375	80,000

* These instance types can support maximum performance for 30 minutes at least once every 24 hours. If you have a workload that requires sustained maximum performance for longer than 30 minutes, select an instance type according to baseline performance as shown in the following table.

Instance size	Baseline bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)
a1.medium	300	37.5	2,500
a1.large	525	65.625	4,000
a1.xlarge	800	100	6,000
a1.2xlarge	1,750	218.75	10,000
c5.large	650	81.25	4,000
c5.xlarge	1,150	143.75	6,000
c5.2xlarge	2,300	287.5	10,000
c5a.large	200	25	800
c5a.xlarge	400	50	1,600
c5a.2xlarge	800	100	3,200
c5a.4xlarge	1,580	198	6,600
c5ad.large	200	25	800
c5ad.xlarge	400	50	1,600
c5ad.2xlarge	800	100	3,200
c5ad.4xlarge	1,580	198	6,600
c5d.large	650	81.25	4,000

Amazon Elastic Compute Cloud
User Guide for Linux Instances
EBS optimization

Instance size	Baseline bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)
c5d.xlarge	1,150	143.75	6,000
c5d.2xlarge	2,300	287.5	10,000
c5n.large	650	81.25	4,000
c5n.xlarge	1,150	143.75	6,000
c5n.2xlarge	2,300	287.5	10,000
c6g.medium	315	39.375	2,500
c6g.large	630	78.75	3,600
c6g.xlarge	1,188	148.5	6,000
c6g.2xlarge	2,375	296.875	12,000
c6gd.medium	315	39.375	2,500
c6gd.large	630	78.75	3,600
c6gd.xlarge	1,188	148.5	6,000
c6gd.2xlarge	2,375	296.875	12,000
g4dn.xlarge	950	118.75	3,000
g4dn.2xlarge	1,150	143.75	6,000
i3en.large	577	72.1	3,000
i3en.xlarge	1,154	144.2	6,000
i3en.2xlarge	2,307	288.39	12,000
i3en.3xlarge	3,800	475	15,000
inf1.xlarge	1,190	148.75	4,000
inf1.2xlarge	1,190	148.75	6,000
m5.large	650	81.25	3,600
m5.xlarge	1,150	143.75	6,000
m5.2xlarge	2,300	287.5	12,000
m5a.large	650	81.25	3,600
m5a.xlarge	1,085	135.63	6,000
m5a.2xlarge	1,580	197.5	8,333
m5ad.large	650	81.25	3,600
m5ad.xlarge	1,085	135.63	6,000
m5ad.2xlarge	1,580	197.5	8,333

Instance size	Baseline bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)
m5d.large	650	81.25	3,600
m5d.xlarge	1,150	143.75	6,000
m5d.2xlarge	2,300	287.5	12,000
m5dn.large	650	81.25	3,600
m5dn.xlarge	1,150	143.75	6,000
m5dn.2xlarge	2,300	287.5	12,000
m5n.large	650	81.25	3,600
m5n.xlarge	1,150	143.75	6,000
m5n.2xlarge	2,300	287.5	12,000
m6g.medium	315	39.375	2,500
m6g.large	630	78.75	3,600
m6g.xlarge	1,188	148.5	6,000
m6g.2xlarge	2,375	296.875	12,000
m6gd.medium	315	39.375	2,500
m6gd.large	630	78.75	3,600
m6gd.xlarge	1,188	148.5	6,000
m6gd.2xlarge	2,375	296.875	12,000
r5.large	650	81.25	3,600
r5.xlarge	1,150	143.75	6,000
r5.2xlarge	2,300	287.5	12,000
r5a.large	650	81.25	3,600
r5a.xlarge	1,085	135.63	6,000
r5a.2xlarge	1,580	197.5	8,333
r5ad.large	650	81.25	3,600
r5ad.xlarge	1,085	135.63	6,000
r5ad.2xlarge	1,580	197.5	8,333
r5d.large	650	81.25	3,600
r5d.xlarge	1,150	143.75	6,000
r5d.2xlarge	2,300	287.5	12,000
r5dn.large	650	81.25	3,600

Instance size	Baseline bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)
r5dn.xlarge	1,150	143.75	6,000
r5dn.2xlarge	2,300	287.5	12,000
r5n.large	650	81.25	3,600
r5n.xlarge	1,150	143.75	6,000
r5n.2xlarge	2,300	287.5	12,000
r6g.medium	315	39.375	2,500
r6g.large	630	78.75	3,600
r6g.xlarge	1,188	148.5	6,000
r6g.2xlarge	2,375	296.875	12,000
r6gd.medium *	315	39.375	2,500
r6gd.large *	630	78.75	3,600
r6gd.xlarge *	1,188	148.5	6,000
r6gd.2xlarge *	2,375	296.875	12,000
t3.nano	43	5.43	250
t3.micro	87	10.86	500
t3.small	174	21.71	1,000
t3.medium	347	43.43	2,000
t3.large	695	86.86	4,000
t3.xlarge	695	86.86	4,000
t3.2xlarge	695	86.86	4,000
t3a.nano	45	5.63	250
t3a.micro	90	11.25	500
t3a.small	175	21.88	1,000
t3a.medium	350	43.75	2,000
t3a.large	695	86.86	4,000
t3a.xlarge	695	86.86	4,000
t3a.2xlarge	695	86.86	4,000
t4g.nano	32	4	250
t4g.micro	64	8	500
t4g.small	128	16	1,000

Instance size	Baseline bandwidth (Mbps)	Baseline throughput (MB/s, 128 KiB I/O)	Baseline IOPS (16 KiB I/O)
t4g.medium	256	32	2,000
t4g.large	512	64	4,000
t4g.xlarge	1,024	128	4,000
t4g.2xlarge	2,048	256	4,000
z1d.large	800	100	3,333
z1d.xlarge	1,580	197.5	6,667

EBS optimization supported

The following table lists the instance types that support EBS optimization but EBS optimization is not enabled by default. You can enable EBS optimization when you launch these instances or after they are running. Instances must have EBS optimization enabled to achieve the level of performance described. When you enable EBS optimization for an instance that is not EBS-optimized by default, you pay an additional low, hourly fee for the dedicated capacity. For pricing information, see [EBS-Optimized Instances](#) on the [Amazon EC2 Pricing, On-Demand Pricing page](#).

Instance size	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KiB I/O)	Maximum IOPS (16 KiB I/O)
c1.xlarge	1,000	125	8,000
c3.xlarge	500	62.5	4,000
c3.2xlarge	1,000	125	8,000
c3.4xlarge	2,000	250	16,000
g2.2xlarge	1,000	125	8,000
i2.xlarge	500	62.5	4,000
i2.2xlarge	1,000	125	8,000
i2.4xlarge	2,000	250	16,000
m1.large	500	62.5	4,000
m1.xlarge	1,000	125	8,000
m2.2xlarge	500	62.5	4,000
m2.4xlarge	1,000	125	8,000
m3.xlarge	500	62.5	4,000
m3.2xlarge	1,000	125	8,000
r3.xlarge	500	62.5	4,000
r3.2xlarge	1,000	125	8,000
r3.4xlarge	2,000	250	16,000

The `i2.8xlarge`, `c3.8xlarge`, and `r3.8xlarge` instances do not have dedicated EBS bandwidth and therefore do not offer EBS optimization. On these instances, network traffic and Amazon EBS traffic share the same 10-gigabit network interface.

Getting maximum performance

You can use the `EBSIOBalance%` and `EBSByteBalance%` metrics to help you determine whether your instances are sized correctly. You can view these metrics in the CloudWatch console and set an alarm that is triggered based on a threshold you specify. These metrics are expressed as a percentage. Instances with a consistently low balance percentage are candidates to size up. Instances where the balance percentage never drops below 100% are candidates for downsizing. For more information, see [Monitoring your instances using CloudWatch \(p. 715\)](#).

The high memory instances are designed to run large in-memory databases, including production deployments of the SAP HANA in-memory database, in the cloud. To maximize EBS performance, use high memory instances with an even number of `io1` or `io2` volumes with identical provisioned performance. For example, for IOPS heavy workloads, use four `io1` or `io2` volumes with 40,000 provisioned IOPS to get the maximum 160,000 instance IOPS. Similarly, for throughput heavy workloads, use six `io1` or `io2` volumes with 48,000 provisioned IOPS to get the maximum 4,750 MB/s throughput. For additional recommendations, see [Storage Configuration for SAP HANA](#).

Considerations

- G4, I3en, Inf1, M5a, M5ad, R5a, R5ad, T3, T3a, and Z1d instances launched after February 26, 2020 provide the maximum performance listed in the table above. To get the maximum performance from an instance launched before February 26, 2020, stop and start it.
- C5, C5d, C5n, M5, M5d, M5n, M5dn, R5, R5d, R5n, R5dn, and P3dn instances launched after December 3, 2019 provide the maximum performance listed in the table above. To get the maximum performance from an instance launched before December 3, 2019, stop and start it.
- `u-6tb1.metal`, `u-9tb1.metal`, and `u-12tb1.metal` instances launched after March 12, 2020 provide the performance in the table above. Instances of these types launched before March 12, 2020 might provide lower performance. To get the maximum performance from an instance launched before March 12, 2020, contact your account team to upgrade the instance at no additional cost.

Enabling EBS optimization at launch

You can enable optimization for an instance by setting its attribute for EBS optimization.

To enable Amazon EBS optimization when launching an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. In **Step 1: Choose an Amazon Machine Image (AMI)**, select an AMI.
4. In **Step 2: Choose an Instance Type**, select an instance type that is listed as supporting Amazon EBS optimization.
5. In **Step 3: Configure Instance Details**, complete the fields that you need and choose **Launch as EBS-optimized instance**. If the instance type that you selected in the previous step doesn't support Amazon EBS optimization, this option is not present. If the instance type that you selected is Amazon EBS-optimized by default, this option is selected and you can't deselect it.
6. Follow the directions to complete the wizard and launch your instance.

To enable EBS optimization when launching an instance using the command line

You can use one of the following commands with the corresponding option. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [run-instances](#) with `--ebs-optimized` (AWS CLI)
- [New-EC2Instance](#) with `-EbsOptimized` (AWS Tools for Windows PowerShell)

Enable EBS optimization for an existing instance

You can enable or disable optimization for an existing instance by modifying its Amazon EBS-optimized instance attribute. If the instance is running, you must stop it first.

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

To enable EBS optimization for an existing instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and select the instance.
3. To stop the instance, choose **Actions**, **Instance state**, **Stop instance**. It can take a few minutes for the instance to stop.
4. With the instance still selected, choose **Actions**, **Instance settings**, **Change instance type**.
5. For **Change Instance Type**, do one of the following:
 - If the instance type of your instance is Amazon EBS-optimized by default, **EBS-optimized** is selected and you can't change it. You can choose **Cancel**, because Amazon EBS optimization is already enabled for the instance.
 - If the instance type of your instance supports Amazon EBS optimization, choose **EBS-optimized** and then choose **Apply**.
 - If the instance type of your instance does not support Amazon EBS optimization, you can't choose **EBS-optimized**. You can select an instance type from **Instance type** that supports Amazon EBS optimization, choose **EBS-optimized**, and then choose **Apply**.
6. Choose **Actions**, **Instance state**, **Start instance**.

To enable EBS optimization for an existing instance using the command line

1. If the instance is running, use one of the following commands to stop it:
 - [stop-instances](#) (AWS CLI)
 - [Stop-EC2Instance](#) (AWS Tools for Windows PowerShell)
2. To enable EBS optimization, use one of the following commands with the corresponding option:
 - [modify-instance-attribute](#) with `--ebs-optimized` (AWS CLI)
 - [Edit-EC2InstanceAttribute](#) with `-EbsOptimized` (AWS Tools for Windows PowerShell)

Amazon EBS volume performance on Linux instances

Several factors, including I/O characteristics and the configuration of your instances and volumes, can affect the performance of Amazon EBS. Customers who follow the guidance on our Amazon EBS and Amazon EC2 product detail pages typically achieve good performance out of the box. However, there are some cases where you may need to do some tuning in order to achieve peak performance on the platform. This topic discusses general best practices as well as performance tuning that is specific to certain use cases. We recommend that you tune performance with information from your actual workload, in addition to benchmarking, to determine your optimal configuration. After you learn the basics of working with EBS volumes, it's a good idea to look at the I/O performance you require and at your options for increasing Amazon EBS performance to meet those requirements.

AWS updates to the performance of EBS volume types might not immediately take effect on your existing volumes. To see full performance on an older volume, you might first need to perform a [ModifyVolume](#) action on it. For more information, see [Modifying the Size, IOPS, or Type of an EBS Volume on Linux](#).

Contents

- [Amazon EBS performance tips \(p. 1157\)](#)
- [I/O characteristics and monitoring \(p. 1159\)](#)
- [Initializing Amazon EBS volumes \(p. 1161\)](#)
- [RAID Configuration on Linux \(p. 1163\)](#)
- [Benchmark EBS volumes \(p. 1167\)](#)

Amazon EBS performance tips

These tips represent best practices for getting optimal performance from your EBS volumes in a variety of user scenarios.

Use EBS-optimized instances

On instances without support for EBS-optimized throughput, network traffic can contend with traffic between your instance and your EBS volumes; on EBS-optimized instances, the two types of traffic are kept separate. Some EBS-optimized instance configurations incur an extra cost (such as C3, R3, and M3), while others are always EBS-optimized at no extra cost (such as M4, C4, C5, and D2). For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#).

Understand how performance is calculated

When you measure the performance of your EBS volumes, it is important to understand the units of measure involved and how performance is calculated. For more information, see [I/O characteristics and monitoring \(p. 1159\)](#).

Understand your workload

There is a relationship between the maximum performance of your EBS volumes, the size and number of I/O operations, and the time it takes for each action to complete. Each of these factors (performance, I/O, and latency) affects the others, and different applications are more sensitive to one factor or another. For more information, see [Benchmark EBS volumes \(p. 1167\)](#).

Be aware of the performance penalty When initializing volumes from snapshots

There is a significant increase in latency when you first access each block of data on a new EBS volume that was created from a snapshot. You can avoid this performance hit using one of the following options:

- Access each block prior to putting the volume into production. This process is called *initialization* (formerly known as pre-warming). For more information, see [Initializing Amazon EBS volumes \(p. 1161\)](#).
- Enable fast snapshot restore on a snapshot to ensure that the EBS volumes created from it are fully-initialized at creation and instantly deliver all of their provisioned performance. For more information, see [Amazon EBS fast snapshot restore \(p. 1132\)](#).

Factors that can degrade HDD performance

When you create a snapshot of a Throughput Optimized HDD (st1) or Cold HDD (sc1) volume, performance may drop as far as the volume's baseline value while the snapshot is in progress. This

behavior is specific to these volume types. Other factors that can limit performance include driving more throughput than the instance can support, the performance penalty encountered while initializing volumes created from a snapshot, and excessive amounts of small, random I/O on the volume. For more information about calculating throughput for HDD volumes, see [Amazon EBS volume types \(p. 1023\)](#).

Your performance can also be impacted if your application isn't sending enough I/O requests. This can be monitored by looking at your volume's queue length and I/O size. The queue length is the number of pending I/O requests from your application to your volume. For maximum consistency, HDD-backed volumes must maintain a queue length (rounded to the nearest whole number) of 4 or more when performing 1 MiB sequential I/O. For more information about ensuring consistent performance of your volumes, see [I/O characteristics and monitoring \(p. 1159\)](#)

Increase read-ahead for high-throughput, read-heavy workloads on `st1` and `sc1`

Some workloads are read-heavy and access the block device through the operating system page cache (for example, from a file system). In this case, to achieve the maximum throughput, we recommend that you configure the read-ahead setting to 1 MiB. This is a per-block-device setting that should only be applied to your HDD volumes.

To examine the current value of read-ahead for your block devices, use the following command:

```
[ec2-user ~]$ sudo blockdev --report /dev/<device>
```

Block device information is returned in the following format:

RO	RA	SSZ	BSZ	StartSec	Size	Device
rw	256	512	4096	4096	8587820544	/dev/<device>

The device shown reports a read-ahead value of 256 (the default). Multiply this number by the sector size (512 bytes) to obtain the size of the read-ahead buffer, which in this case is 128 KiB. To set the buffer value to 1 MiB, use the following command:

```
[ec2-user ~]$ sudo blockdev --setra 2048 /dev/<device>
```

Verify that the read-ahead setting now displays 2,048 by running the first command again.

Only use this setting when your workload consists of large, sequential I/Os. If it consists mostly of small, random I/Os, this setting will actually degrade your performance. In general, if your workload consists mostly of small or random I/Os, you should consider using a General Purpose SSD (`gp2`) volume rather than `st1` or `sc1`.

Use a modern Linux kernel

Use a modern Linux kernel with support for indirect descriptors. Any Linux kernel 3.8 and above has this support, as well as any current-generation EC2 instance. If your average I/O size is at or near 44 KiB, you may be using an instance or kernel without support for indirect descriptors. For information about deriving the average I/O size from Amazon CloudWatch metrics, see [I/O characteristics and monitoring \(p. 1159\)](#).

To achieve maximum throughput on `st1` or `sc1` volumes, we recommend applying a value of 256 to the `xen_blkfront.max` parameter (for Linux kernel versions below 4.6) or the `xen_blkfront.max_indirect_segments` parameter (for Linux kernel version 4.6 and above). The appropriate parameter can be set in your OS boot command line.

For example, in an Amazon Linux AMI with an earlier kernel, you can add it to the end of the kernel line in the GRUB configuration found in `/boot/grub/menu.lst`:

```
kernel /boot/vmlinuz-4.4.5-15.26.amzn1.x86_64 root=LABEL=/ console=ttyS0
xen_blkfront.max=256
```

For a later kernel, the command would be similar to the following:

```
kernel /boot/vmlinuz-4.9.20-11.31.amzn1.x86_64 root=LABEL=/ console=tty1 console=ttyS0
xen_blkfront.max_indirect_segments=256
```

Reboot your instance for this setting to take effect.

For more information, see [Configuring GRUB \(p. 194\)](#). Other Linux distributions, especially those that do not use the GRUB boot loader, may require a different approach to adjusting the kernel parameters.

For more information about EBS I/O characteristics, see the [Amazon EBS: Designing for Performance](#) re:Invent presentation on this topic.

Use RAID 0 to maximize utilization of instance resources

Some instance types can drive more I/O throughput than what you can provision for a single EBS volume. You can join multiple gp2, io1, io2, st1, or sc1 volumes together in a RAID 0 configuration to use the available bandwidth for these instances. For more information, see [RAID Configuration on Linux \(p. 1163\)](#).

Track performance using Amazon CloudWatch

Amazon Web Services provides performance metrics for Amazon EBS that you can analyze and view with Amazon CloudWatch and status checks that you can use to monitor the health of your volumes. For more information, see [Monitoring the status of your volumes \(p. 1051\)](#).

I/O characteristics and monitoring

On a given volume configuration, certain I/O characteristics drive the performance behavior for your EBS volumes. SSD-backed volumes—General Purpose SSD (gp2) and Provisioned IOPS SSD (io1 and io2)—deliver consistent performance whether an I/O operation is random or sequential. HDD-backed volumes—Throughput Optimized HDD (st1) and Cold HDD (sc1)—deliver optimal performance only when I/O operations are large and sequential. To understand how SSD and HDD volumes will perform in your application, it is important to know the connection between demand on the volume, the quantity of IOPS available to it, the time it takes for an I/O operation to complete, and the volume's throughput limits.

IOPS

IOPS are a unit of measure representing input/output operations per second. The operations are measured in KiB, and the underlying drive technology determines the maximum amount of data that a volume type counts as a single I/O. I/O size is capped at 256 KiB for SSD volumes and 1,024 KiB for HDD volumes because SSD volumes handle small or random I/O much more efficiently than HDD volumes.

When small I/O operations are physically contiguous, Amazon EBS attempts to merge them into a single I/O operation up to the maximum size. For example, for SSD volumes, a single 1,024 KiB I/O operation counts as 4 operations ($1,024 \div 256 = 4$), while 8 contiguous I/O operations at 32 KiB each count as 1 operation ($8 \times 32 = 256$). However, 8 random non-contiguous I/O operations at 32 KiB each count as 8 operations. In this case, each I/O operation under 32 KiB counts as 1 operation.

Similarly, for HDD-backed volumes, both a single 1,024 KiB I/O operation and 8 sequential 128 KiB operations would count as one operation. However, 8 random 128 KiB I/O operations would count as 8 operations.

Consequently, when you create an SSD-backed volume supporting 3,000 IOPS (either by provisioning an `io1` or `io2` volume at 3,000 IOPS or by sizing a `gp2` volume at 1000 GiB), and you attach it to an EBS-optimized instance that can provide sufficient bandwidth, you can transfer up to 3,000 I/Os of data per second, with throughput determined by I/O size.

Volume queue length and latency

The volume queue length is the number of pending I/O requests for a device. Latency is the true end-to-end client time of an I/O operation, in other words, the time elapsed between sending an I/O to EBS and receiving an acknowledgement from EBS that the I/O read or write is complete. Queue length must be correctly calibrated with I/O size and latency to avoid creating bottlenecks either on the guest operating system or on the network link to EBS.

Optimal queue length varies for each workload, depending on your particular application's sensitivity to IOPS and latency. If your workload is not delivering enough I/O requests to fully use the performance available to your EBS volume, then your volume might not deliver the IOPS or throughput that you have provisioned.

Transaction-intensive applications are sensitive to increased I/O latency and are well-suited for SSD-backed `io1`, `io2`, and `gp2` volumes. You can maintain high IOPS while keeping latency down by maintaining a low queue length and a high number of IOPS available to the volume. Consistently driving more IOPS to a volume than it has available can cause increased I/O latency.

Throughput-intensive applications are less sensitive to increased I/O latency, and are well-suited for HDD-backed `st1` and `sc1` volumes. You can maintain high throughput to HDD-backed volumes by maintaining a high queue length when performing large, sequential I/O.

I/O size and volume throughput limits

For SSD-backed volumes, if your I/O size is very large, you may experience a smaller number of IOPS than you provisioned because you are hitting the throughput limit of the volume. For example, a `gp2` volume under 1000 GiB with burst credits available has an IOPS limit of 3,000 and a volume throughput limit of 250 MiB/s. If you are using a 256 KiB I/O size, your volume reaches its throughput limit at 1000 IOPS ($1000 \times 256 \text{ KiB} = 250 \text{ MiB}$). For smaller I/O sizes (such as 16 KiB), this same volume can sustain 3,000 IOPS because the throughput is well below 250 MiB/s. (These examples assume that your volume's I/O is not hitting the throughput limits of the instance.) For more information about the throughput limits for each EBS volume type, see [Amazon EBS volume types \(p. 1023\)](#).

For smaller I/O operations, you may see a higher-than-provisioned IOPS value as measured from inside your instance. This happens when the instance operating system merges small I/O operations into a larger operation before passing them to Amazon EBS.

If your workload uses sequential I/Os on HDD-backed `st1` and `sc1` volumes, you may experience a higher than expected number of IOPS as measured from inside your instance. This happens when the instance operating system merges sequential I/Os and counts them in 1,024 KiB-sized units. If your workload uses small or random I/Os, you may experience a lower throughput than you expect. This is because we count each random, non-sequential I/O toward the total IOPS count, which can cause you to hit the volume's IOPS limit sooner than expected.

Whatever your EBS volume type, if you are not experiencing the IOPS or throughput you expect in your configuration, ensure that your EC2 instance bandwidth is not the limiting factor. You should always use a current-generation, EBS-optimized instance (or one that includes 10 Gb/s network connectivity) for optimal performance. For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#). Another possible cause for not experiencing the expected IOPS is that you are not driving enough I/O to the EBS volumes.

Monitor I/O characteristics using CloudWatch

You can monitor these I/O characteristics with each volume's [CloudWatch volume metrics \(p. 1172\)](#). Important metrics to consider include the following:

- `BurstBalance`
- `VolumeReadBytes`
- `VolumeWriteBytes`
- `VolumeReadOps`
- `VolumeWriteOps`
- `VolumeQueueLength`

`BurstBalance` displays the burst bucket balance for gp2, st1, and sc1 volumes as a percentage of the remaining balance. When your burst bucket is depleted, volume I/O (for gp2 volumes) or volume throughput (for st1 and sc1 volumes) is throttled to the baseline. Check the `BurstBalance` value to determine whether your volume is being throttled for this reason.

HDD-backed st1 and sc1 volumes are designed to perform best with workloads that take advantage of the 1,024 KiB maximum I/O size. To determine your volume's average I/O size, divide `VolumeWriteBytes` by `VolumeWriteOps`. The same calculation applies to read operations. If average I/O size is below 64 KiB, increasing the size of the I/O operations sent to an st1 or sc1 volume should improve performance.

Note

If average I/O size is at or near 44 KiB, you might be using an instance or kernel without support for indirect descriptors. Any Linux kernel 3.8 and above has this support, as well as any current-generation instance.

If your I/O latency is higher than you require, check `VolumeQueueLength` to make sure your application is not trying to drive more IOPS than you have provisioned. If your application requires a greater number of IOPS than your volume can provide, you should consider using a larger gp2 volume with a higher base performance level or an io1 or io2 volume with more provisioned IOPS to achieve faster latencies.

Related resources

For more information about Amazon EBS I/O characteristics, see the following re:Invent presentation: [Amazon EBS: Designing for Performance](#).

Initializing Amazon EBS volumes

Empty EBS volumes receive their maximum performance the moment that they are created and do not require initialization (formerly known as pre-warming).

For volumes that were created from snapshots, the storage blocks must be pulled down from Amazon S3 and written to the volume before you can access them. This preliminary action takes time and can cause a significant increase in the latency of I/O operations the first time each block is accessed. Volume performance is achieved after all blocks have been downloaded and written to the volume.

Important

While initializing io1 and io2 volumes that were created from snapshots, the performance of the volume may drop below 50 percent of its expected level, which causes the volume to display a warning state in the **I/O Performance** status check. This is expected, and you can ignore the warning state on io1 and io2 volumes while you are initializing them. For more information, see [EBS volume status checks \(p. 1051\)](#).

For most applications, amortizing the initialization cost over the lifetime of the volume is acceptable. To avoid this initial performance hit in a production environment, you can use one of the following options:

- Force the immediate initialization of the entire volume. For more information, see [Initializing Amazon EBS volumes on Linux \(p. 1162\)](#).

- Enable fast snapshot restore on a snapshot to ensure that the EBS volumes created from it are fully-initialized at creation and instantly deliver all of their provisioned performance. For more information, see [Amazon EBS fast snapshot restore \(p. 1132\)](#).

Initializing Amazon EBS volumes on Linux

Empty EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). For volumes that have been created from snapshots, use the **dd** or **fio** utilities to read from all of the blocks on a volume. All existing data on the volume will be preserved.

For information about initializing Amazon EBS volumes on Windows, see [Initializing Amazon EBS volumes on Windows](#).

To initialize a volume created from a snapshot on Linux

1. Attach the newly-restored volume to your Linux instance.
2. Use the **lsblk** command to list the block devices on your instance.

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdf  202:80   0 30G  0 disk
xvda1 202:1    0  8G  0 disk /
```

Here you can see that the new volume, `/dev/xvdf`, is attached, but not mounted (because there is no path listed under the `MOUNTPOINT` column).

3. Use the **dd** or **fio** utilities to read all of the blocks on the device. The **dd** command is installed by default on Linux systems, but **fio** is considerably faster because it allows multi-threaded reads.

Note

This step may take several minutes up to several hours, depending on your EC2 instance bandwidth, the IOPS provisioned for the volume, and the size of the volume.

[dd] The `if` (input file) parameter should be set to the drive you wish to initialize. The `of` (output file) parameter should be set to the Linux null virtual device, `/dev/null`. The `bs` parameter sets the block size of the read operation; for optimal performance, this should be set to 1 MB.

Important

Incorrect use of **dd** can easily destroy a volume's data. Be sure to follow precisely the example command below. Only the `if=/dev/xvdf` parameter will vary depending on the name of the device you are reading.

```
[ec2-user ~]$ sudo dd if=/dev/xvdf of=/dev/null bs=1M
```

[fio] If you have **fio** installed on your system, use the following command to initialize your volume. The `--filename` (input file) parameter should be set to the drive you wish to initialize.

```
[ec2-user ~]$ sudo fio --filename=/dev/xvdf --rw=read --bs=128k --iodepth=32 --
ioengine=libaio --direct=1 --name=volume-initialize
```

To install **fio** on Amazon Linux, use the following command:

```
sudo yum install -y fio
```

To install **fio** on Ubuntu, use the following command:

```
sudo apt-get install -y fio
```

When the operation is finished, you will see a report of the read operation. Your volume is now ready for use. For more information, see [Making an Amazon EBS volume available for use on Linux \(p. 1046\)](#).

RAID Configuration on Linux

With Amazon EBS, you can use any of the standard RAID configurations that you can use with a traditional bare metal server, as long as that particular RAID configuration is supported by the operating system for your instance. This is because all RAID is accomplished at the software level. For greater I/O performance than you can achieve with a single volume, RAID 0 can stripe multiple volumes together; for on-instance redundancy, RAID 1 can mirror two volumes together.

Amazon EBS volume data is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component. This replication makes Amazon EBS volumes ten times more reliable than typical commodity disk drives. For more information, see [Amazon EBS Availability and Durability](#) in the Amazon EBS product detail pages.

Note

You should avoid booting from a RAID volume. Grub is typically installed on only one device in a RAID array, and if one of the mirrored devices fails, you may be unable to boot the operating system.

If you need to create a RAID array on a Windows instance, see [RAID Configuration on Windows](#) in the [Amazon EC2 User Guide for Windows Instances](#).

Contents

- [RAID Configuration Options \(p. 1163\)](#)
- [Creating a RAID Array on Linux \(p. 1164\)](#)
- [Creating Snapshots of Volumes in a RAID Array \(p. 1167\)](#)

RAID Configuration Options

The following table compares the common RAID 0 and RAID 1 options.

Configuration	Use	Advantages	Disadvantages
RAID 0	When I/O performance is more important than fault tolerance; for example, as in a heavily used database (where data replication is already set up separately).	I/O is distributed across the volumes in a stripe. If you add a volume, you get the straight addition of throughput and IOPS.	Performance of the stripe is limited to the worst performing volume in the set. Loss of a single volume results in a complete data loss for the array.
RAID 1	When fault tolerance is more important than I/O performance; for example, as in a critical application.	Safer from the standpoint of data durability.	Does not provide a write performance improvement; requires more Amazon EC2 to Amazon EBS bandwidth than non-RAID configurations because the data is written to multiple volumes simultaneously.

Important

RAID 5 and RAID 6 are not recommended for Amazon EBS because the parity write operations of these RAID modes consume some of the IOPS available to your volumes. Depending on the configuration of your RAID array, these RAID modes provide 20-30% fewer usable IOPS than a RAID 0 configuration. Increased cost is a factor with these RAID modes as well; when using identical volume sizes and speeds, a 2-volume RAID 0 array can outperform a 4-volume RAID 6 array that costs twice as much.

Creating a RAID 0 array allows you to achieve a higher level of performance for a file system than you can provision on a single Amazon EBS volume. A RAID 1 array offers a "mirror" of your data for extra redundancy. Before you perform this procedure, you need to decide how large your RAID array should be and how many IOPS you want to provision.

The resulting size of a RAID 0 array is the sum of the sizes of the volumes within it, and the bandwidth is the sum of the available bandwidth of the volumes within it. The resulting size and bandwidth of a RAID 1 array is equal to the size and bandwidth of the volumes in the array. For example, two 500 GiB Amazon EBS io1 volumes with 4,000 provisioned IOPS each will create a 1000 GiB RAID 0 array with an available bandwidth of 8,000 IOPS and 1,000 MiB/s of throughput or a 500 GiB RAID 1 array with an available bandwidth of 4,000 IOPS and 500 MiB/s of throughput.

This documentation provides basic RAID setup examples. For more information about RAID configuration, performance, and recovery, see the Linux RAID Wiki at https://raid.wiki.kernel.org/index.php/Linux_Raid.

Creating a RAID Array on Linux

Use the following procedure to create the RAID array. Note that you can get directions for Windows instances from [Creating a RAID Array on Windows](#) in the *Amazon EC2 User Guide for Windows Instances*.

To create a RAID array on Linux

1. Create the Amazon EBS volumes for your array. For more information, see [Creating an Amazon EBS volume \(p. 1040\)](#).

Important

Create volumes with identical size and IOPS performance values for your array. Make sure you do not create an array that exceeds the available bandwidth of your EC2 instance. For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#).

2. Attach the Amazon EBS volumes to the instance that you want to host the array. For more information, see [Attaching an Amazon EBS volume to an instance \(p. 1042\)](#).
3. Use the **mdadm** command to create a logical RAID device from the newly attached Amazon EBS volumes. Substitute the number of volumes in your array for **number_of_volumes** and the device names for each volume in the array (such as `/dev/xvdf`) for **device_name**. You can also substitute **MY_RAID** with your own unique name for the array.

Note

You can list the devices on your instance with the **lsblk** command to find the device names.

(RAID 0 only) To create a RAID 0 array, execute the following command (note the `--level=0` option to stripe the array):

```
[ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=0 --name=MY_RAID --raid-devices=number_of_volumes device_name1 device_name2
```

(RAID 1 only) To create a RAID 1 array, execute the following command (note the `--level=1` option to mirror the array):

```
[ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=1 --name=MY_RAID --raid-devices=number_of_volumes device_name1 device_name2
```

4. Allow time for the RAID array to initialize and synchronize. You can track the progress of these operations with the following command:

```
[ec2-user ~]$ sudo cat /proc/mdstat
```

The following is example output:

```
Personalities : [raid1]
md0 : active raid1 xvdf[1] xvdf[0]
      20955008 blocks super 1.2 [2/2] [UU]
      [======>.....]  resync = 46.8% (9826112/20955008) finish=2.9min
speed=63016K/sec
```

In general, you can display detailed information about your RAID array with the following command:

```
[ec2-user ~]$ sudo mdadm --detail /dev/md0
```

The following is example output:

```
/dev/md0:
      Version : 1.2
      Creation Time : Mon Jun 27 11:31:28 2016
      Raid Level : raid1
      Array Size : 20955008 (19.98 GiB 21.46 GB)
      Used Dev Size : 20955008 (19.98 GiB 21.46 GB)
      Raid Devices : 2
      Total Devices : 2
      Persistence : Superblock is persistent

      Update Time : Mon Jun 27 11:37:02 2016
      State : clean
      ...
      ...

      Number  Major  Minor  RaidDevice State
          0      202     80        0    active sync   /dev/sdf
          1      202     96        1    active sync   /dev/sdg
```

5. Create a file system on your RAID array, and give that file system a label to use when you mount it later. For example, to create an ext4 file system with the label **MY_RAID**, execute the following command:

```
[ec2-user ~]$ sudo mkfs.ext4 -L MY_RAID /dev/md0
```

Depending on the requirements of your application or the limitations of your operating system, you can use a different file system type, such as ext3 or XFS (consult your file system documentation for the corresponding file system creation command).

6. To ensure that the RAID array is reassembled automatically on boot, create a configuration file to contain the RAID information:

```
[ec2-user ~]$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm.conf
```

Note

If you are using a Linux distribution other than Amazon Linux, this file may need to be placed in different location. For more information, consult **man mdadm.conf** on your Linux system..

7. Create a new ramdisk image to properly preload the block device modules for your new RAID configuration:

```
[ec2-user ~]$ sudo dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

8. Create a mount point for your RAID array.

```
[ec2-user ~]$ sudo mkdir -p /mnt/raid
```

9. Finally, mount the RAID device on the mount point that you created:

```
[ec2-user ~]$ sudo mount LABEL=MY_RAID /mnt/raid
```

Your RAID device is now ready for use.

10. (Optional) To mount this Amazon EBS volume on every system reboot, add an entry for the device to the `/etc/fstab` file.

- a. Create a backup of your `/etc/fstab` file that you can use if you accidentally destroy or delete this file while you are editing it.

```
[ec2-user ~]$ sudo cp /etc/fstab /etc/fstab.orig
```

- b. Open the `/etc/fstab` file using your favorite text editor, such as **nano** or **vim**.
- c. Comment out any lines starting with "UUID=" and, at the end of the file, add a new line for your RAID volume using the following format:

```
device_label mount_point file_system_type fs_mntops fs_freq fs_passno
```

The last three fields on this line are the file system mount options, the dump frequency of the file system, and the order of file system checks done at boot time. If you don't know what these values should be, then use the values in the example below for them (`defaults,nofail 0 2`). For more information about `/etc/fstab` entries, see the **fstab** manual page (by entering **man fstab** on the command line). For example, to mount the ext4 file system on the device with the label `MY_RAID` at the mount point `/mnt/raid`, add the following entry to `/etc/fstab`.

Note

If you ever intend to boot your instance without this volume attached (for example, so this volume could move back and forth between different instances), you should add the `nofail` mount option that allows the instance to boot even if there are errors in mounting the volume. Debian derivatives, such as Ubuntu, must also add the `nobootwait` mount option.

```
LABEL=MY_RAID      /mnt/raid      ext4      defaults,nofail      0      2
```

- d. After you've added the new entry to `/etc/fstab`, you need to check that your entry works. Run the **sudo mount -a** command to mount all file systems in `/etc/fstab`.

```
[ec2-user ~]$ sudo mount -a
```

If the previous command does not produce an error, then your `/etc/fstab` file is OK and your file system will mount automatically at the next boot. If the command does produce any errors, examine the errors and try to correct your `/etc/fstab`.

Warning

Errors in the `/etc/fstab` file can render a system unbootable. Do not shut down a system that has errors in the `/etc/fstab` file.

- e. (Optional) If you are unsure how to correct `/etc/fstab` errors, you can always restore your backup `/etc/fstab` file with the following command.

```
[ec2-user ~]$ sudo mv /etc/fstab.orig /etc/fstab
```

Creating Snapshots of Volumes in a RAID Array

If you want to back up the data on the EBS volumes in a RAID array using snapshots, you must ensure that the snapshots are consistent. This is because the snapshots of these volumes are created independently. To restore EBS volumes in a RAID array from snapshots that are out of sync would degrade the integrity of the array.

To create a consistent set of snapshots for your RAID array, use [EBS multi-volume snapshots](#). Multi-volume snapshots allow you to take point-in-time, data coordinated, and crash-consistent snapshots across multiple EBS volumes attached to an EC2 instance. You do not have to stop your instance to coordinate between volumes to ensure consistency because snapshots are automatically taken across multiple EBS volumes. For more information, see the steps for creating multi-volume snapshots under [Creating Amazon EBS Snapshots](#).

Benchmark EBS volumes

You can test the performance of Amazon EBS volumes by simulating I/O workloads. The process is as follows:

1. Launch an EBS-optimized instance.
2. Create new EBS volumes.
3. Attach the volumes to your EBS-optimized instance.
4. Configure and mount the block device.
5. Install a tool to benchmark I/O performance.
6. Benchmark the I/O performance of your volumes.
7. Delete your volumes and terminate your instance so that you don't continue to incur charges.

Important

Some of the procedures result in the destruction of existing data on the EBS volumes you benchmark. The benchmarking procedures are intended for use on volumes specially created for testing purposes, not production volumes.

Set up your instance

To get optimal performance from EBS volumes, we recommend that you use an EBS-optimized instance. EBS-optimized instances deliver dedicated throughput between Amazon EC2 and Amazon EBS, with instance. EBS-optimized instances deliver dedicated bandwidth between Amazon EC2 and Amazon EBS, with specifications depending on the instance type. For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#).

To create an EBS-optimized instance, choose **Launch as an EBS-Optimized instance** when launching the instance using the Amazon EC2 console, or specify `--ebs-optimized` when using the command line. Be

sure that you launch a current-generation instance that supports this option. For more information, see [Amazon EBS-optimized instances \(p. 1139\)](#).

Setting up Provisioned IOPS SSD (`io1` and `io2`) volumes

To create an `io1` or `io2` volume, choose **Provisioned IOPS SSD (`io1`)** or **Provisioned IOPS SSD (`io2`)** when creating the volume using the Amazon EC2 console, or, at the command line, specify `--volume-type io1|io2 --iops n` where *n* is an integer between 100 and 64,000. For more detailed EBS-volume specifications, see [Amazon EBS volume types \(p. 1023\)](#). For information about creating an EBS volume, see [Creating an Amazon EBS volume \(p. 1040\)](#). For information about attaching a volume to an instance, see [Attaching an Amazon EBS volume to an instance \(p. 1042\)](#).

For the example tests, we recommend that you create a RAID array with 6 volumes, which offers a high level of performance. Because you are charged by gigabytes provisioned (and the number of provisioned IOPS for `io1` and `io2` volumes), not the number of volumes, there is no additional cost for creating multiple, smaller volumes and using them to create a stripe set. If you're using Oracle Orion to benchmark your volumes, it can simulate striping the same way that Oracle ASM does, so we recommend that you let Orion do the striping. If you are using a different benchmarking tool, you need to stripe the volumes yourself.

To create a six-volume stripe set on Amazon Linux, use a command such as the following:

```
[ec2-user ~]$ sudo mdadm --create /dev/md0 --level=0 --chunk=64 --raid-devices=6 /dev/sdf /dev/sdg /dev/sdh /dev/sdi /dev/sdj /dev/sdk
```

For this example, the file system is XFS. Use the file system that meets your requirements. Use the following command to install XFS file system support:

```
[ec2-user ~]$ sudo yum install -y xfsprogs
```

Then, use these commands to create, mount, and assign ownership to the XFS file system:

```
[ec2-user ~]$ sudo mkdir -p /mnt/p_iops_vo10 && sudo mkfs.xfs /dev/md0
[ec2-user ~]$ sudo mount -t xfs /dev/md0 /mnt/p_iops_vo10
[ec2-user ~]$ sudo chown ec2-user:ec2-user /mnt/p_iops_vo10/
```

Setting up Throughput Optimized HDD (`st1`) or Cold HDD (`sc1`) volumes

To create an `st1` volume, choose **Throughput Optimized HDD** when creating the volume using the Amazon EC2 console, or specify `--type st1` when using the command line. To create an `sc1` volume, choose **Cold HDD** when creating the volume using the Amazon EC2 console, or specify `--type sc1` when using the command line. For information about creating EBS volumes, see [Creating an Amazon EBS volume \(p. 1040\)](#). For information about attaching these volumes to your instance, see [Attaching an Amazon EBS volume to an instance \(p. 1042\)](#).

AWS provides a JSON template for use with AWS CloudFormation that simplifies this setup procedure. Access the [template](#) and save it as a JSON file. AWS CloudFormation allows you to configure your own SSH keys and offers an easier way to set up a performance test environment to evaluate `st1` volumes. The template creates a current-generation instance and a 2 TiB `st1` volume, and attaches the volume to the instance at `/dev/xvdf`.

To create an HDD volume using the template

1. Open the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation>.
2. Choose **Create Stack**.
3. Choose **Upload a Template to Amazon S3** and select the JSON template you previously obtained.

4. Give your stack a name like “ebs-perf-testing”, and select an instance type (the default is r3.8xlarge) and SSH key.
5. Choose **Next** twice, and then choose **Create Stack**.
6. After the status for your new stack moves from **CREATE_IN_PROGRESS** to **COMPLETE**, choose **Outputs** to get the public DNS entry for your new instance, which will have a 2 TiB st1 volume attached to it.
7. Connect using SSH to your new stack as user **ec2-user**, with the hostname obtained from the DNS entry in the previous step.
8. Proceed to [Install benchmark tools \(p. 1169\)](#).

Install benchmark tools

The following table lists some of the possible tools you can use to benchmark the performance of EBS volumes.

Tool	Description
fio	<p>For benchmarking I/O performance. (Note that fio has a dependency on libaio-devel.)</p> <p>To install fio on Amazon Linux, run the following command:</p> <div style="border: 1px solid black; padding: 5px;"><pre>[ec2-user ~]\$ sudo yum install -y fio</pre></div> <p>To install fio on Ubuntu, run the following command:</p> <div style="border: 1px solid black; padding: 5px;"><pre>sudo apt-get install -y fio</pre></div>
Oracle Orion Calibration Tool	For calibrating the I/O performance of storage systems to be used with Oracle databases.

These benchmarking tools support a wide variety of test parameters. You should use commands that approximate the workloads your volumes will support. These commands provided below are intended as examples to help you get started.

Choosing the volume queue length

Choosing the best volume queue length based on your workload and volume type.

Queue length on SSD-backed volumes

To determine the optimal queue length for your workload on SSD-backed volumes, we recommend that you target a queue length of 1 for every 1000 IOPS available (baseline for gp2 volumes and the provisioned amount for io1 and io2 volumes). Then you can monitor your application performance and tune that value based on your application requirements.

Increasing the queue length is beneficial until you achieve the provisioned IOPS, throughput or optimal system queue length value, which is currently set to 32. For example, a volume with 3,000 provisioned IOPS should target a queue length of 3. You should experiment with tuning these values up or down to see what performs best for your application.

Queue length on HDD-backed volumes

To determine the optimal queue length for your workload on HDD-backed volumes, we recommend that you target a queue length of at least 4 while performing 1MiB sequential I/Os. Then you can monitor

your application performance and tune that value based on your application requirements. For example, a 2 TiB `st1` volume with burst throughput of 500 MiB/s and IOPS of 500 should target a queue length of 4, 8, or 16 while performing 1,024 KiB, 512 KiB, or 256 KiB sequential I/Os respectively. You should experiment with tuning these values up or down to see what performs best for your application.

Disable C-states

Before you run benchmarking, you should disable processor C-states. Temporarily idle cores in a supported CPU can enter a C-state to save power. When the core is called on to resume processing, a certain amount of time passes until the core is again fully operational. This latency can interfere with processor benchmarking routines. For more information about C-states and which EC2 instance types support them, see [Processor State Control for Your EC2 Instance](#).

Disabling C-states on Linux

You can disable C-states on Amazon Linux, RHEL, and CentOS as follows:

1. Get the number of C-states.

```
$ cpupower idle-info | grep "Number of idle states:"
```

2. Disable the C-states from `c1` to `cN`. Ideally, the cores should be in state `c0`.

```
$ for i in `seq 1 $((N-1))` ; do cpupower idle-set -d $i; done
```

Perform benchmarking

The following procedures describe benchmarking commands for various EBS volume types.

Run the following commands on an EBS-optimized instance with attached EBS volumes. If the EBS volumes were created from snapshots, be sure to initialize them before benchmarking. For more information, see [Initializing Amazon EBS volumes \(p. 1161\)](#).

When you are finished testing your volumes, see the following topics for help cleaning up: [Deleting an Amazon EBS volume \(p. 1060\)](#) and [Terminate your instance \(p. 607\)](#).

Benchmarking `io1` and `io2` volumes

Run `fio` on the stripe set that you created.

The following command performs 16 KB random write operations.

```
[ec2-user ~]$ sudo fio --directory=/mnt/p_iops_volo --name fio_test_file --direct=1 --rw=randwrite --bs=16k --size=1G --numjobs=16 --time_based --runtime=180 --group_reporting --norandommap
```

The following command performs 16 KB random read operations.

```
[ec2-user ~]$ sudo fio --directory=/mnt/p_iops_volo --name fio_test_file --direct=1 --rw=randread --bs=16k --size=1G --numjobs=16 --time_based --runtime=180 --group_reporting --norandommap
```

For more information about interpreting the results, see this tutorial: [Inspecting disk IO performance with fio](#).

Benchmarking `st1` and `sc1` volumes

Run `fio` on your `st1` or `sc1` volume.

Note

Prior to running these tests, set buffered I/O on your instance as described in [Increase read-ahead for high-throughput, read-heavy workloads on st1 and sc1 \(p. 1158\)](#).

The following command performs 1 MiB sequential read operations against an attached st1 block device (e.g., /dev/xvdf):

```
[ec2-user ~]$ sudo fio --filename=/dev/<device> --direct=1 --rw=read --randrepeat=0  
--ioengine=libaio --bs=1024k --iodepth=8 --time_based=1 --runtime=180 --  
name=fio_direct_read_test
```

The following command performs 1 MiB sequential write operations against an attached st1 block device:

```
[ec2-user ~]$ sudo fio --filename=/dev/<device> --direct=1 --rw=write --randrepeat=0  
--ioengine=libaio --bs=1024k --iodepth=8 --time_based=1 --runtime=180 --  
name=fio_direct_write_test
```

Some workloads perform a mix of sequential reads and sequential writes to different parts of the block device. To benchmark such a workload, we recommend that you use separate, simultaneous **fio** jobs for reads and writes, and use the **fio offset_increment** option to target different block device locations for each job.

Running this workload is a bit more complicated than a sequential-write or sequential-read workload. Use a text editor to create a fio job file, called **fio_rw_mix.cfg** in this example, that contains the following:

```
[global]  
clocksource=clock_gettime  
randrepeat=0  
runtime=180  
offset_increment=100g  
  
[sequential-write]  
bs=1M  
ioengine=libaio  
direct=1  
iodepth=8  
filename=/dev/<device>  
do_verify=0  
rw=write  
rwmixread=0  
rwmixwrite=100  
  
[sequential-read]  
bs=1M  
ioengine=libaio  
direct=1  
iodepth=8  
filename=/dev/<device>  
do_verify=0  
rw=read  
rwmixread=100  
rwmixwrite=0
```

Then run the following command:

```
[ec2-user ~]$ sudo fio fio_rw_mix.cfg
```

For more information about interpreting the results, see this tutorial: [Inspecting disk I/O performance with fio](#).

Multiple **fio** jobs for direct I/O, even though using sequential read or write operations, can result in lower than expected throughput for `st1` and `sc1` volumes. We recommend that you use one direct I/O job and use the `iodepth` parameter to control the number of concurrent I/O operations.

Amazon CloudWatch metrics for Amazon EBS

CloudWatch metrics are statistical data that you can use to view, analyze, and set alarms on the operational behavior of your volumes.

The following table describes the types of monitoring data available for your Amazon EBS volumes.

Type	Description
Basic	Data is available automatically in 5-minute periods at no charge. This includes data for the root device volumes for EBS-backed instances.
Detailed	Provisioned IOPS SSD (<code>io1</code> and <code>io2</code>) volumes automatically send one-minute metrics to CloudWatch.

When you get data from CloudWatch, you can include a `Period` request parameter to specify the granularity of the returned data. This is different than the period that we use when we collect the data (5-minute periods). We recommend that you specify a period in your request that is equal to or larger than the collection period to ensure that the returned data is valid.

You can get the data using either the CloudWatch API or the Amazon EC2 console. The console takes the raw data from the CloudWatch API and displays a series of graphs based on the data. Depending on your needs, you might prefer to use either the data from the API or the graphs in the console.

Amazon EBS metrics

Amazon Elastic Block Store (Amazon EBS) sends data points to CloudWatch for several metrics. Amazon EBS General Purpose SSD (`gp2`), Throughput Optimized HDD (`st1`), Cold HDD (`sc1`), and Magnetic (standard) volumes automatically send five-minute metrics to CloudWatch. Provisioned IOPS SSD (`io1` and `io2`) volumes automatically send one-minute metrics to CloudWatch. Data is only reported to CloudWatch when the volume is attached to an instance.

Some of these metrics have differences on Nitro-based instances. For a list of instance types based on the Nitro system, see [Instances built on the Nitro System \(p. 205\)](#).

The `AWS/EBS` namespace includes the following metrics.

Metrics

- [Volume metrics \(p. 1172\)](#)
- [Fast snapshot restore metrics \(p. 1176\)](#)

Volume metrics

The `AWS/EBS` namespace includes the following metrics for EBS volumes. To get information about the available disk space from the operating system on an instance, see [Viewing free disk space \(p. 1050\)](#).

Metric	Description
VolumeReadBytes	<p>Provides information on the read operations in a specified period of time. The Sum statistic reports the total number of bytes transferred during the period. The Average statistic reports the average size of each read operation during the period, except on volumes attached to a Nitro-based instance, where the average represents the average over the specified period. The SampleCount statistic reports the total number of read operations during the period, except on volumes attached to a Nitro-based instance, where the sample count represents the number of data points used in the statistical calculation. For Xen instances, data is reported only when there is read activity on the volume.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Bytes</p>
VolumeWriteBytes	<p>Provides information on the write operations in a specified period of time. The Sum statistic reports the total number of bytes transferred during the period. The Average statistic reports the average size of each write operation during the period, except on volumes attached to a Nitro-based instance, where the average represents the average over the specified period. The SampleCount statistic reports the total number of write operations during the period, except on volumes attached to a Nitro-based instance, where the sample count represents the number of data points used in the statistical calculation. For Xen instances, data is reported only when there is write activity on the volume.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Bytes</p>
VolumeReadOps	<p>The total number of read operations in a specified period of time.</p> <p>To calculate the average read operations per second (read IOPS) for the period, divide the total read operations in the period by the number of seconds in that period.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Count</p>
VolumeWriteOps	<p>The total number of write operations in a specified period of time.</p> <p>To calculate the average write operations per second (write IOPS) for the period, divide the total write operations in the period by the number of seconds in that period.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p>

Metric	Description
	<p>Units: Count</p>
VolumeTotalReadTime	<p>Note This metric is not supported with Multi-Attach enabled volumes.</p> <p>The total number of seconds spent by all read operations that completed in a specified period of time. If multiple requests are submitted at the same time, this total could be greater than the length of the period. For example, for a period of 5 minutes (300 seconds): if 700 operations completed during that period, and each operation took 1 second, the value would be 700 seconds. For Xen instances, data is reported only when there is read activity on the volume.</p> <p>The Average statistic on this metric is not relevant for volumes attached to Nitro-based instances.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Seconds</p>
VolumeTotalWriteTime	<p>Note This metric is not supported with Multi-Attach enabled volumes.</p> <p>The total number of seconds spent by all write operations that completed in a specified period of time. If multiple requests are submitted at the same time, this total could be greater than the length of the period. For example, for a period of 5 minutes (300 seconds): if 700 operations completed during that period, and each operation took 1 second, the value would be 700 seconds. For Xen instances, data is reported only when there is write activity on the volume.</p> <p>The Average statistic on this metric is not relevant for volumes attached to Nitro-based instances.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Seconds</p>

Metric	Description
VolumeIdleTime	<p>Note This metric is not supported with Multi-Attach enabled volumes.</p> <p>The total number of seconds in a specified period of time when no read or write operations were submitted.</p> <p>The Average statistic on this metric is not relevant for volumes attached to Nitro-based instances.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Seconds</p>
VolumeQueueLength	<p>The number of read and write operation requests waiting to be completed in a specified period of time.</p> <p>The Sum statistic on this metric is not relevant for volumes attached to Nitro-based instances.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Count</p>
VolumeThroughputPercentage	<p>Note This metric is not supported with Multi-Attach enabled volumes.</p> <p>Used with Provisioned IOPS SSD volumes only. The percentage of I/O operations per second (IOPS) delivered of the total IOPS provisioned for an Amazon EBS volume. Provisioned IOPS SSD volumes deliver their provisioned performance 99.9 percent of the time.</p> <p>During a write, if there are no other pending I/O requests in a minute, the metric value will be 100 percent. Also, a volume's I/O performance may become degraded temporarily due to an action you have taken (for example, creating a snapshot of a volume during peak usage, running the volume on a non-EBS-optimized instance, or accessing data on the volume for the first time).</p> <p>Units: Percent</p>
VolumeConsumedReadWriteOps	<p>Used with Provisioned IOPS SSD volumes only. The total amount of read and write operations (normalized to 256K capacity units) consumed in a specified period of time.</p> <p>I/O operations that are smaller than 256K each count as 1 consumed IOPS. I/O operations that are larger than 256K are counted in 256K capacity units. For example, a 1024K I/O would count as 4 consumed IOPS.</p> <p>Units: Count</p>

Metric	Description
BurstBalance	<p>Used with General Purpose SSD (gp2), Throughput Optimized HDD (st1), and Cold HDD (sc1) volumes only. Provides information about the percentage of I/O credits (for gp2) or throughput credits (for st1 and sc1) remaining in the burst bucket. Data is reported to CloudWatch only when the volume is active. If the volume is not attached, no data is reported.</p> <p>The <code>Sum</code> statistic on this metric is not relevant for volumes attached to Nitro-based instances.</p> <p>If the baseline performance of the volume exceeds the maximum burst performance, credits are never spent. If the volume is attached to a Nitro-based instance, the burst balance is not reported. For a non-Nitro-based instance, the reported burst balance is 100%. For more information, see I/O Credits and burst performance (p. 1026).</p> <p>Units: Percent</p>

Fast snapshot restore metrics

AWS/EBS namespace includes the following metrics for [fast snapshot restore \(p. 1132\)](#).

Metric	Description
FastSnapshotRestoreCreditsBucket	<p>The maximum number of volume create credits that can be accumulated. This metric is reported per snapshot per Availability Zone.</p> <p>The most meaningful statistic is <code>Average</code>. The results for the <code>Minimum</code> and <code>Maximum</code> statistics are the same as for <code>Average</code> and could be used instead.</p>
FastSnapshotRestoreCreditsAvailable	<p>The number of volume create credits available. This metric is reported per snapshot per Availability Zone.</p> <p>The most meaningful statistic is <code>Average</code>. The results for the <code>Minimum</code> and <code>Maximum</code> statistics are the same as for <code>Average</code> and could be used instead.</p>

Dimensions for Amazon EBS metrics

The supported dimension is the volume ID (`VolumeId`). All available statistics are filtered by volume ID.

For the [volume metrics \(p. 1172\)](#), the supported dimension is the volume ID (`VolumeId`). All available statistics are filtered by volume ID.

For the [fast snapshot restore metrics \(p. 1176\)](#), the supported dimensions are the snapshot ID (`SnapshotId`) and the Availability Zone (`AvailabilityZone`).

Graphs in the Amazon EC2 console

After you create a volume, you can view the volume's monitoring graphs in the Amazon EC2 console. Select a volume on the **Volumes** page in the console and choose **Monitoring**. The following table lists

the graphs that are displayed. The column on the right describes how the raw data metrics from the CloudWatch API are used to produce each graph. The period for all the graphs is 5 minutes.

Graph	Description using raw metrics
Read Bandwidth (KiB/s)	$\text{Sum}(\text{VolumeReadBytes}) / \text{Period} / 1024$
Write Bandwidth (KiB/s)	$\text{Sum}(\text{VolumeWriteBytes}) / \text{Period} / 1024$
Read Throughput (IOPS)	$\text{Sum}(\text{VolumeReadOps}) / \text{Period}$
Write Throughput (IOPS)	$\text{Sum}(\text{VolumeWriteOps}) / \text{Period}$
Avg Queue Length (Operations)	$\text{Avg}(\text{VolumeQueueLength})$
% Time Spent Idle	$\text{Sum}(\text{VolumeIdleTime}) / \text{Period} \times 100$
Avg Read Size (KiB/Operation)	<p>$\text{Avg}(\text{VolumeReadBytes}) / 1024$</p> <p>For Nitro-based instances, the following formula derives Average Read Size using CloudWatch Metric Math:</p> $(\text{Sum}(\text{VolumeReadBytes}) / \text{Sum}(\text{VolumeReadOps})) / 1024$ <p>The <code>VolumeReadBytes</code> and <code>VolumeReadOps</code> metrics are available in the EBS CloudWatch console.</p>
Avg Write Size (KiB/Operation)	<p>$\text{Avg}(\text{VolumeWriteBytes}) / 1024$</p> <p>For Nitro-based instances, the following formula derives Average Write Size using CloudWatch Metric Math:</p> $(\text{Sum}(\text{VolumeWriteBytes}) / \text{Sum}(\text{VolumeWriteOps})) / 1024$ <p>The <code>VolumeWriteBytes</code> and <code>VolumeWriteOps</code> metrics are available in the EBS CloudWatch console.</p>
Avg Read Latency (ms/Operation)	<p>$\text{Avg}(\text{VolumeTotalReadTime}) \times 1000$</p> <p>For Nitro-based instances, the following formula derives Average Read Latency using CloudWatch Metric Math:</p> $(\text{Sum}(\text{VolumeTotalReadTime}) / \text{Sum}(\text{VolumeReadOps})) \times 1000$ <p>The <code>VolumeTotalReadTime</code> and <code>VolumeReadOps</code> metrics are available in the EBS CloudWatch console.</p>
Avg Write Latency (ms/Operation)	<p>$\text{Avg}(\text{VolumeTotalWriteTime}) \times 1000$</p> <p>For Nitro-based instances, the following formula derives Average Write Latency using CloudWatch Metric Math:</p> $(\text{Sum}(\text{VolumeTotalWriteTime}) / \text{Sum}(\text{VolumeWriteOps})) \times 1000$ <p>The <code>VolumeTotalWriteTime</code> and <code>VolumeWriteOps</code> metrics are available in the EBS CloudWatch console.</p>

For the average latency graphs and average size graphs, the average is calculated over the total number of operations (read or write, whichever is applicable to the graph) that completed during the period.

Amazon CloudWatch Events for Amazon EBS

Amazon EBS emits notifications based on Amazon CloudWatch Events for a variety of volume, snapshot, and encryption status changes. With CloudWatch Events, you can establish rules that trigger programmatic actions in response to a change in volume, snapshot, or encryption key state. For example, when a snapshot is created, you can trigger an AWS Lambda function to share the completed snapshot with another account or copy it to another Region for disaster-recovery purposes.

Events in CloudWatch are represented as JSON objects. The fields that are unique to the event are contained in the "detail" section of the JSON object. The "event" field contains the event name. The "result" field contains the completed status of the action that triggered the event. For more information, see [Event Patterns in CloudWatch Events](#) in the *Amazon CloudWatch Events User Guide*.

For more information, see [Using Events](#) in the *Amazon CloudWatch User Guide*.

Contents

- [EBS volume events \(p. 1178\)](#)
- [EBS snapshot events \(p. 1181\)](#)
- [EBS volume modification events \(p. 1185\)](#)
- [EBS fast snapshot restore events \(p. 1185\)](#)
- [Using AWS Lambda to handle CloudWatch events \(p. 1186\)](#)

EBS volume events

Amazon EBS sends events to CloudWatch Events when the following volume events occur.

Events

- [Create volume \(createVolume\) \(p. 1178\)](#)
- [Delete volume \(deleteVolume\) \(p. 1179\)](#)
- [Volume attach or reattach \(attachVolume, reattachVolume\) \(p. 1180\)](#)

Create volume (createVolume)

The `createVolume` event is sent to your AWS account when an action to create a volume completes. However it is not saved, logged, or archived. This event can have a result of either `available` or `failed`. Creation will fail if an invalid KMS key was provided, as shown in the examples below.

Event data

The listing below is an example of a JSON object emitted by EBS for a successful `createVolume` event.

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-east-1:012345678901:volume/vol-01234567"  
    ],  
}
```

```
    "detail": {
        "result": "available",
        "cause": "",
        "event": "createVolume",
        "request-id": "01234567-0123-0123-0123-0123456789ab"
    }
}
```

The listing below is an example of a JSON object emitted by EBS after a failed `createVolume` event. The cause for the failure was a disabled KMS key.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "EBS Volume Notification",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "sa-east-1",
    "resources": [
        "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567",
    ],
    "detail": {
        "event": "createVolume",
        "result": "failed",
        "cause": "arn:aws:kms:sa-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab
is disabled.",
        "request-id": "01234567-0123-0123-0123-0123456789ab",
    }
}
```

The following is an example of a JSON object that is emitted by EBS after a failed `createVolume` event. The cause for the failure was a KMS key pending import.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "EBS Volume Notification",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "sa-east-1",
    "resources": [
        "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567",
    ],
    "detail": {
        "event": "createVolume",
        "result": "failed",
        "cause": "arn:aws:kms:sa-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab
is pending import.",
        "request-id": "01234567-0123-0123-0123-0123456789ab",
    }
}
```

Delete volume (`deleteVolume`)

The `deleteVolume` event is sent to your AWS account when an action to delete a volume completes. However it is not saved, logged, or archived. This event has the result `deleted`. If the deletion does not complete, the event is never sent.

Event data

The listing below is an example of a JSON object emitted by EBS for a successful deleteVolume event.

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-east-1:012345678901:volume/vol-01234567"  
    ],  
    "detail": {  
        "result": "deleted",  
        "cause": "",  
        "event": "deleteVolume",  
        "request-id": "01234567-0123-0123-0123-0123456789ab"  
    }  
}
```

Volume attach or reattach (attachVolume, reattachVolume)

The `attachVolume` or `reattachVolume` event is sent to your AWS account if a volume fails to attach or reattach to an instance. However it is not saved, logged, or archived. If you use a KMS key to encrypt an EBS volume and the key becomes invalid, EBS will emit an event if that key is later used to attach or reattach to an instance, as shown in the examples below.

Event data

The listing below is an example of a JSON object emitted by EBS after a failed `attachVolume` event. The cause for the failure was a KMS key pending deletion.

Note

AWS may attempt to reattach to a volume following routine server maintenance.

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-0123456789ab",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-east-1:0123456789ab:volume/vol-01234567",  
        "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab"  
    ],  
    "detail": {  
        "event": "attachVolume",  
        "result": "failed",  
        "cause": "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab  
is pending deletion.",  
        "request-id": ""  
    }  
}
```

The listing below is an example of a JSON object emitted by EBS after a failed `reattachVolume` event. The cause for the failure was a KMS key pending deletion.

```
{
```

```
"version": "0",
"id": "01234567-0123-0123-0123-0123456789ab",
"detail-type": "EBS Volume Notification",
"source": "aws.ec2",
"account": "012345678901",
"time": "yyyy-mm-ddThh:mm:ssZ",
"region": "us-east-1",
"resources": [
    "arn:aws:ec2:us-east-1:0123456789ab:volume/vol-01234567",
    "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab"
],
"detail": {
    "event": "reattachVolume",
    "result": "failed",
    "cause": "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab is pending deletion.",
    "request-id": ""
}
}
```

EBS snapshot events

Amazon EBS sends events to CloudWatch Events when the following volume events occur.

Events

- [Create snapshot \(createSnapshot\) \(p. 1181\)](#)
- [Create snapshots \(createSnapshots\) \(p. 1182\)](#)
- [Copy snapshot \(copySnapshot\) \(p. 1183\)](#)
- [Share snapshot \(shareSnapshot\) \(p. 1184\)](#)

Create snapshot (createSnapshot)

The `createSnapshot` event is sent to your AWS account when an action to create a snapshot completes. However it is not saved, logged, or archived. This event can have a result of either succeeded or failed.

Event data

The listing below is an example of a JSON object emitted by EBS for a successful `createSnapshot` event. In the detail section, the `source` field contains the ARN of the source volume. The `startTime` and `endTime` fields indicate when creation of the snapshot started and completed.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-012345678901",
    "detail-type": "EBS Snapshot Notification",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-west-2::snapshot/snap-01234567"
    ],
    "detail": {
        "event": "createSnapshot",
        "result": "succeeded",
        "cause": "",
        "request-id": "",
        "snapshot_id": "arn:aws:ec2:us-west-2::snapshot/snap-01234567",
        "source": "arn:aws:ec2:us-west-2::volume/vol-01234567",
        "start_time": "2015-03-03T14:45:00Z",
        "end_time": "2015-03-03T14:45:05Z"
    }
}
```

```
        "startTime": "yyyy-mm-ddThh:mm:ssZ",
        "endTime": "yyyy-mm-ddThh:mm:ssZ"  }
    }
```

Create snapshots (createSnapshots)

The `createSnapshots` event is sent to your AWS account when an action to create a multi-volume snapshot completes. This event can have a result of either succeeded or failed.

Event data

The listing below is an example of a JSON object emitted by EBS for a successful `createSnapshots` event. In the detail section, the `source` field contains the ARNs of the source volumes of the multi-volume snapshot set. The `startTime` and `endTime` fields indicate when creation of the snapshot started and completed.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Multi-Volume Snapshots Completion Status",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
    "arn:aws:ec2::us-east-1:snapshot/snap-01234568"
  ],
  "detail": {
    "event": "createSnapshots",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "snapshots": [
      {
        "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
        "source": "arn:aws:ec2::us-east-1:volume/vol-01234567",
        "status": "completed"
      },
      {
        "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-012345678",
        "source": "arn:aws:ec2::us-east-1:volume/vol-012345678",
        "status": "completed"
      }
    ]
  }
}
```

The listing below is an example of a JSON object emitted by EBS after a failed `createSnapshots` event. The cause for the failure was one or more snapshots failed to complete. The values of `snapshot_id` are the ARNs of the failed snapshots. `startTime` and `endTime` represent when the `create-snapshots` action started and ended.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Multi-Volume Snapshots Completion Status",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
```

```

"region": "us-east-1",
"resources": [
    "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
    "arn:aws:ec2::us-east-1:snapshot/snap-01234568"
],
"detail": {
    "event": "createSnapshots",
    "result": "failed",
    "cause": "Snapshot snap-01234567 is in status deleted",
    "request-id": "",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "snapshots": [
        {
            "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
            "source": "arn:aws:ec2::us-east-1:volume/vol-01234567",
            "status": "error"
        },
        {
            "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-012345678",
            "source": "arn:aws:ec2::us-east-1:volume/vol-012345678",
            "status": "deleted"
        }
    ]
}
}

```

Copy snapshot (copySnapshot)

The copySnapshot event is sent to your AWS account when an action to copy a snapshot completes. However it is not saved, logged, or archived. This event can have a result of either succeeded or failed.

Event data

The listing below is an example of a JSON object emitted by EBS after a successful copySnapshot event. The value of snapshot_id is the ARN of the newly created snapshot. In the detail section, the value of source is the ARN of the source snapshot. startTime and endTime represent when the copySnapshot action started and ended.

```

{
    "version": "0",
    "id": "01234567-0123-0123-0123-012345678901",
    "detail-type": "EBS Snapshot Notification",
    "source": "aws.ec2",
    "account": "123456789012",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
    ],
    "detail": {
        "event": "copySnapshot",
        "result": "succeeded",
        "cause": "",
        "request-id": "",
        "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
        "source": "arn:aws:ec2::eu-west-1:snapshot/snap-76543210",
        "startTime": "yyyy-mm-ddThh:mm:ssZ",
        "endTime": "yyyy-mm-ddThh:mm:ssZ",
        "Incremental": "True"
    }
}

```

The listing below is an example of a JSON object emitted by EBS after a failed copySnapshot event. The cause for the failure was an invalid source snapshot ID. The value of `snapshot_id` is the ARN of the failed snapshot. In the detail section, the value of `source` is the ARN of the source snapshot. `startTime` and `endTime` represent when the copy-snapshot action started and ended.

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Snapshot Notification",  
    "source": "aws.ec2",  
    "account": "123456789012",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-west-2::snapshot/snap-01234567"  
    ],  
    "detail": {  
        "event": "copySnapshot",  
        "result": "failed",  
        "cause": "Source snapshot ID is not valid",  
        "request-id": "",  
        "snapshot_id": "arn:aws:ec2:us-west-2::snapshot/snap-01234567",  
        "source": "arn:aws:ec2:eu-west-1::snapshot/snap-76543210",  
        "startTime": "yyyy-mm-ddThh:mm:ssZ",  
        "endTime": "yyyy-mm-ddThh:mm:ssZ"  
    }  
}
```

Share snapshot (shareSnapshot)

The `shareSnapshot` event is sent to your AWS account when another account shares a snapshot with it. However it is not saved, logged, or archived. The result is always succeeded.

Event data

The following is an example of a JSON object emitted by EBS after a completed `shareSnapshot` event. In the detail section, the value of `source` is the AWS account number of the user that shared the snapshot with you. `startTime` and `endTime` represent when the share-snapshot action started and ended. The `shareSnapshot` event is emitted only when a private snapshot is shared with another user. Sharing a public snapshot does not trigger the event.

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Snapshot Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-west-2::snapshot/snap-01234567"  
    ],  
    "detail": {  
        "event": "shareSnapshot",  
        "result": "succeeded",  
        "cause": "",  
        "request-id": "",  
        "snapshot_id": "arn:aws:ec2:us-west-2::snapshot/snap-01234567",  
        "source": "012345678901",  
        "startTime": "yyyy-mm-ddThh:mm:ssZ",  
        "endTime": "yyyy-mm-ddThh:mm:ssZ"  
    }  
}
```

}

EBS volume modification events

Amazon EBS sends modifyVolume events to CloudWatch Events when a volume is modified. However it is not saved, logged, or archived.

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-east-1:012345678901:volume/vol-03a55cf56513fa1b6"  
    ],  
    "detail": {  
        "result": "optimizing",  
        "cause": "",  
        "event": "modifyVolume",  
        "request-id": "01234567-0123-0123-0123-0123456789ab"  
    }  
}
```

EBS fast snapshot restore events

Amazon EBS sends events to CloudWatch Events when the state of fast snapshot restore for a snapshot changes.

The following is example data for this event.

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Fast Snapshot Restore State-change Notification",  
    "source": "aws.ec2",  
    "account": "123456789012",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-east-1::snapshot/snap-03a55cf56513fa1b6"  
    ],  
    "detail": {  
        "snapshot-id": "snap-1234567890abcdef0",  
        "state": "optimizing",  
        "zone": "us-east-1a",  
        "message": "Client.UserInitiated - Lifecycle state transition",  
    }  
}
```

The possible values for state are enabling, optimizing, enabled, disabling, and disabled.

The possible values for message are as follows:

`Client.InvalidSnapshot.InvalidState` – The requested snapshot transitioned to an invalid state (Error)

A request to enable fast snapshot restore failed and the state transitioned to disabling or disabled. Fast snapshot restore cannot be enabled for this snapshot.

`Client.UserInitiated`

The state successfully transitioned to enabling or disabling.

`Client.UserInitiated - Lifecycle state transition`

The state successfully transitioned to optimizing, enabled, or disabled.

`Server.InsufficientCapacity - There was insufficient capacity available to satisfy the request`

A request to enable fast snapshot restore failed due to insufficient capacity, and the state transitioned to disabling or disabled. Wait and then try again.

`Server.InternalError - An internal error caused the operation to fail`

A request to enable fast snapshot restore failed due to an internal error, and the state transitioned to disabling or disabled. Wait and then try again.

`Client.InvalidSnapshot.InvalidState - The requested snapshot was deleted or access permissions were revoked`

The fast snapshot restore state for the snapshot has transitioned to disabling or disabled because the snapshot was deleted or unshared by the snapshot owner. Fast snapshot restore cannot be enabled for a snapshot that has been deleted or is no longer shared with you.

Using AWS Lambda to handle CloudWatch events

You can use Amazon EBS and CloudWatch Events to automate your data-backup workflow. This requires you to create an IAM policy, a AWS Lambda function to handle the event, and an Amazon CloudWatch Events rule that matches incoming events and routes them to the Lambda function.

The following procedure uses the `createSnapshot` event to automatically copy a completed snapshot to another Region for disaster recovery.

To copy a completed snapshot to another Region

1. Create an IAM policy, such as the one shown in the following example, to provide permissions to execute a `CopySnapshot` action and write to the CloudWatch Events log. Assign the policy to the IAM user that will handle the CloudWatch event.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "logs>CreateLogGroup",  
                "logs>CreateLogStream",  
                "logs:PutLogEvents"  
            ],  
            "Resource": "arn:aws:logs:*:*:  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CopySnapshot"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

2. Define a function in Lambda that will be available from the CloudWatch console. The sample Lambda function below, written in Node.js, is invoked by CloudWatch when a matching `createSnapshot` event is emitted by Amazon EBS (signifying that a snapshot was completed). When invoked, the function copies the snapshot from `us-east-2` to `us-east-1`.

```
// Sample Lambda function to copy an EBS snapshot to a different region

var AWS = require('aws-sdk');
var ec2 = new AWS.EC2();

// define variables
var destinationRegion = 'us-east-1';
var sourceRegion = 'us-east-2';
console.log ('Loading function');

//main function
exports.handler = (event, context, callback) => {

    // Get the EBS snapshot ID from the CloudWatch event details
    var snapshotArn = event.detail.snapshot_id.split('/');
    const snapshotId = snapshotArn[1];
    const description = `Snapshot copy from ${snapshotId} in ${sourceRegion}.`;
    console.log ("snapshotId:", snapshotId);

    // Load EC2 class and update the configuration to use destination Region to
    // initiate the snapshot.
    AWS.config.update({region: destinationRegion});
    var ec2 = new AWS.EC2();

    // Prepare variables for ec2.modifySnapshotAttribute call
    const copySnapshotParams = {
        Description: description,
        DestinationRegion: destinationRegion,
        SourceRegion: sourceRegion,
        SourceSnapshotId: snapshotId
    };

    // Execute the copy snapshot and log any errors
    ec2.copySnapshot(copySnapshotParams, (err, data) => {
        if (err) {
            const errorMessage = `Error copying snapshot ${snapshotId} to Region
${destinationRegion}.`;
            console.log(errorMessage);
            console.log(err);
            callback(errorMessage);
        } else {
            const successMessage = `Successfully started copy of snapshot ${snapshotId}
to Region ${destinationRegion}.`;
            console.log(successMessage);
            console.log(data);
            callback(null, successMessage);
        }
    });
};

}
```

To ensure that your Lambda function is available from the CloudWatch console, create it in the Region where the CloudWatch event will occur. For more information, see the [AWS Lambda Developer Guide](#).

3. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
4. Choose **Events**, **Create rule**, **Select event source**, and **Amazon EBS Snapshots**.
5. For **Specific Event(s)**, choose `createSnapshot` and for **Specific Result(s)**, choose `succeeded`.
6. For **Rule target**, find and choose the sample function that you previously created.

7. Choose **Target, Add Target**.
8. For **Lambda function**, select the Lambda function that you previously created and choose **Configure details**.
9. On the **Configure rule details** page, type values for **Name** and **Description**. Select the **State** check box to activate the function (setting it to **Enabled**).
10. Choose **Create rule**.

Your rule should now appear on the **Rules** tab. In the example shown, the event that you configured should be emitted by EBS the next time you copy a snapshot.

Amazon EBS quotas

To view the quotas for your Amazon EBS resources, open the Service Quotas console at <https://console.aws.amazon.com/servicequotas/>. In the navigation pane, choose **AWS services**, and select **Amazon Elastic Block Store (Amazon EBS)**.

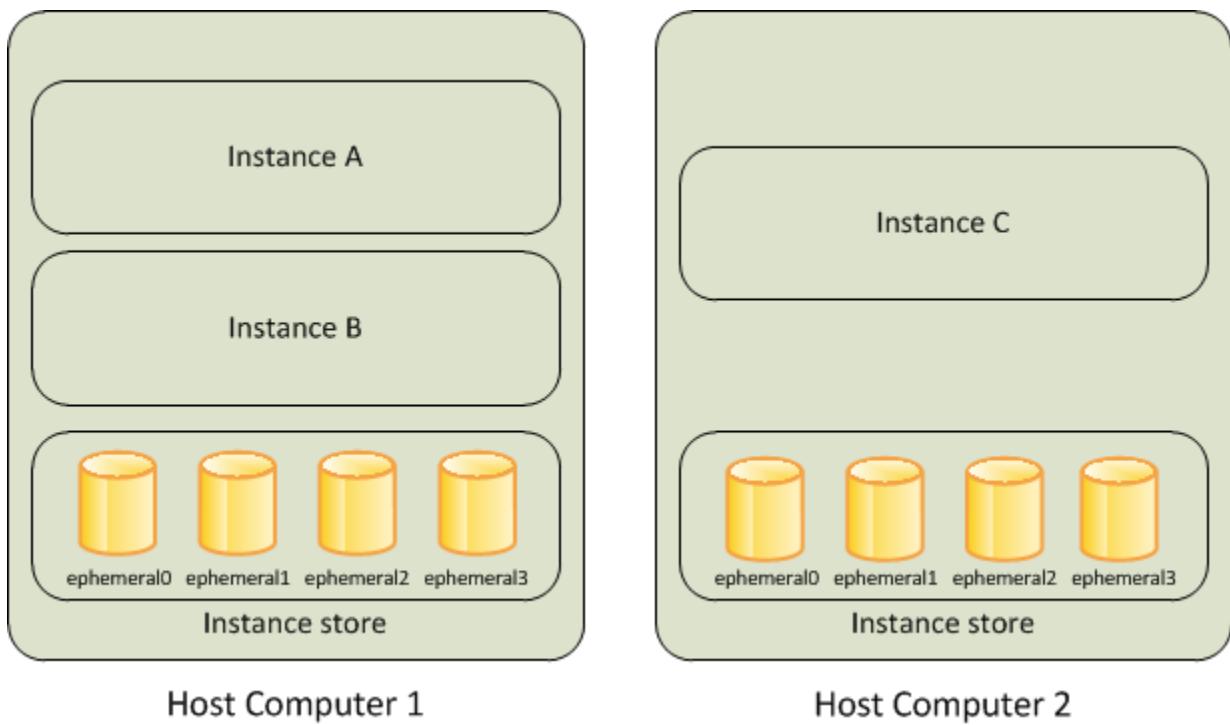
For a list of Amazon EBS service quotas, see [Amazon Elastic Block Store endpoints and quotas](#) in the *AWS General Reference*.

Amazon EC2 instance store

An *instance store* provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.

An instance store consists of one or more instance store volumes exposed as block devices. The size of an instance store as well as the number of devices available varies by instance type.

The virtual devices for instance store volumes are `ephemeral[0–23]`. Instance types that support one instance store volume have `ephemeral0`. Instance types that support two instance store volumes have `ephemeral0` and `ephemeral1`, and so on.



Contents

- [Instance store lifetime \(p. 1189\)](#)
- [Instance store volumes \(p. 1190\)](#)
- [Add instance store volumes to your EC2 instance \(p. 1196\)](#)
- [SSD instance store volumes \(p. 1199\)](#)
- [Instance store swap volumes \(p. 1201\)](#)
- [Optimizing disk performance for instance store volumes \(p. 1203\)](#)

Instance store lifetime

You can specify instance store volumes for an instance only when you launch it. You can't detach an instance store volume from one instance and attach it to a different instance.

The data in an instance store persists only during the lifetime of its associated instance. If an instance reboots (intentionally or unintentionally), data in the instance store persists. However, data in the instance store is lost under any of the following circumstances:

- The underlying disk drive fails
- The instance stops
- The instance terminates

Therefore, do not rely on instance store for valuable, long-term data. Instead, use more durable data storage, such as Amazon S3, Amazon EBS, or Amazon EFS.

When you stop or terminate an instance, every block of storage in the instance store is reset. Therefore, your data cannot be accessed through the instance store of another instance.

If you create an AMI from an instance, the data on its instance store volumes isn't preserved and isn't present on the instance store volumes of the instances that you launch from the AMI.

If you change the instance type, an instance store will not be attached to the new instance type. For more information, see [Changing the instance type \(p. 294\)](#).

Instance store volumes

The instance type determines the size of the instance store available and the type of hardware used for the instance store volumes. Instance store volumes are included as part of the instance's usage cost. You must specify the instance store volumes that you'd like to use when you launch the instance (except for NVMe instance store volumes, which are available by default). Then format and mount the instance store volumes before using them. You can't make an instance store volume available after you launch the instance. For more information, see [Add instance store volumes to your EC2 instance \(p. 1196\)](#).

Some instance types use NVMe or SATA-based solid state drives (SSD) to deliver high random I/O performance. This is a good option when you need storage with very low latency, but you don't need the data to persist when the instance terminates or you can take advantage of fault-tolerant architectures. For more information, see [SSD instance store volumes \(p. 1199\)](#).

The following table provides the quantity, size, type, and performance optimizations of instance store volumes available on each supported instance type. For a complete list of instance types, including EBS-only types, see [Amazon EC2 Instance Types](#).

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
c1.medium	1 x 350 GB†	HDD	✓	
c1.xlarge	4 x 420 GB (1.6 TB)	HDD	✓	
c3.large	2 x 16 GB (32 GB)	SSD	✓	
c3.xlarge	2 x 40 GB (80 GB)	SSD	✓	
c3.2xlarge	2 x 80 GB (160 GB)	SSD	✓	
c3.4xlarge	2 x 160 GB (320 GB)	SSD	✓	
c3.8xlarge	2 x 320 GB (640 GB)	SSD	✓	
c5ad.large	1 x 75 GB	NVMe SSD		✓
c5ad.xlarge	1 x 150 GB	NVMe SSD		✓
c5ad.2xlarge	1 x 300 GB	NVMe SSD		✓
c5ad.4xlarge	2 x 300 GB (600 GB)	NVMe SSD		✓
c5ad.8xlarge	2 x 600 GB (1.2 TB)	NVMe SSD		✓
c5ad.12xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓
c5ad.16xlarge	2 x 1.2 TB (2.4 TB)	NVMe SSD		✓
c5ad.24xlarge	2 x 1.9 TB (3.8 TB)	NVMe SSD		✓
c5d.large	1 x 50 GB	NVMe SSD		✓
c5d.xlarge	1 x 100 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
c5d.2xlarge	1 x 200 GB	NVMe SSD		✓
c5d.4xlarge	1 x 400 GB	NVMe SSD		✓
c5d.9xlarge	1 x 900 GB	NVMe SSD		✓
c5d.12xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓
c5d.18xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓
c5d.24xlarge	4 x 900 GB (3.6 TB)	NVMe SSD		✓
c5d.metal	4 x 900 GB (3.6 TB)	NVMe SSD		✓
c6gd.medium	1 x 59 GB	NVMe SSD		✓
c6gd.large	1 x 118 GB	NVMe SSD		✓
c6gd.xlarge	1 x 237 GB	NVMe SSD		✓
c6gd.2xlarge	1 x 474 GB	NVMe SSD		✓
c6gd.4xlarge	1 x 950 GB	NVMe SSD		✓
c6gd.8xlarge	1 x 1900 GB	NVMe SSD		✓
c6gd.12xlarge	2 x 1425 GB (2.85 TB)	NVMe SSD		✓
c6gd.16xlarge	2 x 1900 GB (3.8 TB)	NVMe SSD		✓
c6gd.metal	2 x 1900 GB (3.8 TB)	NVMe SSD		✓
cc2.8xlarge	4 x 840 GB (3.36 TB)	HDD	✓	
cr1.8xlarge	2 x 120 GB (240 GB)	SSD	✓	
d2.xlarge	3 x 2,000 GB (6 TB)	HDD		
d2.2xlarge	6 x 2,000 GB (12 TB)	HDD		
d2.4xlarge	12 x 2,000 GB (24 TB)	HDD		
d2.8xlarge	24 x 2,000 GB (48 TB)	HDD		
f1.2xlarge	1 x 470 GB	NVMe SSD		✓
f1.4xlarge	1 x 940 GB	NVMe SSD		✓
f1.16xlarge	4 x 940 GB (3.76 TB)	NVMe SSD		✓
g2.2xlarge	1 x 60 GB	SSD	✓	
g2.8xlarge	2 x 120 GB (240 GB)	SSD	✓	
g4dn.xlarge	1 x 125 GB	NVMe SSD		✓
g4dn.2xlarge	1 x 225 GB	NVMe SSD		✓
g4dn.4xlarge	1 x 225 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
g4dn.8xlarge	1 x 900 GB	NVMe SSD		✓
g4dn.12xlarge	1 x 900 GB	NVMe SSD		✓
g4dn.16xlarge	1 x 900 GB	NVMe SSD		✓
g4dn.metal	2 x 900 GB (1.8 TB)	NVMe SSD		✓
h1.2xlarge	1 x 2000 GB (2 TB)	HDD		
h1.4xlarge	2 x 2000 GB (4 TB)	HDD		
h1.8xlarge	4 x 2000 GB (8 TB)	HDD		
h1.16xlarge	8 x 2000 GB (16 TB)	HDD		
hs1.8xlarge	24 x 2,000 GB (48 TB)	HDD	✓	
i2.xlarge	1 x 800 GB	SSD		✓
i2.2xlarge	2 x 800 GB (1.6 TB)	SSD		✓
i2.4xlarge	4 x 800 GB (3.2 TB)	SSD		✓
i2.8xlarge	8 x 800 GB (6.4 TB)	SSD		✓
i3.large	1 x 475 GB	NVMe SSD		✓
i3.xlarge	1 x 950 GB	NVMe SSD		✓
i3.2xlarge	1 x 1,900 GB	NVMe SSD		✓
i3.4xlarge	2 x 1,900 GB (3.8 TB)	NVMe SSD		✓
i3.8xlarge	4 x 1,900 GB (7.6 TB)	NVMe SSD		✓
i3.16xlarge	8 x 1,900 GB (15.2 TB)	NVMe SSD		✓
i3.metal	8 x 1,900 GB (15.2 TB)	NVMe SSD		✓
i3en.large	1 x 1,250 GB	NVMe SSD		✓
i3en.xlarge	1 x 2,500 GB	NVMe SSD		✓
i3en.2xlarge	2 x 2,500 GB (5 TB)	NVMe SSD		✓
i3en.3xlarge	1 x 7,500 GB	NVMe SSD		✓
i3en.6xlarge	2 x 7,500 GB (15 TB)	NVMe SSD		✓
i3en.12xlarge	4 x 7,500 GB (30 TB)	NVMe SSD		✓
i3en.24xlarge	8 x 7,500 GB (60 TB)	NVMe SSD		✓
i3en.metal	8 x 7,500 GB (60 TB)	NVMe SSD		✓
m1.small	1 x 160 GB†	HDD	✓	
m1.medium	1 x 410 GB	HDD	✓	

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
m1.large	2 x 420 GB (840 GB)	HDD	✓	
m1.xlarge	4 x 420 GB (1.6 TB)	HDD	✓	
m2.xlarge	1 x 420 GB	HDD	✓	
m2.2xlarge	1 x 850 GB	HDD	✓	
m2.4xlarge	2 x 840 GB (1.68 TB)	HDD	✓	
m3.medium	1 x 4 GB	SSD	✓	
m3.large	1 x 32 GB	SSD	✓	
m3.xlarge	2 x 40 GB (80 GB)	SSD	✓	
m3.2xlarge	2 x 80 GB (160 GB)	SSD	✓	
m5ad.large	1 x 75 GB	NVMe SSD		✓
m5ad.xlarge	1 x 150 GB	NVMe SSD		✓
m5ad.2xlarge	1 x 300 GB	NVMe SSD		✓
m5ad.4xlarge	2 x 300 GB (600 GB)	NVMe SSD		✓
m5ad.8xlarge	2 x 600 GB (1.2 TB)	NVMe SSD		✓
m5ad.12xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓
m5ad.16xlarge	4 x 600 GB (2.4 TB)	NVMe SSD		✓
m5ad.24xlarge	4 x 900 GB (3.6 TB)	NVMe SSD		✓
m5d.large	1 x 75 GB	NVMe SSD		✓
m5d.xlarge	1 x 150 GB	NVMe SSD		✓
m5d.2xlarge	1 x 300 GB	NVMe SSD		✓
m5d.4xlarge	2 x 300 GB (600 GB)	NVMe SSD		✓
m5d.8xlarge	2 x 600 GB (1.2 TB)	NVMe SSD		✓
m5d.12xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓
m5d.16xlarge	4 x 600 GB (2.4 TB)	NVMe SSD		✓
m5d.24xlarge	4 x 900 GB (3.6 TB)	NVMe SSD		✓
m5d.metal	4 x 900 GB (3.6 TB)	NVMe SSD		✓
m5dn.large	1 x 75 GB	NVMe SSD		✓
m5dn.xlarge	1 x 150 GB	NVMe SSD		✓
m5dn.2xlarge	1 x 300 GB	NVMe SSD		✓
m5dn.4xlarge	2 x 300 GB (600 GB)	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
m5dn.8xlarge	2 x 600 GB (1.2 TB)	NVMe SSD		✓
m5dn.12xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓
m5dn.16xlarge	4 x 600 GB (2.4 TB)	NVMe SSD		✓
m5dn.24xlarge	4 x 900 GB (3.6 TB)	NVMe SSD		✓
m6gd.medium	1 x 59 GB	NVMe SSD		✓
m6gd.large	1 x 118 GB	NVMe SSD		✓
m6gd.xlarge	1 x 237 GB	NVMe SSD		✓
m6gd.2xlarge	1 x 474 GB	NVMe SSD		✓
m6gd.4xlarge	1 x 950 GB	NVMe SSD		✓
m6gd.8xlarge	1 x 1900 GB	NVMe SSD		✓
m6gd.12xlarge	2 x 1425 GB (2.85 TB)	NVMe SSD		✓
m6gd.16xlarge	2 x 1900 GB (3.8 TB)	NVMe SSD		✓
m6gd.metal	2 x 1900 GB (3.8 TB)	NVMe SSD		✓
p3dn.24xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓
r3.large	1 x 32 GB	SSD		✓
r3.xlarge	1 x 80 GB	SSD		✓
r3.2xlarge	1 x 160 GB	SSD		✓
r3.4xlarge	1 x 320 GB	SSD		✓
r3.8xlarge	2 x 320 GB (640 GB)	SSD		✓
r5ad.large	1 x 75 GB	NVMe SSD		✓
r5ad.xlarge	1 x 150 GB	NVMe SSD		✓
r5ad.2xlarge	1 x 300 GB	NVMe SSD		✓
r5ad.4xlarge	2 x 300 GB (600 GB)	NVMe SSD		✓
r5ad.8xlarge	2 x 600 GB (1.2 TB)	NVMe SSD		✓
r5ad.12xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓
r5ad.16xlarge	4 x 600 GB (2.4 TB)	NVMe SSD		✓
r5ad.24xlarge	4 x 900 GB (3.6 TB)	NVMe SSD		✓
r5d.large	1 x 75 GB	NVMe SSD		✓
r5d.xlarge	1 x 150 GB	NVMe SSD		✓
r5d.2xlarge	1 x 300 GB	NVMe SSD		✓

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
r5d.4xlarge	2 x 300 GB (600 GB)	NVMe SSD		✓
r5d.8xlarge	2 x 600 GB (1.2 TB)	NVMe SSD		✓
r5d.12xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓
r5d.16xlarge	4 x 600 GB (2.4 TB)	NVMe SSD		✓
r5d.24xlarge	4 x 900 GB (3.6 TB)	NVMe SSD		✓
r5d.metal	4 x 900 GB (3.6 TB)	NVMe SSD		✓
r5dn.large	1 x 75 GB	NVMe SSD		✓
r5dn.xlarge	1 x 150 GB	NVMe SSD		✓
r5dn.2xlarge	1 x 300 GB	NVMe SSD		✓
r5dn.4xlarge	2 x 300 GB (600 GB)	NVMe SSD		✓
r5dn.8xlarge	2 x 600 GB (1.2 TB)	NVMe SSD		✓
r5dn.12xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓
r5dn.16xlarge	4 x 600 GB (2.4 TB)	NVMe SSD		✓
r5dn.24xlarge	4 x 900 GB (3.6 TB)	NVMe SSD		✓
r6gd.medium	1 x 59 GB	NVMe SSD		✓
r6gd.large	1 x 118 GB	NVMe SSD		✓
r6gd.xlarge	1 x 237 GB	NVMe SSD		✓
r6gd.2xlarge	1 x 474 GB	NVMe SSD		✓
r6gd.4xlarge	1 x 950 GB	NVMe SSD		✓
r6gd.8xlarge	1 x 1900 GB	NVMe SSD		✓
r6gd.12xlarge	2 x 1425 GB (2.85 TB)	NVMe SSD		✓
r6gd.16xlarge	2 x 1900 GB (3.8 TB)	NVMe SSD		✓
r6gd.metal	2 x 1900 GB (3.8 TB)	NVMe SSD		✓
x1.16xlarge	1 x 1,920 GB	SSD		
x1.32xlarge	2 x 1,920 GB (3.84 TB)	SSD		
x1e.xlarge	1 x 120 GB	SSD		
x1e.2xlarge	1 x 240 GB	SSD		
x1e.4xlarge	1 x 480 GB	SSD		
x1e.8xlarge	1 x 960 GB	SSD		
x1e.16xlarge	1 x 1,920 GB	SSD		

Instance type	Instance store volumes	Type	Needs initialization*	TRIM support**
x1e.32xlarge	2 x 1,920 GB (3.84 TB)	SSD		
z1d.large	1 x 75 GB	NVMe SSD		✓
z1d.xlarge	1 x 150 GB	NVMe SSD		✓
z1d.2xlarge	1 x 300 GB	NVMe SSD		✓
z1d.3xlarge	1 x 450 GB	NVMe SSD		✓
z1d.6xlarge	1 x 900 GB	NVMe SSD		✓
z1d.12xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓
z1d.metal	2 x 900 GB (1.8 TB)	NVMe SSD		✓

* Volumes attached to certain instances suffer a first-write penalty unless initialized. For more information, see [Optimizing disk performance for instance store volumes \(p. 1203\)](#).

** For more information, see [Instance store volume TRIM support \(p. 1201\)](#).

† The `c1.medium` and `m1.small` instance types also include a 900 MB instance store swap volume, which may not be automatically enabled at boot time. For more information, see [Instance store swap volumes \(p. 1201\)](#).

Add instance store volumes to your EC2 instance

You specify the EBS volumes and instance store volumes for your instance using a block device mapping. Each entry in a block device mapping includes a device name and the volume that it maps to. The default block device mapping is specified by the AMI you use. Alternatively, you can specify a block device mapping for the instance when you launch it.

All the NVMe instance store volumes supported by an instance type are automatically enumerated and assigned a device name on instance launch; including them in the block device mapping for the AMI or the instance has no effect. For more information, see [Block device mapping \(p. 1211\)](#).

A block device mapping always specifies the root volume for the instance. The root volume is either an Amazon EBS volume or an instance store volume. For more information, see [Storage for the root device \(p. 100\)](#). The root volume is mounted automatically. For instances with an instance store volume for the root volume, the size of this volume varies by AMI, but the maximum size is 10 GB.

You can use a block device mapping to specify additional EBS volumes when you launch your instance, or you can attach additional EBS volumes after your instance is running. For more information, see [Amazon EBS volumes \(p. 1021\)](#).

You can specify the instance store volumes for your instance only when you launch it. You can't attach instance store volumes to an instance after you've launched it.

If you change the instance type, an instance store will not be attached to the new instance type. For more information, see [Changing the instance type \(p. 294\)](#).

The number and size of available instance store volumes for your instance varies by instance type. Some instance types do not support instance store volumes. If the number of instance store volumes in a block device mapping exceeds the number of instance store volumes available to an instance, the additional

volumes are ignored. For more information about the instance store volumes supported by each instance type, see [Instance store volumes \(p. 1190\)](#).

If the instance type you choose for your instance supports non-NVMe instance store volumes, you must add them to the block device mapping for the instance when you launch it. NVMe instance store volumes are available by default. After you launch an instance, you must ensure that the instance store volumes for your instance are formatted and mounted before you can use them. The root volume of an instance store-backed instance is mounted automatically.

Contents

- [Adding instance store volumes to an AMI \(p. 1197\)](#)
- [Adding instance store volumes to an instance \(p. 1198\)](#)
- [Making instance store volumes available on your instance \(p. 1198\)](#)

Adding instance store volumes to an AMI

You can create an AMI with a block device mapping that includes instance store volumes. If you launch an instance with an instance type that supports instance store volumes and an AMI that specifies instance store volumes in its block device mapping, the instance includes these instance store volumes. If the number of instance store volumes in the block device mapping exceeds the number of instance store volumes available to the instance, the additional instance store volumes are ignored.

Considerations

- For M3 instances, specify instance store volumes in the block device mapping of the instance, not the AMI. Amazon EC2 might ignore instance store volumes that are specified only in the block device mapping of the AMI.
- When you launch an instance, you can omit non-NVMe instance store volumes specified in the AMI block device mapping or add instance store volumes.

To add instance store volumes to an Amazon EBS-backed AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select the instance.
3. Choose **Actions, Image, Create Image**.
4. In the **Create Image** dialog box, type a meaningful name and description for your image.
5. For each instance store volume to add, choose **Add New Volume**, from **Volume Type** select an instance store volume, and from **Device** select a device name. (For more information, see [Device naming on Linux instances \(p. 1210\)](#).) The number of available instance store volumes depends on the instance type. For instances with NVMe instance store volumes, the device mapping of these volumes depends on the order in which the operating system enumerates the volumes.
6. Choose **Create Image**.

To add instance store volumes to an AMI using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [create-image](#) or [register-image](#) (AWS CLI)
- [New-EC2Image](#) and [Register-EC2Image](#) (AWS Tools for Windows PowerShell)

Adding instance store volumes to an instance

When you launch an instance, the default block device mapping is provided by the specified AMI. If you need additional instance store volumes, you must add them to the instance as you launch it. You can also omit devices specified in the AMI block device mapping.

Considerations

- For M3 instances, you might receive instance store volumes even if you do not specify them in the block device mapping for the instance.
- For HS1 instances, no matter how many instance store volumes you specify in the block device mapping of an AMI, the block device mapping for an instance launched from the AMI automatically includes the maximum number of supported instance store volumes. You must explicitly remove the instance store volumes that you don't want from the block device mapping for the instance before you launch it.

To update the block device mapping for an instance using the console

1. Open the Amazon EC2 console.
2. From the dashboard, choose **Launch instance**.
3. In **Step 1: Choose an Amazon Machine Image (AMI)**, select the AMI to use and choose **Select**.
4. Follow the wizard to complete **Step 1: Choose an Amazon Machine Image (AMI)**, **Step 2: Choose an Instance Type**, and **Step 3: Configure Instance Details**.
5. In **Step 4: Add Storage**, modify the existing entries as needed. For each instance store volume to add, choose **Add New Volume**, from **Volume Type** select an instance store volume, and from **Device** select a device name. The number of available instance store volumes depends on the instance type.
6. Complete the wizard and launch the instance.
7. (Optional) To view the instance store volumes available on your instance, run the **lsblk** command.

To update the block device mapping for an instance using the command line

You can use one of the following options commands with the corresponding command. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `--block-device-mappings` with [run-instances](#) (AWS CLI)
- `-BlockDeviceMapping` with [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

Making instance store volumes available on your instance

After you launch an instance, the instance store volumes are available to the instance, but you can't access them until they are mounted. For Linux instances, the instance type determines which instance store volumes are mounted for you and which are available for you to mount yourself. For Windows instances, the EC2Config service mounts the instance store volumes for an instance. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different than the name that Amazon EC2 recommends.

Many instance store volumes are pre-formatted with the ext3 file system. SSD-based instance store volumes that support TRIM instruction are not pre-formatted with any file system. However, you can format volumes with the file system of your choice after you launch your instance. For more information, see [Instance store volume TRIM support \(p. 1201\)](#). For Windows instances, the EC2Config service reformats the instance store volumes with the NTFS file system.

You can confirm that the instance store devices are available from within the instance itself using instance metadata. For more information, see [Viewing the instance block device mapping for instance store volumes \(p. 1220\)](#).

For Windows instances, you can also view the instance store volumes using Windows Disk Management. For more information, see [Listing disks using Windows Disk Management](#).

For Linux instances, you can view and mount the instance store volumes as described in the following procedure.

To make an instance store volume available on Linux

1. Connect to the instance using an SSH client. For more information, see [Connect to your Linux instance \(p. 563\)](#).
2. Use the `df -h` command to view the volumes that are formatted and mounted.

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.8G   72K  3.8G   1% /dev
tmpfs          3.8G     0  3.8G   0% /dev/shm
/dev/nvme0n1p1  7.9G  1.2G  6.6G  15% /
```

3. Use the `lsblk` to view any volumes that were mapped at launch but not formatted and mounted.

```
[ec2-user ~]$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1    259:1    0    8G  0 disk
##nvme0n1p1 259:2    0    8G  0 part /
##nvme0n1p128 259:3   0    1M  0 part
nvme1n1    259:0    0  69.9G 0 disk
```

4. To format and mount an instance store volume that was mapped only, do the following:

- a. Create a file system on the device using the `mkfs` command.

```
[ec2-user ~]$ sudo mkfs -t xfs /dev/nvme1n1
```

- b. Create a directory on which to mount the device using the `mkdir` command.

```
[ec2-user ~]$ sudo mkdir /data
```

- c. Mount the device on the newly created directory using the `mount` command.

```
[ec2-user ~]$ sudo mount /dev/nvme1n1 /data
```

For instructions on how to mount an attached volume automatically after reboot, see [Automatically mount an attached volume after reboot \(p. 1048\)](#).

SSD instance store volumes

To ensure the best IOPS performance from your SSD instance store volumes on Linux, we recommend that you use the most recent version of Amazon Linux, or another Linux AMI with a kernel version of 3.8 or later. If you do not use a Linux AMI with a kernel version of 3.8 or later, your instance won't achieve the maximum IOPS performance available for these instance types.

Like other instance store volumes, you must map the SSD instance store volumes for your instance when you launch it. The data on an SSD instance volume persists only for the life of its associated instance. For more information, see [Add instance store volumes to your EC2 instance \(p. 1196\)](#).

NVMe SSD volumes

Some instances offer non-volatile memory express (NVMe) solid state drives (SSD) instance store volumes. For more information about the type of instance store volume supported by each instance type, see [Instance store volumes \(p. 1190\)](#).

To access NVMe volumes, the [NVMe drivers \(p. 1136\)](#) must be installed. The following AMIs meet this requirement:

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (with `linux-aws` kernel) or later
- Red Hat Enterprise Linux 7.4 or later
- SUSE Linux Enterprise Server 12 SP2 or later
- CentOS 7.4.1708 or later
- FreeBSD 11.1 or later
- Debian GNU/Linux 9 or later

After you connect to your instance, you can list the NVMe devices using the `lspci` command. The following is example output for an `i3.8xlarge` instance, which supports four NVMe devices.

```
[ec2-user ~]$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 01)
00:02.0 VGA compatible controller: Cirrus Logic GD 5446
00:03.0 Ethernet controller: Device 1d0f:ec20
00:17.0 Non-Volatile memory controller: Device 1d0f:cd01
00:18.0 Non-Volatile memory controller: Device 1d0f:cd01
00:19.0 Non-Volatile memory controller: Device 1d0f:cd01
00:1a.0 Non-Volatile memory controller: Device 1d0f:cd01
00:1f.0 Unassigned class [ff80]: XenSource, Inc. Xen Platform Device (rev 01)
```

If you are using a supported operating system but you do not see the NVMe devices, verify that the NVMe module is loaded using the following command.

- Amazon Linux, Amazon Linux 2, Ubuntu 14/16, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, CentOS 7

```
$ lsmod | grep nvme
nvme           48813   0
```

- Ubuntu 18

```
$ cat /lib/modules/$(uname -r)/modules.builtin | grep nvme
s/nvme/host/nvme-core.ko
kernel/drivers/nvme/host/nvme.ko
kernel/drivers/nvmmem/nvmmem_core.ko
```

The NVMe volumes are compliant with the NVMe 1.0e specification. You can use the NVMe commands with your NVMe volumes. With Amazon Linux, you can install the `nvme-cli` package from the repo using the `yum install` command. With other supported versions of Linux, you can download the `nvme-cli` package if it's not available in the image.

The data on NVMe instance storage is encrypted using an XTS-AES-256 block cipher implemented in a hardware module on the instance. The encryption keys are generated using the hardware module and are unique to each NVMe instance storage device. All encryption keys are destroyed when the instance is stopped or terminated and cannot be recovered. You cannot disable this encryption and you cannot provide your own encryption key.

Non-NVMe SSD volumes

The following instances support instance store volumes that use non-NVMe SSDs to deliver high random I/O performance: C3, G2, I2, M3, R3, and X1. For more information about the instance store volumes supported by each instance type, see [Instance store volumes \(p. 1190\)](#).

Instance store volume TRIM support

Some instance types support SSD volumes with TRIM. For more information, see [Instance store volumes \(p. 1190\)](#).

Instance store volumes that support TRIM are fully trimmed before they are allocated to your instance. These volumes are not formatted with a file system when an instance launches, so you must format them before they can be mounted and used. For faster access to these volumes, you should skip the TRIM operation when you format them.

With instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller when you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. On Linux, use the `fstrim` command to enable periodic TRIM.

Instance store swap volumes

Swap space in Linux can be used when a system requires more memory than it has been physically allocated. When swap space is enabled, Linux systems can swap infrequently used memory pages from physical memory to swap space (either a dedicated partition or a swap file in an existing file system) and free up that space for memory pages that require high-speed access.

Note

Using swap space for memory paging is not as fast or efficient as using RAM. If your workload is regularly paging memory into swap space, you should consider migrating to a larger instance type with more RAM. For more information, see [Changing the instance type \(p. 294\)](#).

The `c1.medium` and `m1.small` instance types have a limited amount of physical memory to work with, and they are given a 900 MiB swap volume at launch time to act as virtual memory for Linux AMIs. Although the Linux kernel sees this swap space as a partition on the root device, it is actually a separate instance store volume, regardless of your root device type.

Amazon Linux automatically enables and uses this swap space, but your AMI may require some additional steps to recognize and use this swap space. To see if your instance is using swap space, you can use the `swapon -s` command.

```
[ec2-user ~]$ swapon -s
Filename                                Type      Size    Used    Priority
/dev/xvda3                               partition 917500   0       -1
```

The above instance has a 900 MiB swap volume attached and enabled. If you don't see a swap volume listed with this command, you may need to enable swap space for the device. Check your available disks using the `lsblk` command.

```
[ec2-user ~]$ lsblk
```

```
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1    0   8G  0 disk /
xvda3 202:3    0  896M 0 disk
```

Here, the swap volume `xvda3` is available to the instance, but it is not enabled (notice that the `MOUNTPOINT` field is empty). You can enable the swap volume with the `swapon` command.

Note

You must prepend `/dev/` to the device name listed by `lsblk`. Your device may be named differently, such as `sda3`, `sde3`, or `xvde3`. Use the device name for your system in the command below.

```
[ec2-user ~]$ sudo swapon /dev/xvda3
```

Now the swap space should show up in `lsblk` and `swapon -s` output.

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1    0   8G  0 disk /
xvda3 202:3    0  896M 0 disk [SWAP]
[ec2-user ~]$ swapon -s
Filename                                Type      Size  Used  Priority
/dev/xvda3                               partition 917500   0     -1
```

You also need to edit your `/etc/fstab` file so that this swap space is automatically enabled at every system boot.

```
[ec2-user ~]$ sudo vim /etc/fstab
```

Append the following line to your `/etc/fstab` file (using the swap device name for your system):

```
/dev/xvda3      none      swap      sw      0      0
```

To use an instance store volume as swap space

Any instance store volume can be used as swap space. For example, the `m3.medium` instance type includes a 4 GB SSD instance store volume that is appropriate for swap space. If your instance store volume is much larger (for example, 350 GB), you may consider partitioning the volume with a smaller swap partition of 4-8 GB and the rest for a data volume.

Note

This procedure applies only to instance types that support instance storage. For a list of supported instance types, see [Instance store volumes \(p. 1190\)](#).

1. List the block devices attached to your instance to get the device name for your instance store volume.

```
[ec2-user ~]$ lsblk -p
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
/dev/xvdb  202:16   0   4G  0 disk /media/ephemeral0
/dev/xvda1 202:1    0   8G  0 disk /
```

In this example, the instance store volume is `/dev/xvdb`. Because this is an Amazon Linux instance, the instance store volume is formatted and mounted at `/media/ephemeral0`; not all Linux operating systems do this automatically.

2. (Optional) If your instance store volume is mounted (it lists a `MOUNTPOINT` in the `lsblk` command output), unmount it with the following command.

```
[ec2-user ~]$ sudo umount /dev/xvdb
```

3. Set up a Linux swap area on the device with the **mkswap** command.

```
[ec2-user ~]$ sudo mkswap /dev/xvdb
mkswap: /dev/xvdb: warning: wiping old ext3 signature.
Setting up swapspace version 1, size = 4188668 KiB
no label, UUID=b4f63d28-67ed-46f0-b5e5-6928319e620b
```

4. Enable the new swap space.

```
[ec2-user ~]$ sudo swapon /dev/xvdb
```

5. Verify that the new swap space is being used.

```
[ec2-user ~]$ swapon -s
Filename      Type  Size Used Priority
/dev/xvdb            partition 4188668 0 -1
```

6. Edit your **/etc/fstab** file so that this swap space is automatically enabled at every system boot.

```
[ec2-user ~]$ sudo vim /etc/fstab
```

If your **/etc/fstab** file has an entry for **/dev/xvdb** (or **/dev/sdb**) change it to match the line below; if it does not have an entry for this device, append the following line to your **/etc/fstab** file (using the swap device name for your system):

```
/dev/xvdb      none      swap      sw  0      0
```

Important

Instance store volume data is lost when an instance is stopped; this includes the instance store swap space formatting created in [Step 3 \(p. 1203\)](#). If you stop and restart an instance that has been configured to use instance store swap space, you must repeat [Step 1 \(p. 1202\)](#) through [Step 5 \(p. 1203\)](#) on the new instance store volume.

Optimizing disk performance for instance store volumes

Because of the way that Amazon EC2 virtualizes disks, the first write to any location on some instance store volumes performs more slowly than subsequent writes. For most applications, amortizing this cost over the lifetime of the instance is acceptable. However, if you require high disk performance, we recommend that you initialize your drives by writing once to every drive location before production use.

Note

Some instance types with direct-attached solid state drives (SSD) and TRIM support provide maximum performance at launch time, without initialization. For information about the instance store for each instance type, see [Instance store volumes \(p. 1190\)](#).

If you require greater flexibility in latency or throughput, we recommend using Amazon EBS.

To initialize the instance store volumes, use the following **dd** commands, depending on the store to initialize (for example, **/dev/sdb** or **/dev/nvme1n1**).

Note

Make sure to unmount the drive before performing this command.

Initialization can take a long time (about 8 hours for an extra large instance).

To initialize the instance store volumes, use the following commands on the `m1.large`, `m1.xlarge`, `c1.xlarge`, `m2.xlarge`, `m2.2xlarge`, and `m2.4xlarge` instance types:

```
dd if=/dev/zero of=/dev/sdb bs=1M
dd if=/dev/zero of=/dev/sdc bs=1M
dd if=/dev/zero of=/dev/sdd bs=1M
dd if=/dev/zero of=/dev/sde bs=1M
```

To perform initialization on all instance store volumes at the same time, use the following command:

```
dd if=/dev/zero bs=1M|tee /dev/sdb|tee /dev/sdc|tee /dev/sde > /dev/sdd
```

Configuring drives for RAID initializes them by writing to every drive location. When configuring software-based RAID, make sure to change the minimum reconstruction speed:

```
echo $((30*1024)) > /proc/sys/dev/raid/speed_limit_min
```

File storage

Cloud file storage is a method for storing data in the cloud that provides servers and applications access to data through shared file systems. This compatibility makes cloud file storage ideal for workloads that rely on shared file systems and provides simple integration without code changes.

There are many file storage solutions that exist, ranging from a single node file server on a compute instance using block storage as the underpinnings with no scalability or few redundancies to protect the data, to a do-it-yourself clustered solution, to a fully-managed solution. The following content introduces some of the storage services provided by AWS for use with Linux.

Contents

- [Using Amazon S3 with Amazon EC2 \(p. 1204\)](#)
- [Using Amazon EFS with Amazon EC2 \(p. 1206\)](#)

Using Amazon S3 with Amazon EC2

Amazon S3 is a repository for internet data. Amazon S3 provides access to reliable, fast, and inexpensive data storage infrastructure. It is designed to make web-scale computing easier by enabling you to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web. Amazon S3 stores data objects redundantly on multiple devices across multiple facilities and allows concurrent read or write access to these data objects by many separate clients or application threads. You can use the redundant data stored in Amazon S3 to recover quickly and reliably from instance or application failures.

Amazon EC2 uses Amazon S3 for storing Amazon Machine Images (AMIs). You use AMIs for launching EC2 instances. In case of instance failure, you can use the stored AMI to immediately launch another instance, thereby allowing for fast recovery and business continuity.

Amazon EC2 also uses Amazon S3 to store snapshots (backup copies) of the data volumes. You can use snapshots for recovering data quickly and reliably in case of application or system failures. You can also use snapshots as a baseline to create multiple new data volumes, expand the size of an existing data volume, or move data volumes across multiple Availability Zones, thereby making your data usage highly scalable. For more information about using data volumes and snapshots, see [Amazon Elastic Block Store \(p. 1019\)](#).

Objects are the fundamental entities stored in Amazon S3. Every object stored in Amazon S3 is contained in a bucket. Buckets organize the Amazon S3 namespace at the highest level and identify the account responsible for that storage. Amazon S3 buckets are similar to internet domain names. Objects stored in the buckets have a unique key value and are retrieved using a URL. For example, if an object with a key value /photos/mygarden.jpg is stored in the [DOC-EXAMPLE-BUCKET1](#) bucket, then it is addressable using the URL <https://DOC-EXAMPLE-BUCKET1.s3.amazonaws.com/photos/mygarden.jpg>.

For more information about the features of Amazon S3, see the [Amazon S3 product page](#).

Usage examples

Given the benefits of Amazon S3 for storage, you might decide to use this service to store files and data sets for use with EC2 instances. There are several ways to move data to and from Amazon S3 to your instances. In addition to the examples discussed below, there are a variety of tools that people have written that you can use to access your data in Amazon S3 from your computer or your instance. Some of the common ones are discussed in the AWS forums.

If you have permission, you can copy a file to or from Amazon S3 and your instance using one of the following methods.

GET or wget

The **wget** utility is an HTTP and FTP client that allows you to download public objects from Amazon S3. It is installed by default in Amazon Linux and most other distributions, and available for download on Windows. To download an Amazon S3 object, use the following command, substituting the URL of the object to download.

```
[ec2-user ~]$ wget https://my_bucket.s3.amazonaws.com/path-to-file
```

This method requires that the object you request is public; if the object is not public, you receive an "ERROR 403: Forbidden" message. If you receive this error, open the Amazon S3 console and change the permissions of the object to public. For more information, see the [Amazon Simple Storage Service Developer Guide](#).

AWS Command Line Interface

The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. The AWS CLI enables users to authenticate themselves and download restricted items from Amazon S3 and also to upload items. For more information, such as how to install and configure the tools, see the [AWS Command Line Interface detail page](#).

The **aws s3 cp** command is similar to the Unix **cp** command. You can copy files from Amazon S3 to your instance, copy files from your instance to Amazon S3, and copy files from one Amazon S3 location to another.

Use the following command to copy an object from Amazon S3 to your instance.

```
[ec2-user ~]$ aws s3 cp s3://my_bucket/my_folder/my_file.ext my_copied_file.ext
```

Use the following command to copy an object from your instance back into Amazon S3.

```
[ec2-user ~]$ aws s3 cp my_copied_file.ext s3://my_bucket/my_folder/my_file.ext
```

The **aws s3 sync** command can synchronize an entire Amazon S3 bucket to a local directory location. This can be helpful for downloading a data set and keeping the local copy up-to-date with the remote set. If you have the proper permissions on the Amazon S3 bucket, you can push your local directory back up to the cloud when you are finished by reversing the source and destination locations in the command.

Use the following command to download an entire Amazon S3 bucket to a local directory on your instance.

```
[ec2-user ~]$ aws s3 sync s3://remote_S3_bucket local_directory
```

Amazon S3 API

If you are a developer, you can use an API to access data in Amazon S3. For more information, see the [Amazon Simple Storage Service Developer Guide](#). You can use this API and its examples to help develop your application and integrate it with other APIs and SDKs, such as the boto Python interface.

Using Amazon EFS with Amazon EC2

Amazon EFS provides scalable file storage for use with Amazon EC2. You can create an EFS file system and configure your instances to mount the file system. You can use an EFS file system as a common data source for workloads and applications running on multiple instances. For more information, see the [Amazon Elastic File System product page](#).

In this tutorial, you create an EFS file system and two Linux instances that can share data using the file system.

Important

Amazon EFS is not supported on Windows instances.

Tasks

- [Prerequisites \(p. 1206\)](#)
- [Step 1: Create an EFS file system \(p. 1206\)](#)
- [Step 2: Mount the file system \(p. 1207\)](#)
- [Step 3: Test the file system \(p. 1208\)](#)
- [Step 4: Clean up \(p. 1208\)](#)

Prerequisites

- Create a security group (for example, efs-sg) to associate with the EC2 instances and EFS mount target, and add the following rules:
 - Allow inbound SSH connections to the EC2 instances from your computer (the source is the CIDR block for your network).
 - Allow inbound NFS connections to the file system via the EFS mount target from the EC2 instances that are associated with this security group (the source is the security group itself). For more information, see [Amazon EFS rules \(p. 1015\)](#), and [Creating security Groups](#) in the *Amazon Elastic File System User Guide*.
- Create a key pair. You must specify a key pair when you configure your instances or you can't connect to them. For more information, see [Create a key pair \(p. 26\)](#).

Step 1: Create an EFS file system

Amazon EFS enables you to create a file system that multiple instances can mount and access at the same time. For more information, see [Creating Resources for Amazon EFS](#) in the *Amazon Elastic File System User Guide*.

To create a file system

1. Open the Amazon Elastic File System console at <https://console.aws.amazon.com/efs/>.
2. Choose **Create file system**.

3. (Optional) For **Name**, enter a name for the file system. This creates a tag with **Name** as the key and the name of the file system as the value.
4. For **Virtual Private Cloud (VPC)**, select the VPC to use for your instances.
5. Choose **Create**.
6. After the file system is created, note the file system ID. It is used later in this tutorial.
7. Choose the file system ID.
8. On the file systems page, choose **Network, Manage**. View the mount targets that Amazon EFS creates in each Availability Zone in the Region in which your VPC resides. For each Availability Zone for your instances, ensure that the value for **Security groups** is the security group that you created in [Prerequisites \(p. 1206\)](#).
9. Choose **Save**.

Step 2: Mount the file system

Use the following procedure to launch two `t2.micro` instances. Note that T2 instances must be launched in a subnet. You can use a default VPC or a nondefault VPC.

Note

There are other ways that you can mount the volume (for example, on an already running instance). For more information, see [Mounting File Systems](#) in the *Amazon Elastic File System User Guide*.

To launch two instances and mount an EFS file system

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. For **Step 1: Choose an Amazon Machine Image (AMI)**, select an Amazon Linux AMI.
4. For **Step 2: Choose an Instance Type**, keep the default instance type, `t2.micro`, and choose **Next: Configure Instance Details**.
5. For **Step 3: Configure Instance Details**, do the following:
 - a. For **Number of instances**, enter 2.
 - b. [Default VPC] If you have a default VPC, it is the default value for **Network**. Keep the default VPC and the default value for **Subnet** to use the default subnet in the Availability Zone that Amazon EC2 chooses for your instances.
[Nondefault VPC] Select your VPC for **Network**, and a public subnet from **Subnet**.
 - c. [Nondefault VPC] For **Auto-assign Public IP**, choose **Enable**. Otherwise, your instances do not get public IP addresses or public DNS names.
 - d. For **File systems**, choose **Add file system**. Ensure that the value matches the file system ID that you created in [Step 1: Create an EFS file system \(p. 1206\)](#). The path shown next to the file system ID is the mount point that the instance will use, which you can change. Under **Advanced Details**, the **User data** is automatically generated, and includes the commands needed to mount the file system.
 - e. Advance to Step 6 of the wizard.
6. On the **Configure Security Group** page, choose **Select an existing security group** and select the security group that you created in [Prerequisites \(p. 1206\)](#). Then choose **Review and Launch**.
7. On the **Review Instance Launch** page, choose **Launch**.
8. In the **Select an existing key pair or create a new key pair** dialog box, select **Choose an existing key pair** and choose your key pair. Select the acknowledgment check box, and choose **Launch Instances**.
9. In the navigation pane, choose **Instances** to see the status of your instances. Initially, their status is `pending`. After the status changes to `running`, your instances are ready for use.

Your instance is now configured to mount the Amazon EFS file system at launch and whenever it's rebooted.

Step 3: Test the file system

You can connect to your instances and verify that the file system is mounted to the directory that you specified (for example, /mnt/efs).

To verify that the file system is mounted

1. Connect to your instances. For more information, see [Connect to your Linux instance \(p. 563\)](#).
2. From the terminal window for each instance, run the `df -T` command to verify that the EFS file system is mounted.

```
$ df -T
Filesystem      Type            1K-blocks   Used      Available Use% Mounted on
/dev/xvda1      ext4           8123812  1949800       6073764  25% /
devtmpfs        devtmpfs        4078468      56       4078412  1% /dev
tmpfs          tmpfs           4089312      0       4089312  0% /dev/shm
efs-dns         nfs4           9007199254740992      0  9007199254740992  0% /mnt/efs
```

Note that the name of the file system, shown in the example output as `efs-dns`, has the following form.

```
file-system-id.efs.aws-region.amazonaws.com:/
```

3. (Optional) Create a file in the file system from one instance, and then verify that you can view the file from the other instance.
 - a. From the first instance, run the following command to create the file.

```
$ sudo touch /mnt/efs/test-file.txt
```

- b. From the second instance, run the following command to view the file.

```
$ ls /mnt/efs
test-file.txt
```

Step 4: Clean up

When you are finished with this tutorial, you can terminate the instances and delete the file system.

To terminate the instances

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instances to terminate.
4. Choose **Actions, Instance State, Terminate**.
5. Choose **Yes, Terminate** when prompted for confirmation.

To delete the file system

1. Open the Amazon Elastic File System console at <https://console.aws.amazon.com/efs/>.

2. Select the file system to delete.
3. Choose **Actions, Delete file system**.
4. When prompted for confirmation, enter the file system ID and choose **Delete file system**.

Instance volume limits

The maximum number of volumes that your instance can have depends on the operating system and instance type. When considering how many volumes to add to your instance, you should consider whether you need increased I/O bandwidth or increased storage capacity.

Contents

- [Nitro System volume limits \(p. 1209\)](#)
- [Linux-specific volume limits \(p. 1209\)](#)
- [Bandwidth versus capacity \(p. 1210\)](#)

Nitro System volume limits

Instances built on the [Nitro System \(p. 205\)](#) support a maximum number of attachments, which are shared between network interfaces, EBS volumes, and NVMe instance store volumes. Every instance has at least one network interface attachment. NVMe instance store volumes are automatically attached. For more information, see [Elastic network interfaces \(p. 789\)](#) and [Instance store volumes \(p. 1190\)](#).

Most of these instances support a maximum of 28 attachments. For example, if you have no additional network interface attachments on an EBS-only instance, you can attach up to 27 EBS volumes to it. If you have one additional network interface on an instance with 2 NVMe instance store volumes, you can attach 24 EBS volumes to it.

For other instances, the following limits apply:

- `inf1.xlarge` and `inf1.2xlarge` instances support a maximum of 26 EBS volumes.
- `inf1.6xlarge` instances support a maximum of 23 volumes.
- `inf1.24xlarge` instances support a maximum of 11 EBS volumes.
- Most bare metal instances support a maximum of 31 EBS volumes.
- `u-6tb1.metal`, `u-9tb1.metal`, and `u-12tb1.metal` instances support a maximum of 19 EBS volumes if launched after March 12, 2020 and a maximum of 14 EBS volumes otherwise. To attach more than 14 EBS volumes to an instance launched before March 12, 2020, contact your account team to upgrade the instance at no additional cost.
- `u-18tb1.metal` and `u-24tb1.metal` instances support a maximum of 19 EBS volumes.

Linux-specific volume limits

Attaching more than 40 volumes can cause boot failures. This number includes the root volume, plus any attached instance store volumes and EBS volumes. If you experience boot problems on an instance with a large number of volumes, stop the instance, detach any volumes that are not essential to the boot process, and then reattach the volumes after the instance is running.

Important

Attaching more than 40 volumes to a Linux instance is supported on a best effort basis only and is not guaranteed.

Bandwidth versus capacity

For consistent and predictable bandwidth use cases, use EBS-optimized or 10 Gigabit network connectivity instances and General Purpose SSD or Provisioned IOPS SSD volumes. Follow the guidance in [Amazon EBS-optimized instances \(p. 1139\)](#) to match the IOPS you have provisioned for your volumes to the bandwidth available from your instances for maximum performance. For RAID configurations, many administrators find that arrays larger than 8 volumes have diminished performance returns due to increased I/O overhead. Test your individual application performance and tune it as required.

Device naming on Linux instances

When you attach a volume to your instance, you include a device name for the volume. This device name is used by Amazon EC2. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 uses.

The number of volumes that your instance can support is determined by the operating system. For more information, see [Instance volume limits \(p. 1209\)](#).

Contents

- [Available device names \(p. 1210\)](#)
- [Device name considerations \(p. 1211\)](#)

For information about device names on Windows instances, see [Device naming on Windows instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Available device names

There are two types of virtualization available for Linux instances: paravirtual (PV) and hardware virtual machine (HVM). The virtualization type of an instance is determined by the AMI used to launch the instance. All instance types support HVM AMIs. Some previous generation instance types support PV AMIs. Be sure to note the virtualization type of your AMI because the recommended and available device names that you can use depend on the virtualization type of your instance. For more information, see [Linux AMI virtualization types \(p. 102\)](#).

The following table lists the available device names that you can specify in a block device mapping or when attaching an EBS volume.

Virtualization type	Available	Reserved for root	Recommended for EBS volumes	Instance store volumes
Paravirtual	/dev/sd[a-z] /dev/sd[a-z][1-15] /dev/hd[a-z] /dev/hd[a-z][1-15]	/dev/sda1	/dev/sd[f-p] /dev/sd[f-p][1-6]	/dev/sd[b-e]
HVM	/dev/sd[a-z] /dev/xvd[b-c][a-z]	Differs by AMI /dev/sda1 or /dev/xvda	/dev/sd[f-p] * /dev/sd[b-h] (h1.16xlarge)	/dev/sd[b-e] /dev/sd[b-h] (h1.16xlarge)

Virtualization type	Available	Reserved for root	Recommended for EBS volumes	Instance store volumes
				/dev/sd[b-y] (d2.8xlarge) /dev/sd[b-i] (i2.8xlarge) **

* The device names that you specify for NVMe EBS volumes in a block device mapping are renamed using NVMe device names (/dev/nvme[0-26]n1). The block device driver can assign NVMe device names in a different order than you specified for the volumes in the block device mapping.

** NVMe instance store volumes are automatically enumerated and assigned an NVMe device name.

For more information about instance store volumes, see [Amazon EC2 instance store \(p. 1188\)](#). For more information about NVMe EBS volumes (Nitro-based instances), including how to identify the EBS device, see [Amazon EBS and NVMe on Linux instances \(p. 1136\)](#).

Device name considerations

Keep the following in mind when selecting a device name:

- Although you can attach your EBS volumes using the device names used to attach instance store volumes, we strongly recommend that you don't because the behavior can be unpredictable.
- The number of NVMe instance store volumes for an instance depends on the size of the instance. NVMe instance store volumes are automatically enumerated and assigned an NVMe device name (/dev/nvme[0-26]n1).
- Depending on the block device driver of the kernel, the device could be attached with a different name than you specified. For example, if you specify a device name of /dev/sdh, your device could be renamed /dev/xvdh or /dev/hdh. In most cases, the trailing letter remains the same. In some versions of Red Hat Enterprise Linux (and its variants, such as CentOS), the trailing letter could change (/dev/sda could become /dev/xvde). In these cases, the trailing letter of each device name is incremented the same number of times. For example, if /dev/sdb is renamed /dev/xvdf, then /dev/sdc is renamed /dev/xvdg. Amazon Linux creates a symbolic link for the name you specified to the renamed device. Other operating systems could behave differently.
- HVM AMIs do not support the use of trailing numbers on device names, except for /dev/sda1, which is reserved for the root device, and /dev/sda2. While using /dev/sda2 is possible, we do not recommend using this device mapping with HVM instances.
- When using PV AMIs, you cannot attach volumes that share the same device letters both with and without trailing digits. For example, if you attach a volume as /dev/sdc and another volume as /dev/sdc1, only /dev/sdc is visible to the instance. To use trailing digits in device names, you must use trailing digits on all device names that share the same base letters (such as /dev/sdc1, /dev/sdc2, /dev/sdc3).
- Some custom kernels might have restrictions that limit use to /dev/sd[f-p] or /dev/sd[f-p][1-6]. If you're having trouble using /dev/sd[q-z] or /dev/sd[q-z][1-6], try switching to /dev/sd[f-p] or /dev/sd[f-p][1-6].

Block device mapping

Each instance that you launch has an associated root device volume, which is either an Amazon EBS volume or an instance store volume. You can use block device mapping to specify additional

EBS volumes or instance store volumes to attach to an instance when it's launched. You can also attach additional EBS volumes to a running instance; see [Attaching an Amazon EBS volume to an instance \(p. 1042\)](#). However, the only way to attach instance store volumes to an instance is to use block device mapping to attach the volumes as the instance is launched.

For more information about root device volumes, see [Changing the root volume to persist \(p. 23\)](#).

Contents

- [Block device mapping concepts \(p. 1212\)](#)
- [AMI block device mapping \(p. 1215\)](#)
- [Instance block device mapping \(p. 1217\)](#)

Block device mapping concepts

A *block device* is a storage device that moves data in sequences of bytes or bits (blocks). These devices support random access and generally use buffered I/O. Examples include hard disks, CD-ROM drives, and flash drives. A block device can be physically attached to a computer or accessed remotely as if it were physically attached to the computer.

Amazon EC2 supports two types of block devices:

- Instance store volumes (virtual devices whose underlying hardware is physically attached to the host computer for the instance)
- EBS volumes (remote storage devices)

A *block device mapping* defines the block devices (instance store volumes and EBS volumes) to attach to an instance. You can specify a block device mapping as part of creating an AMI so that the mapping is used by all instances launched from the AMI. Alternatively, you can specify a block device mapping when you launch an instance, so this mapping overrides the one specified in the AMI from which you launched the instance. Note that all NVMe instance store volumes supported by an instance type are automatically enumerated and assigned a device name on instance launch; including them in your block device mapping has no effect.

Contents

- [Block device mapping entries \(p. 1212\)](#)
- [Block device mapping instance store caveats \(p. 1213\)](#)
- [Example block device mapping \(p. 1213\)](#)
- [How devices are made available in the operating system \(p. 1214\)](#)

Block device mapping entries

When you create a block device mapping, you specify the following information for each block device that you need to attach to the instance:

- The device name used within Amazon EC2. The block device driver for the instance assigns the actual volume name when mounting the volume. The name assigned can be different from the name that Amazon EC2 recommends. For more information, see [Device naming on Linux instances \(p. 1210\)](#).

For Instance store volumes, you also specify the following information:

- The virtual device: `ephemeral[0-23]`. Note that the number and size of available instance store volumes for your instance varies by instance type.

For NVMe instance store volumes, the following information also applies:

- These volumes are automatically enumerated and assigned a device name; including them in your block device mapping has no effect.

For EBS volumes, you also specify the following information:

- The ID of the snapshot to use to create the block device (snap-xxxxxxxx). This value is optional as long as you specify a volume size.
- The size of the volume, in GiB. The specified size must be greater than or equal to the size of the specified snapshot.
- Whether to delete the volume on instance termination (`true` or `false`). The default value is `true` for the root device volume and `false` for attached volumes. When you create an AMI, its block device mapping inherits this setting from the instance. When you launch an instance, it inherits this setting from the AMI.
- The volume type, which can be `gp2` for General Purpose SSD, `io1` or `io2` for Provisioned IOPS SSD, `st1` for Throughput Optimized HDD, `sc1` for Cold HDD, or `standard` for Magnetic. The default value is `gp2`.
- The number of input/output operations per second (IOPS) that the volume supports. (Not used with `gp2`, `st1`, `sc1`, or `standard` volumes.)

Block device mapping instance store caveats

There are several caveats to consider when launching instances with AMIs that have instance store volumes in their block device mappings.

- Some instance types include more instance store volumes than others, and some instance types contain no instance store volumes at all. If your instance type supports one instance store volume, and your AMI has mappings for two instance store volumes, then the instance launches with one instance store volume.
- Instance store volumes can only be mapped at launch time. You cannot stop an instance without instance store volumes (such as the `t2.micro`), change the instance to a type that supports instance store volumes, and then restart the instance with instance store volumes. However, you can create an AMI from the instance and launch it on an instance type that supports instance store volumes, and map those instance store volumes to the instance.
- If you launch an instance with instance store volumes mapped, and then stop the instance and change it to an instance type with fewer instance store volumes and restart it, the instance store volume mappings from the initial launch still show up in the instance metadata. However, only the maximum number of supported instance store volumes for that instance type are available to the instance.

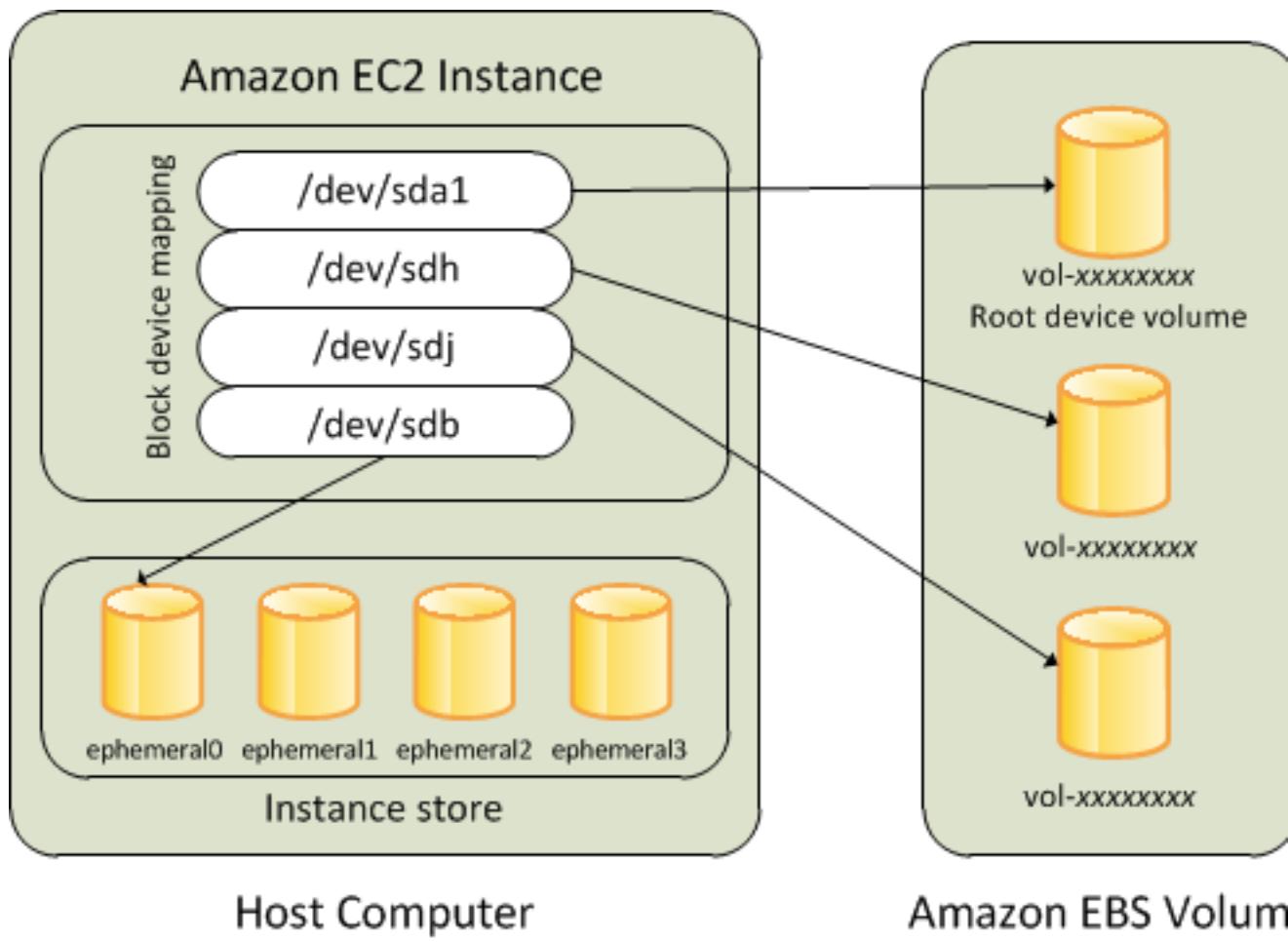
Note

When an instance is stopped, all data on the instance store volumes is lost.

- Depending on instance store capacity at launch time, M3 instances may ignore AMI instance store block device mappings at launch unless they are specified at launch. You should specify instance store block device mappings at launch time, even if the AMI you are launching has the instance store volumes mapped in the AMI, to ensure that the instance store volumes are available when the instance launches.

Example block device mapping

This figure shows an example block device mapping for an EBS-backed instance. It maps `/dev/sdb` to `ephemeral10` and maps two EBS volumes, one to `/dev/sdh` and the other to `/dev/sdj`. It also shows the EBS volume that is the root device volume, `/dev/sda1`.



Note that this example block device mapping is used in the example commands and APIs in this topic. You can find example commands and APIs that create block device mappings in [Specifying a block device mapping for an AMI \(p. 1215\)](#) and [Updating the block device mapping when launching an instance \(p. 1217\)](#).

How devices are made available in the operating system

Device names like /dev/sdh and xvdh are used by Amazon EC2 to describe block devices. The block device mapping is used by Amazon EC2 to specify the block devices to attach to an EC2 instance. After a block device is attached to an instance, it must be mounted by the operating system before you can access the storage device. When a block device is detached from an instance, it is unmounted by the operating system and you can no longer access the storage device.

With a Linux instance, the device names specified in the block device mapping are mapped to their corresponding block devices when the instance first boots. The instance type determines which instance store volumes are formatted and mounted by default. You can mount additional instance store volumes at launch, as long as you don't exceed the number of instance store volumes available for your instance type. For more information, see [Amazon EC2 instance store \(p. 1188\)](#). The block device driver for the instance determines which devices are used when the volumes are formatted and mounted. For more information, see [Attaching an Amazon EBS volume to an instance \(p. 1042\)](#).

AMI block device mapping

Each AMI has a block device mapping that specifies the block devices to attach to an instance when it is launched from the AMI. An AMI that Amazon provides includes a root device only. To add more block devices to an AMI, you must create your own AMI.

Contents

- [Specifying a block device mapping for an AMI \(p. 1215\)](#)
- [Viewing the EBS volumes in an AMI block device mapping \(p. 1216\)](#)

Specifying a block device mapping for an AMI

There are two ways to specify volumes in addition to the root volume when you create an AMI. If you've already attached volumes to a running instance before you create an AMI from the instance, the block device mapping for the AMI includes those same volumes. For EBS volumes, the existing data is saved to a new snapshot, and it's this new snapshot that's specified in the block device mapping. For instance store volumes, the data is not preserved.

For an EBS-backed AMI, you can add EBS volumes and instance store volumes using a block device mapping. For an instance store-backed AMI, you can add instance store volumes only by modifying the block device mapping entries in the image manifest file when registering the image.

Note

For M3 instances, you must specify instance store volumes in the block device mapping for the instance when you launch it. When you launch an M3 instance, instance store volumes specified in the block device mapping for the AMI may be ignored if they are not specified as part of the instance block device mapping.

To add volumes to an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**.
3. Select an instance and choose **Actions, Image, Create Image**.
4. In the **Create Image** dialog box, choose **Add New Volume**.
5. Select a volume type from the **Type** list and a device name from the **Device** list. For an EBS volume, you can optionally specify a snapshot, volume size, and volume type.
6. Choose **Create Image**.

To add volumes to an AMI using the command line

Use the [create-image](#) AWS CLI command to specify a block device mapping for an EBS-backed AMI. Use the [register-image](#) AWS CLI command to specify a block device mapping for an instance store-backed AMI.

Specify the block device mapping using the `--block-device-mappings` parameter. Arguments encoded in JSON can be supplied either directly on the command line or by reference to a file:

```
--block-device-mappings [mapping, ...]  
--block-device-mappings [file://mapping.json]
```

To add an instance store volume, use the following mapping.

```
{
```

```
        "DeviceName": "/dev/sdf",
        "VirtualName": "ephemeral0"
    }
```

To add an empty 100 GiB gp2 volume, use the following mapping.

```
{
    "DeviceName": "/dev/sdg",
    "Ebs": {
        "VolumeSize": 100
    }
}
```

To add an EBS volume based on a snapshot, use the following mapping.

```
{
    "DeviceName": "/dev/sdh",
    "Ebs": {
        "SnapshotId": "snap-xxxxxxxx"
    }
}
```

To omit a mapping for a device, use the following mapping.

```
{
    "DeviceName": "/dev/sdj",
    "NoDevice": ""
}
```

Alternatively, you can use the `-BlockDeviceMapping` parameter with the following commands (AWS Tools for Windows PowerShell):

- [New-EC2Image](#)
- [Register-EC2Image](#)

Viewing the EBS volumes in an AMI block device mapping

You can easily enumerate the EBS volumes in the block device mapping for an AMI.

To view the EBS volumes for an AMI using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **AMIs**.
3. Choose **EBS images** from the **Filter** list to get a list of EBS-backed AMIs.
4. Select the desired AMI, and look at the **Details** tab. At a minimum, the following information is available for the root device:
 - **Root Device Type (ebs)**
 - **Root Device Name** (for example, `/dev/sda1`)
 - **Block Devices** (for example, `/dev/sda1=snap-1234567890abcdef0:8:true`)

If the AMI was created with additional EBS volumes using a block device mapping, the **Block Devices** field displays the mapping for those additional volumes as well. (This screen doesn't display instance store volumes.)

To view the EBS volumes for an AMI using the command line

Use the [describe-images](#) (AWS CLI) command or [Get-EC2Image](#) (AWS Tools for Windows PowerShell) command to enumerate the EBS volumes in the block device mapping for an AMI.

Instance block device mapping

By default, an instance that you launch includes any storage devices specified in the block device mapping of the AMI from which you launched the instance. You can specify changes to the block device mapping for an instance when you launch it, and these updates overwrite or merge with the block device mapping of the AMI.

Limitations

- For the root volume, you can only modify the following: volume size, volume type, and the **Delete on Termination** flag.
- When you modify an EBS volume, you can't decrease its size. Therefore, you must specify a snapshot whose size is equal to or greater than the size of the snapshot specified in the block device mapping of the AMI.

Contents

- [Updating the block device mapping when launching an instance \(p. 1217\)](#)
- [Updating the block device mapping of a running instance \(p. 1219\)](#)
- [Viewing the EBS volumes in an instance block device mapping \(p. 1219\)](#)
- [Viewing the instance block device mapping for instance store volumes \(p. 1220\)](#)

Updating the block device mapping when launching an instance

You can add EBS volumes and instance store volumes to an instance when you launch it. Note that updating the block device mapping for an instance doesn't make a permanent change to the block device mapping of the AMI from which it was launched.

To add volumes to an instance using the console

1. Open the Amazon EC2 console.
2. From the dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, select the AMI to use and choose **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, you can modify the root volume, EBS volumes, and instance store volumes as follows:
 - To change the size of the root volume, locate the **Root** volume under the **Type** column, and change its **Size** field.
 - To suppress an EBS volume specified by the block device mapping of the AMI used to launch the instance, locate the volume and click its **Delete** icon.
 - To add an EBS volume, choose **Add New Volume**, choose **EBS** from the **Type** list, and fill in the fields (**Device**, **Snapshot**, and so on).
 - To suppress an instance store volume specified by the block device mapping of the AMI used to launch the instance, locate the volume, and choose its **Delete** icon.
 - To add an instance store volume, choose **Add New Volume**, select **Instance Store** from the **Type** list, and select a device name from **Device**.

6. Complete the remaining wizard pages, and choose **Launch**.

To add volumes to an instance using the AWS CLI

Use the [run-instances](#) AWS CLI command with the `--block-device-mappings` option to specify a block device mapping for an instance at launch.

For example, suppose that an EBS-backed AMI specifies the following block device mapping:

- `/dev/sdb=ephemeral0`
- `/dev/sdh=snap-1234567890abcdef0`
- `/dev/sdj=:100`

To prevent `/dev/sdj` from attaching to an instance launched from this AMI, use the following mapping.

```
{  
    "DeviceName": "/dev/sdj",  
    "NoDevice": ""  
}
```

To increase the size of `/dev/sdh` to 300 GiB, specify the following mapping. Notice that you don't need to specify the snapshot ID for `/dev/sdh`, because specifying the device name is enough to identify the volume.

```
{  
    "DeviceName": "/dev/sdh",  
    "Ebs": {  
        "VolumeSize": 300  
    }  
}
```

To increase the size of the root volume at instance launch, first call [describe-images](#) with the ID of the AMI to verify the device name of the root volume. For example, `"RootDeviceName": "/dev/xvda"`. To override the size of the root volume, specify the device name of the root device used by the AMI and the new volume size.

```
{  
    "DeviceName": "/dev/xvda",  
    "Ebs": {  
        "VolumeSize": 100  
    }  
}
```

To attach an additional instance store volume, `/dev/sdc`, specify the following mapping. If the instance type doesn't support multiple instance store volumes, this mapping has no effect. If the instance supports NVMe instance store volumes, they are automatically enumerated and assigned an NVMe device name.

```
{  
    "DeviceName": "/dev/sdc",  
    "VirtualName": "ephemeral1"  
}
```

To add volumes to an instance using the AWS Tools for Windows PowerShell

Use the `-BlockDeviceMapping` parameter with the [New-EC2Instance](#) command (AWS Tools for Windows PowerShell).

Updating the block device mapping of a running instance

You can use the [modify-instance-attribute](#) AWS CLI command to update the block device mapping of a running instance. You do not need to stop the instance before changing this attribute.

```
aws ec2 modify-instance-attribute --instance-id i-1a2b3c4d --block-device-mappings file://mapping.json
```

For example, to preserve the root volume at instance termination, specify the following in `mapping.json`.

```
[  
  {  
    "DeviceName": "/dev/sda1",  
    "Ebs": {  
      "DeleteOnTermination": false  
    }  
  }  
]
```

Alternatively, you can use the `--BlockDeviceMapping` parameter with the [Edit-EC2InstanceAttribute](#) command (AWS Tools for Windows PowerShell).

Viewing the EBS volumes in an instance block device mapping

You can easily enumerate the EBS volumes mapped to an instance.

Note

For instances launched before the release of the 2009-10-31 API, AWS can't display the block device mapping. You must detach and reattach the volumes so that AWS can display the block device mapping.

To view the EBS volumes for an instance using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**.
3. In the search box, enter **Root Device Type**, and then choose **EBS**. This displays a list of EBS-backed instances.
4. Select the desired instance and look at the details displayed in the **Description** tab. At a minimum, the following information is available for the root device:
 - **Root device type (ebs)**
 - **Root device** (for example, `/dev/sda1`)
 - **Block devices** (for example, `/dev/sda1`, `/dev/sdh`, and `/dev/sdj`)

If the instance was launched with additional EBS volumes using a block device mapping, the **Block devices** field displays those additional volumes as well as the root device. (This screen doesn't display instance store volumes.)

Root device type	ebs
Root device	/dev/sda1
Block devices	/dev/sda1 /dev/sdf

5. To display additional information about a block device, choose its entry next to **Block devices**. This displays the following information for the block device:

- **EBS ID** (vol-xxxxxxxx)
- **Root device type** (ebs)
- **Attachment time** (yyyy-mmThh:mm:ss.ssTZD)
- **Block device status** (attaching, attached, detaching, detached)
- **Delete on termination** (Yes, No)

To view the EBS volumes for an instance using the command line

Use the [describe-instances](#) (AWS CLI) command or [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) command to enumerate the EBS volumes in the block device mapping for an instance.

Viewing the instance block device mapping for instance store volumes

When you view the block device mapping for your instance, you can see only the EBS volumes, not the instance store volumes. You can use instance metadata to query the non-NVMe instance store volumes in the block device mapping. NVMe instance store volumes are not included.

The base URI for all requests for instance metadata is `http://169.254.169.254/latest/`. For more information, see [Instance metadata and user data \(p. 658\)](#).

First, connect to your running instance. From the instance, use this query to get its block device mapping.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/block-device-mapping/
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/
```

The response includes the names of the block devices for the instance. For example, the output for an instance store-backed m1.small instance looks like this.

```
ami
ephemeral0
root
swap
```

The `ami` device is the root device as seen by the instance. The instance store volumes are named `ephemeral[0-23]`. The `swap` device is for the page file. If you've also mapped EBS volumes, they appear as `ebs1`, `ebs2`, and so on.

To get details about an individual block device in the block device mapping, append its name to the previous query, as shown here.

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
```

```
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

The instance type determines the number of instance store volumes that are available to the instance. If the number of instance store volumes in a block device mapping exceeds the number of instance store volumes available to an instance, the additional volumes are ignored. To view the instance store volumes for your instance, run the **lsblk** command. To learn how many instance store volumes are supported by each instance type, see [Instance store volumes \(p. 1190\)](#).

Resources and tags

Amazon EC2 provides different *resources* that you can create and use. Some of these resources include images, instances, volumes, and snapshots. When you create a resource, we assign the resource a unique resource ID.

Some resources can be tagged with values that you define, to help you organize and identify them.

The following topics describe resources and tags, and how you can work with them.

Contents

- [Resource locations \(p. 1222\)](#)
- [Resource IDs \(p. 1223\)](#)
- [Listing and filtering your resources \(p. 1224\)](#)
- [Tagging your Amazon EC2 resources \(p. 1229\)](#)
- [Amazon EC2 service quotas \(p. 1241\)](#)
- [Amazon EC2 usage reports \(p. 1243\)](#)

Resource locations

Some resources can be used in all regions (global), and some resources are specific to the region or Availability Zone in which they reside.

Resource	Type	Description
AWS account	Global	You can use the same AWS account in all regions.
Key pairs	Global or Regional	<p>The key pairs that you create using Amazon EC2 are tied to the Region where you created them. You can create your own RSA key pair and upload it to the region in which you want to use it; therefore, you can make your key pair globally available by uploading it to each Region.</p> <p>For more information, see Amazon EC2 key pairs and Linux instances (p. 985).</p>
Amazon EC2 resource identifiers	Regional	Each resource identifier, such as an AMI ID, instance ID, EBS volume ID, or EBS snapshot ID, is tied to its Region and can be used only in the Region where you created the resource.
User-supplied resource names	Regional	Each resource name, such as a security group name or key pair name, is tied to its region and can be used only in the Region where you created the resource. Although you can create resources with the same name in multiple regions, they aren't related to each other.
AMIs	Regional	An AMI is tied to the Region where its files are located within Amazon S3. You can copy an AMI from one Region to another. For more information, see Copying an AMI (p. 163) .

Resource	Type	Description
Elastic IP addresses	Regional	An Elastic IP address is tied to a Region and can be associated only with an instance in the same Region.
Security groups	Regional	A security group is tied to a Region and can be assigned only to instances in the same Region. You can't enable an instance to communicate with an instance outside its Region using security group rules. Traffic from an instance in another Region is seen as WAN bandwidth.
EBS snapshots	Regional	An EBS snapshot is tied to its Region and can only be used to create volumes in the same Region. You can copy a snapshot from one Region to another. For more information, see Copying an Amazon EBS snapshot (p. 1068) .
EBS volumes	Availability Zone	An Amazon EBS volume is tied to its Availability Zone and can be attached only to instances in the same Availability Zone.
Instances	Availability Zone	An instance is tied to the Availability Zones in which you launched it. However, its instance ID is tied to the Region.

Resource IDs

When resources are created, we assign each resource a unique resource ID. A resource ID takes the form of a resource identifier (such as `snap` for a snapshot) followed by a hyphen and a unique combination of letters and numbers.

You can use resource IDs to find your resources in the Amazon EC2 console. If you are using a command line tool or the Amazon EC2 API to work with Amazon EC2, resource IDs are required for certain commands. For example, if you are using the [stop-instances](#) AWS CLI command to stop an instance, you must specify the instance ID in the command.

Resource ID length

Prior to January 2016, the IDs assigned to newly created resources of certain resource types used 8 characters after the hyphen (for example, `i-1a2b3c4d`). From January 2016 to June 2018, we changed the IDs of these resource types to use 17 characters after the hyphen (for example, `i-1234567890abcdef0`). Depending on when your account was created, you might have resources of the following resource types with short IDs, though any new resources of these types receive the longer IDs:

- `bundle`
- `conversion-task`
- `customer-gateway`
- `dhcp-options`
- `elastic-ip-allocation`
- `elastic-ip-association`
- `export-task`
- `flow-log`
- `image`

- import-task
- instance
- internet-gateway
- network-acl
- network-acl-association
- network-interface
- network-interface-attachment
- prefix-list
- route-table
- route-table-association
- security-group
- snapshot
- subnet
- subnet-cidr-block-association
- reservation
- volume
- vpc
- vpc-cidr-block-association
- vpc-endpoint
- vpc-peering-connection
- vpn-connection
- vpn-gateway

Listing and filtering your resources

You can get a list of some types of resources using the Amazon EC2 console. You can get a list of each type of resource using its corresponding command or API action. If you have many resources, you can filter the results to include only the resources that match certain criteria.

Contents

- [Listing and filtering resources using the console \(p. 1224\)](#)
- [Listing and filtering using the CLI and API \(p. 1227\)](#)

Listing and filtering resources using the console

Contents

- [Listing resources using the console \(p. 1224\)](#)
- [Filtering resources using the console \(p. 1225\)](#)

Listing resources using the console

You can view the most common Amazon EC2 resource types using the console. To view additional resources, use the command line interface or the API actions.

To list EC2 resources using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose the option that corresponds to the resource type. For example, to list your instances, choose **Instances**.
3. The page displays all resources of the selected resource type.

Filtering resources using the console

Search functionality differs slightly between the *old* and *new* Amazon EC2 console.

New console

The new console supports two types of filtering.

- *API filtering* happens on the server side. The filtering is applied on the API call and it reduces the number of resources returned by the server. It allows for quick filtering across large sets of resources, and it can reduce data transfer time and cost between the server and the browser.
- *Client filtering* happens on the client side. It enables you to filter down on data that is already available in the browser (in other words, data that has already been returned by the API). Client filtering works well in conjunction with an API filter to filter down to smaller data sets in the browser.

The new Amazon EC2 console supports the following types of searches:

Search by keyword

Searching by keyword is a free text search that lets you search for a value across all of your resources' attributes, without specifying an attribute to search.

Note

All keyword searches use *client filtering*.

To search by keyword, enter or paste what you're looking for in the search field, and then choose **Enter**. For example, searching for 123 matches all instances that have 123 in any of their attributes, such as an IP address, instance ID, VPC ID, or AMI ID. If your free text search returns unexpected matches, apply additional filters.

Search by attributes

Searching by an attribute lets you search a specific attribute across all of your resources.

Note

Attribute searches use either *API filtering* or *client filtering*, depending on the selected attribute. When performing an attribute search, the attributes are grouped accordingly.

For example, you can search the **Instance State** attribute for all of your instances to return only instances that are in the stopped state. To do this:

1. In the search field on the Instances screen, start entering `Instance State`. As you enter characters, a list of matching attributes appears.
2. Select **Instance State** from the list. A list of possible values for the selected attribute appears.
3. Select **Stopped** from the list.

You can use the following techniques to enhance or refine your searches:

Inverse search

Inverse searches let you search for resources that do **not** match a specified value. Inverse searches are performed by prefixing the search keyword with the exclamation mark (!) character. For example, to list all instances that are **not** assigned the security group named `launch-wizard-1`, search by the **Security group name** attribute, and for the keyword, enter `!launch-wizard-1`.

Note

Inverse search is supported with keyword searches and attribute searches on client filters only. It is not supported with attribute searches on API filters.

Partial search

With partial searches, you can search for partial string values. To perform a partial search, enter only a part of the keyword that you want to search for. For example, to search for all t2.micro, t2.small, and t2.medium instances, search by the **Instance Type** attribute, and for the keyword, enter t2.

Note

Partial search is supported with keyword searches and attribute searches on client filters only. It is not supported with attribute searches on API filters.

Regular expression search

To use regular expression searches, you must enable **Use regular expression matching** in the Preferences.

Regular expressions are useful when you need to match the values in a field with a specific pattern. For example, to search for a value that starts with s, search for ^s. To search for a value that ends with xyz, search for xyz\$. Or to search for a value that starts with a number that is followed by one or more characters, search for [0-9]+.*. Regular expression searches are not case-sensitive.

Note

Regular expression search is supported with keyword searches and attribute searches on client filters only. It is not supported with attribute searches on API filters.

Wildcard search

Use the * wildcard to match zero or more characters. Use the ? wildcard to match zero or one character. For example, if you have a data set with the following values: prod, prods, and production; "prod*" matches all values, whereas "prod?" matches only prod and prods. To use the literals values, escape them with a backslash (\). For example, "prod*" would match prod*.

Note

Wildcard search is supported with attribute searches on API filters only. It is not supported with keyword searches and attribute searches on client filters only.

Combining searches

In general, multiple filters with the same attribute are automatically joined with OR. For example, searching for `Instance State : Running` and `Instance State : Stopped` returns all instances that are either running OR stopped. To join search with AND, search across different attributes. For example, searching for `Instance State : Running` and `Instance Type : c4.large` returns only instances that are of type `c4.large` AND that are in the stopped state.

Old console

The old Amazon EC2 console supports the following types of searches:

Search by keyword

Searching by keyword is a free text search that lets you search for a value across all of your resources' attributes. To search by keyword, enter or paste what you're looking for in the search field, and then choose **Enter**. For example, searching for 123 matches all instances that have 123 in any of their attributes, such as an IP address, instance ID, VPC ID, or AMI ID. If your free text search returns unexpected matches, apply additional filters.

Search by attributes

Searching by an attribute lets you search a specific attribute across all of your resources. For example, you can search the **State** attribute for all of your instances to return only instances that are in the stopped state. To do this:

1. In the search field on the Instances screen, start entering `Instance State`. As you enter characters, a list of matching attributes appears.
2. Select **Instance State** from the list. A list of possible values for the selected attribute appears.
3. Select **Stopped** from the list.

You can use the following techniques to enhance or refine your searches:

Inverse search

Inverse searches let you search for resources that do **not** match a specified value. Inverse searches are performed by prefixing the search keyword with the exclamation mark (!) character. For example, to list all instances that are **not** terminated, search by the **Instance State** attribute, and for the keyword, enter `!Terminated`.

Partial search

With partial searches, you can search for partial string values. To perform a partial search, enter only a part of the keyword you want to search for. For example, to search for all `t2.micro`, `t2.small`, and `t2.medium` instances, search by the **Instance Type** attribute, and for the keyword, enter `t2`.

Regular expression search

Regular expressions are useful when you need to match the values in a field with a specific pattern. For example, to search for all instances that have an attribute value that starts with `s`, search for `^s`. Or to search for all instances that have an attribute value that ends with `xyz`, search for `xyz$`. Regular expression searches are not case-sensitive.

Combining searches

In general, multiple filters with the same attribute are automatically joined with OR. For example, searching for `Instance State : Running` and `Instance State : Stopped` returns all instances that are either running OR stopped. To join search with AND, search across different attributes. For example, searching for `Instance State : Running` and `Instance Type : c4.large` returns only instances that are of type `c4.large` AND that are in the stopped state.

To filter a list of resources

1. In the navigation pane, select a resource type (for example, **Instances**).
2. Choose the search field.
3. Choose the filter from in the list.
4. Specify a filter value.
5. When you are finished, remove the filter.

Listing and filtering using the CLI and API

Each resource type has a corresponding CLI command and API action that you use to list resources of that type. The resulting lists of resources can be long, so it can be faster and more useful to filter the results to include only the resources that match specific criteria.

Filtering considerations

- You can specify multiple filters and multiple filter values in a single request.
- You can use wildcards with the filter values. An asterisk (*) matches zero or more characters, and a question mark (?) matches zero or one character.
- Filter values are case sensitive.

- Your search can include the literal values of the wildcard characters; you just need to escape them with a backslash before the character. For example, a value of *amazon\?\\" searches for the literal string *amazon?\\.

Supported filters

To see the supported filters for each Amazon EC2 resource, see the following documentation:

- AWS CLI: The `describe` commands in the [AWS CLI Command Reference-Amazon EC2](#).
- Tools for Windows PowerShell: The `Get` commands in the [AWS Tools for PowerShell Cmdlet Reference-Amazon EC2](#).
- Query API: The `Describe` API actions in the [Amazon EC2 API Reference](#).

Example Example: Specify a single filter

You can list your Amazon EC2 instances using [describe-instances](#). Without filters, the response contains information for all of your resources. You can use the following command to include only the running instances in your output.

```
aws ec2 describe-instances --filters Name=instance-state-name,Values=running
```

To list only the instance IDs for your running instances, add the `--query` parameter as follows.

```
aws ec2 describe-instances --filters Name=instance-state-name,Values=running --query "Reservations[*].Instances[*].InstanceId" --output text
```

The following is example output.

```
i-0ef1f57f78d4775a4
i-0626d4edd54f1286d
i-04a636d18e83cfacb
```

Example Example: Specify multiple filters or filter values

If you specify multiple filters or multiple filter values, the resource must match all filters to be included in the results.

You can use the following command to list all instances whose type is either `m5.large` or `m5d.large`.

```
aws ec2 describe-instances --filters Name=instance-type,Values=m5.large,m5d.large
```

You can use the following command to list all stopped instances whose type is `t2.micro`.

```
aws ec2 describe-instances --filters Name=instance-state-name,Values=stopped Name=instance-type,Values=t2.micro
```

Example Example: Use wildcards in a filter value

If you specify `database` as the filter value for the `description` filter when describing EBS snapshots using [describe-snapshots](#), the command returns only the snapshots whose description is "database".

```
aws ec2 describe-snapshots --filters Name=description,Values=database
```

The `*` wildcard matches zero or more characters. If you specify `*database*` as the filter value, the command returns only snapshots whose description includes the word database.

```
aws ec2 describe-snapshots --filters Name=description,Values=*database*
```

The ? wildcard matches exactly 1 character. If you specify database? as the filter value, the command returns only snapshots whose description is "database" or "database" followed by one character.

```
aws ec2 describe-snapshots --filters Name=description,Values=database?
```

If you specify database????, the command returns only snapshots whose description is "database" followed by up to four characters. It excludes descriptions with "database" followed by five or more characters.

```
aws ec2 describe-snapshots --filters Name=description,Values=database????
```

Example Example: Filter based on date

With the AWS CLI, you can use JMESPath to filter results using expressions. For example, the following [describe-snapshots](#) command displays the IDs of all snapshots created by your AWS account (represented by `123456789012`) before the specified date (represented by `2020-03-31`). If you do not specify the owner, the results include all public snapshots.

```
aws ec2 describe-snapshots --filters Name=owner-id,Values=123456789012 --query "Snapshots[?(StartTime<= `2020-03-31`)].[SnapshotId]" --output text
```

The following command displays the IDs of all snapshots created in the specified date range.

```
aws ec2 describe-snapshots --filters Name=owner-id,Values=123456789012 --query "Snapshots[?(StartTime>= `2019-01-01` && (StartTime<= `2019-12-31`)].[SnapshotId]" --output text
```

Filter based on tags

For examples of how to filter a list of resources according to their tags, see [Working with tags using the command line \(p. 1237\)](#).

Tagging your Amazon EC2 resources

To help you manage your instances, images, and other Amazon EC2 resources, you can assign your own metadata to each resource in the form of *tags*. Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type—you can quickly identify a specific resource based on the tags that you've assigned to it. This topic describes tags and shows you how to create them.

Warning

Tag keys and their values are returned by many different API calls. Denying access to `DescribeTags` doesn't automatically deny access to tags returned by other APIs. As a best practice, we recommend that you do not include sensitive data in your tags.

Contents

- [Tag basics \(p. 1230\)](#)
- [Tagging your resources \(p. 1231\)](#)
- [Tag restrictions \(p. 1233\)](#)
- [Tagging your resources for billing \(p. 1234\)](#)
- [Working with tags using the console \(p. 1234\)](#)

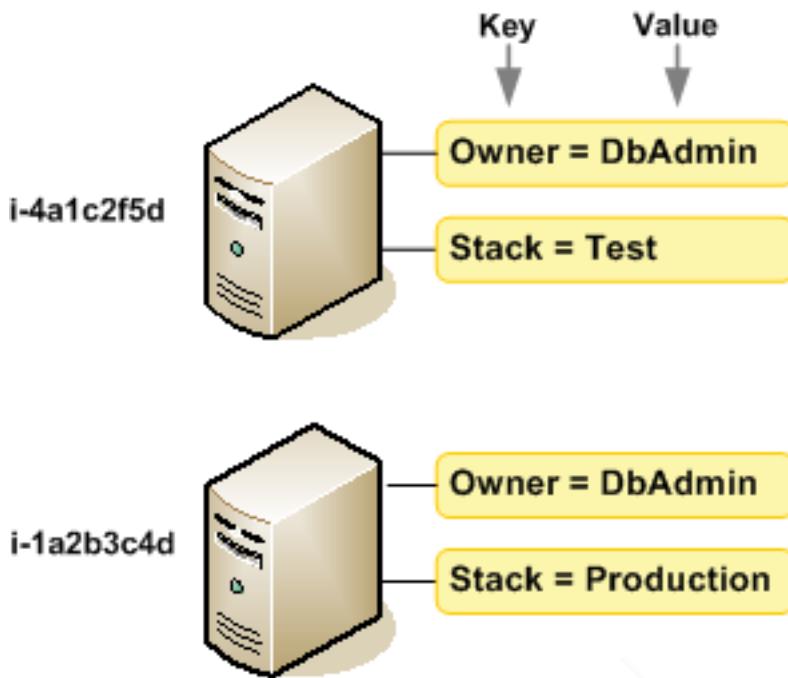
- [Working with tags using the command line \(p. 1237\)](#)
- [Adding tags to a resource using CloudFormation \(p. 1240\)](#)

Tag basics

A tag is a label that you assign to an AWS resource. Each tag consists of a *key* and an optional *value*, both of which you define.

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. For example, you could define a set of tags for your account's Amazon EC2 instances that helps you track each instance's owner and stack level.

The following diagram illustrates how tagging works. In this example, you've assigned two tags to each of your instances—one tag with the key `Owner` and another with the key `Stack`. Each tag also has an associated value.



We recommend that you devise a set of tag keys that meets your needs for each resource type. Using a consistent set of tag keys makes it easier for you to manage your resources. You can search and filter the resources based on the tags you add. For more information about how to implement an effective resource tagging strategy, see the AWS whitepaper [Tagging Best Practices](#).

Tags don't have any semantic meaning to Amazon EC2 and are interpreted strictly as a string of characters. Also, tags are not automatically assigned to your resources. You can edit tag keys and values, and you can remove tags from a resource at any time. You can set the value of a tag to an empty string, but you can't set the value of a tag to null. If you add a tag that has the same key as an existing tag on that resource, the new value overwrites the old value. If you delete a resource, any tags for the resource are also deleted.

You can work with tags using the AWS Management Console, the AWS CLI, and the Amazon EC2 API.

If you're using AWS Identity and Access Management (IAM), you can control which users in your AWS account have permission to create, edit, or delete tags. For more information, see [Identity and access management for Amazon EC2 \(p. 919\)](#).

Tagging your resources

You can tag most Amazon EC2 resources that already exist in your account. The [table \(p. 1231\)](#) below lists the resources that support tagging.

If you're using the Amazon EC2 console, you can apply tags to resources by using the **Tags** tab on the relevant resource screen, or you can use the **Tags** screen. Some resource screens enable you to specify tags for a resource when you create the resource; for example, a tag with a key of `Name` and a value that you specify. In most cases, the console applies the tags immediately after the resource is created (rather than during resource creation). The console may organize resources according to the `Name` tag, but this tag doesn't have any semantic meaning to the Amazon EC2 service.

If you're using the Amazon EC2 API, the AWS CLI, or an AWS SDK, you can use the `CreateTags` EC2 API action to apply tags to existing resources. Additionally, some resource-creating actions enable you to specify tags for a resource when the resource is created. If tags cannot be applied during resource creation, we roll back the resource creation process. This ensures that resources are either created with tags or not created at all, and that no resources are left untagged at any time. By tagging resources at the time of creation, you can eliminate the need to run custom tagging scripts after resource creation.

The following table describes the Amazon EC2 resources that can be tagged, and the resources that can be tagged on creation using the Amazon EC2 API, the AWS CLI, or an AWS SDK.

Tagging support for Amazon EC2 resources

Resource	Supports tags	Supports tagging on creation
AFI	Yes	Yes
AMI	Yes	No
Bundle task	No	No
Capacity Reservation	Yes	Yes
Carrier gateway	Yes	Yes
Client VPN endpoint	Yes	Yes
Client VPN route	No	No
Customer gateway	Yes	Yes
Dedicated Host	Yes	Yes
Dedicated Host Reservation	Yes	Yes
DHCP option	Yes	Yes
EBS snapshot	Yes	Yes
EBS volume	Yes	Yes
EC2 Fleet	Yes	Yes
Egress-only internet gateway	Yes	Yes
Elastic IP address	Yes	No

Resource	Supports tags	Supports tagging on creation
Elastic Graphics accelerator	Yes	No
Instance	Yes	Yes
Instance store volume	N/A	N/A
Internet gateway	Yes	Yes
IP address pool (BYOIP)	Yes	Yes
Key pair	Yes	Yes
Launch template	Yes	Yes
Launch template version	No	No
Local gateway	Yes	No
Local gateway route table	Yes	No
Local gateway virtual interface	Yes	No
Local gateway virtual interface group	Yes	No
Local gateway route table VPC association	Yes	Yes
Local gateway route table virtual interface group association	Yes	No
NAT gateway	Yes	Yes
Network ACL	Yes	Yes
Network interface	Yes	Yes
Placement group	Yes	Yes
Prefix list	Yes	Yes
Reserved Instance	Yes	No
Reserved Instance listing	No	No
Route table	Yes	Yes
Spot Fleet request	Yes	Yes
Spot Instance request	Yes	Yes
Security group	Yes	Yes
Subnet	Yes	Yes
Traffic Mirror filter	Yes	Yes
Traffic Mirror session	Yes	Yes
Traffic Mirror target	Yes	Yes

Resource	Supports tags	Supports tagging on creation
Transit gateway	Yes	Yes
Transit gateway route table	Yes	Yes
Transit gateway VPC attachment	Yes	Yes
Virtual private gateway	Yes	Yes
VPC	Yes	Yes
VPC endpoint	Yes	Yes
VPC endpoint service	Yes	Yes
VPC endpoint service configuration	Yes	Yes
VPC flow log	Yes	Yes
VPC peering connection	Yes	Yes
VPN connection	Yes	Yes

You can tag instances and volumes on creation using the Amazon EC2 Launch Instances wizard in the Amazon EC2 console. You can tag your EBS volumes on creation using the Volumes screen, or EBS snapshots using the Snapshots screen. Alternatively, use the resource-creating Amazon EC2 APIs (for example, [RunInstances](#)) to apply tags when creating your resource.

You can apply tag-based resource-level permissions in your IAM policies to the Amazon EC2 API actions that support tagging on creation to implement granular control over the users and groups that can tag resources on creation. Your resources are properly secured from creation—tags are applied immediately to your resources, therefore any tag-based resource-level permissions controlling the use of resources are immediately effective. Your resources can be tracked and reported on more accurately. You can enforce the use of tagging on new resources, and control which tag keys and values are set on your resources.

You can also apply resource-level permissions to the `CreateTags` and `DeleteTags` Amazon EC2 API actions in your IAM policies to control which tag keys and values are set on your existing resources. For more information, see [Example: Tagging resources \(p. 958\)](#).

For more information about tagging your resources for billing, see [Using Cost Allocation Tags in the AWS Billing and Cost Management User Guide](#).

Tag restrictions

The following basic restrictions apply to tags:

- Maximum number of tags per resource – 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length – 128 Unicode characters in UTF-8
- Maximum value length – 256 Unicode characters in UTF-8
- Although EC2 allows for any character in its tags, other services are more restrictive. The allowed characters across services are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.

- The `aws:` prefix is reserved for AWS use. If a tag has a tag key with this prefix, then you can't edit or delete the tag's key or value. Tags with the `aws:` prefix do not count against your tags per resource limit.

You can't terminate, stop, or delete a resource based solely on its tags; you must specify the resource identifier. For example, to delete snapshots that you tagged with a tag key called `DeleteMe`, you must use the `DeleteSnapshots` action with the resource identifiers of the snapshots, such as `snap-1234567890abcdef0`.

You can tag public or shared resources, but the tags you assign are available only to your AWS account and not to the other accounts sharing the resource.

You can't tag all resources. For more information, see [Tagging support for Amazon EC2 resources \(p. 1231\)](#).

Tagging your resources for billing

You can use tags to organize your AWS bill to reflect your own cost structure. To do this, sign up to get your AWS account bill with tag key values included. For more information about setting up a cost allocation report with tags, see [The Monthly Cost Allocation Report](#) in *AWS Billing and Cost Management User Guide*. To see the cost of your combined resources, you can organize your billing information based on resources that have the same tag key values. For example, you can tag several resources with a specific application name, and then organize your billing information to see the total cost of that application across several services. For more information, see [Using Cost Allocation Tags](#) in the *AWS Billing and Cost Management User Guide*.

Note

If you've just enabled reporting, data for the current month is available for viewing after 24 hours.

Cost allocation tags can indicate which resources are contributing to costs, but deleting or deactivating resources doesn't always reduce costs. For example, snapshot data that is referenced by another snapshot is preserved, even if the snapshot that contains the original data is deleted. For more information, see [Amazon Elastic Block Store Volumes and Snapshots](#) in the *AWS Billing and Cost Management User Guide*.

Note

Elastic IP addresses that are tagged do not appear on your cost allocation report.

Working with tags using the console

Using the Amazon EC2 console, you can see which tags are in use across all of your Amazon EC2 resources in the same Region. You can view tags by resource and by resource type, and you can also view how many items of each resource type are associated with a specified tag. You can also use the Amazon EC2 console to apply or remove tags from one or more resources at a time.

For more information about using filters when listing your resources, see [Listing and filtering your resources \(p. 1224\)](#).

For ease of use and best results, use Tag Editor in the AWS Management Console, which provides a central, unified way to create and manage your tags. For more information, see [Working with Tag Editor](#) in *Getting Started with the AWS Management Console*.

Tasks

- [Displaying tags \(p. 1235\)](#)
- [Adding and deleting tags on an individual resource \(p. 1235\)](#)

- [Adding and deleting tags to a group of resources \(p. 1236\)](#)
- [Adding a tag when you launch an instance \(p. 1237\)](#)
- [Filtering a list of resources by tag \(p. 1237\)](#)

Displaying tags

You can display tags in two different ways in the Amazon EC2 console. You can display the tags for an individual resource or for all resources.

Displaying tags for individual resources

When you select a resource-specific page in the Amazon EC2 console, it displays a list of those resources. For example, if you select **Instances** from the navigation pane, the console displays your Amazon EC2 instances. When you select a resource from one of these lists (for example, an instance), if the resource supports tags, you can view and manage its tags. On most resource pages, you can view the tags by selecting the **Tags** tab.

You can add a column to the resource list that displays all values for tags with the same key. This column enables you to sort and filter the resource list by the tag. There are two ways to add a new column to the resource list to display your tags:

- On the **Tags** tab, select **Show Column**. A new column is added to the console.
- Choose the **Show/Hide Columns** gear-shaped icon, and in the **Show/Hide Columns** dialog box, select the tag key under **Your Tag Keys**.

Displaying tags for all resources

You can display tags across all resources by selecting **Tags** from the navigation pane in the Amazon EC2 console. The following image shows the **Tags** pane, which lists all tags in use by resource type.

The screenshot shows a table titled "Manage Tags" with a header row containing columns for Tag Key, Tag Value, Total, Instances, AMIs, and Volumes. Below the header, there are seven data rows. The data is as follows:

Tag Key	Tag Value	Total	Instances	AMIs	Volumes
Manage Tag	Name	DNS Server	1	1	0
Manage Tag	Owner	TeamB	2	0	2
Manage Tag	Owner	TeamA	2	0	2
Manage Tag	Purpose	Project2	1	0	1
Manage Tag	Purpose	Logs	1	0	1
Manage Tag	Purpose	Network Management	1	1	0
Manage Tag	Purpose	Project1	2	0	2

Adding and deleting tags on an individual resource

You can manage tags for an individual resource directly from the resource's page.

To add a tag to an individual resource

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can't. For more information, see [Resource locations \(p. 1222\)](#).
3. In the navigation pane, select a resource type (for example, [Instances](#)).
4. Select the resource from the resource list and choose the **Tags** tab.
5. Choose **Manage tags**, **Add tag**. Enter the key and value for the tag. When you are finished adding tags, choose **Save**.

To delete a tag from an individual resource

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can't. For more information, see [Resource locations \(p. 1222\)](#).
3. In the navigation pane, choose a resource type (for example, [Instances](#)).
4. Select the resource from the resource list and choose the **Tags** tab.
5. Choose **Manage tags**. For each tag, choose **Remove**. When you are finished removing tags, choose **Save**.

Adding and deleting tags to a group of resources

To add a tag to a group of resources

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can't. For more information, see [Resource locations \(p. 1222\)](#).
3. In the navigation pane, choose **Tags**.
4. At the top of the content pane, choose **Manage Tags**.
5. For **Filter**, select the type of resource (for example, instances).
6. In the resources list, select the check box next to each resource.
7. Under **Add Tag**, enter the tag key and value and choose **Add Tag**.

Note

If you add a new tag with the same tag key as an existing tag, the new tag overwrites the existing tag.

To remove a tag from a group of resources

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the Region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can't. For more information, see [Resource locations \(p. 1222\)](#).
3. In the navigation pane, choose **Tags**, **Manage Tags**.
4. To view the tags in use, select the **Show/Hide Columns** gear-shaped icon, and in the **Show/Hide Columns** dialog box, select the tag keys to view and choose **Close**.
5. For **Filter**, select the type of resource (for example, instances).
6. In the resource list, select the check box next to each resource.

7. Under **Remove Tag**, enter the tag key and choose **Remove Tag**.

Adding a tag when you launch an instance

To add a tag using the Launch Wizard

1. From the navigation bar, select the Region for the instance. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can't. Select the Region that meets your needs. For more information, see [Resource locations \(p. 1222\)](#).
2. Choose **Launch Instance**.
3. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations called Amazon Machine Images (AMIs). Select the AMI to use and choose **Select**. For more information about selecting an AMI, see [Finding an AMI](#).
4. On the **Configure Instance Details** page, configure the instance settings as necessary, and then choose **Next: Add Storage**.
5. On the **Add Storage** page, you can specify additional storage volumes for your instance. Choose **Next: Add Tags** when done.
6. On the **Add Tags** page, specify tags for the instance, the volumes, or both. Choose **Add another tag** to add more than one tag to your instance. Choose **Next: Configure Security Group** when you are done.
7. On the **Configure Security Group** page, you can choose from an existing security group that you own, or let the wizard create a new security group for you. Choose **Review and Launch** when you are done.
8. Review your settings. When you're satisfied with your selections, choose **Launch**. Select an existing key pair or create a new one, select the acknowledgment check box, and then choose **Launch Instances**.

Filtering a list of resources by tag

You can filter your list of resources based on one or more tag keys and tag values.

To filter a list of resources by tag

1. In the navigation pane, select a resource type (for example, **Instances**).
2. Choose the search field.
3. Choose the tag key from in the list.
4. Choose the corresponding tag value from the list.
5. When you are finished, remove the filter.

For more information about filters, see [Listing and filtering your resources \(p. 1224\)](#).

Working with tags using the command line

You can add tags to many EC2 resource when you create them, using the tag specifications parameter for the create command. You can view the tags for a resource using the describe command for the resource. You can also add, update, or delete tags for your existing resources using the following commands.

Task	AWS CLI	AWS Tools for Windows PowerShell
Add or overwrite one or more tags	create-tags	New-EC2Tag

Task	AWS CLI	AWS Tools for Windows PowerShell
Delete one or more tags	delete-tags	Remove-EC2Tag
Describe one or more tags	describe-tags	Get-EC2Tag

Tasks

- [Adding tags on resource creation \(p. 1238\)](#)
- [Adding tags to an existing resource \(p. 1239\)](#)
- [Describing tagged resources \(p. 1240\)](#)

Adding tags on resource creation

The following examples demonstrate how to apply tags when you create resources.

The way you enter JSON-formatted parameters on the command line differs depending on your operating system. Linux, macOS, or Unix and Windows PowerShell use single quotes ('') to enclose the JSON data structure. Omit the single quotes when using the commands with the Windows command line. For more information, see [Specifying Parameter Values for the AWS Command Line Interface](#).

Example Example: Launch an instance and apply tags to the instance and volume

The following `run-instances` command launches an instance and applies a tag with the key `webserver` and the value `production` to the instance. The command also applies a tag with the key `cost-center` and the value `cc123` to any EBS volume that's created (in this case, the root volume).

```
aws ec2 run-instances \
--image-id ami-abc12345 \
--count 1 \
--instance-type t2.micro \
--key-name MyKeyPair \
--subnet-id subnet-6e7f829e \
--tag-specifications 'ResourceType=instance,Tags=[{"Key=webserver,Value=production"}]' \
'ResourceType=volume,Tags=[{"Key=cost-center,Value=cc123"}]'
```

You can apply the same tag keys and values to both instances and volumes during launch. The following command launches an instance and applies a tag with a key of `cost-center` and a value of `cc123` to both the instance and any EBS volume that's created.

```
aws ec2 run-instances \
--image-id ami-abc12345 \
--count 1 \
--instance-type t2.micro \
--key-name MyKeyPair \
--subnet-id subnet-6e7f829e \
--tag-specifications 'ResourceType=instance,Tags=[{"Key=cost-center,Value=cc123"}]' \
'ResourceType=volume,Tags=[{"Key=cost-center,Value=cc123"}]'
```

Example Example: Create a volume and apply a tag

The following `create-volume` command creates a volume and applies two tags: `purpose=production` and `cost-center=cc123`.

```
aws ec2 create-volume \
```

```
--availability-zone us-east-1a \
--volume-type gp2 \
--size 80 \
--tag-specifications 'ResourceType=volume,Tags=[{Key=purpose,Value=production},
{Key=cost-center,Value=cc123}]'
```

Adding tags to an existing resource

The following examples demonstrate how to add tags to an existing resource using the [create-tags](#) command.

Example Example: Add a tag to a resource

The following command adds the tag **Stack=production** to the specified image, or overwrites an existing tag for the AMI where the tag key is **Stack**. If the command succeeds, no output is returned.

```
aws ec2 create-tags \
--resources ami-78a54011 \
--tags Key=Stack,Value=production
```

Example Example: Add tags to multiple resources

This example adds (or overwrites) two tags for an AMI and an instance. One of the tags contains just a key (**webserver**), with no value (we set the value to an empty string). The other tag consists of a key (**stack**) and value (**Production**). If the command succeeds, no output is returned.

```
aws ec2 create-tags \
--resources ami-1a2b3c4d i-1234567890abcdef0 \
--tags Key=webserver,Value= Key=stack,Value=Production
```

Example Example: Add tags with special characters

This example adds the tag **[Group]=test** to an instance. The square brackets ([and]) are special characters, which must be escaped.

If you are using Linux or OS X, to escape the special characters, enclose the element with the special character with double quotes ("), and then enclose the entire key and value structure with single quotes (').

```
aws ec2 create-tags \
--resources i-1234567890abcdef0 \
--tags 'Key="["Group"]",Value=test'
```

If you are using Windows, to escape the special characters, enclose the element that has special characters with double quotes ("), and then precede each double quote character with a backslash (\) as follows:

```
aws ec2 create-tags ^
--resources i-1234567890abcdef0 ^
--tags Key="\["Group"\]",Value=test
```

If you are using Windows PowerShell, to escape the special characters, enclose the value that has special characters with double quotes ("), precede each double quote character with a backslash (\), and then enclose the entire key and value structure with single quotes (') as follows:

```
aws ec2 create-tags ^
```

```
--resources i-1234567890abcdef0
--tags 'Key=\"[Group]\",Value=test'
```

Describing tagged resources

The following examples show you how to use filters with the [describe-instances](#) to view instances with specific tags. All EC2 describe commands use this syntax to filter by tag across a single resource type. Alternatively, you can use the [describe-tags](#) command to filter by tag across EC2 resource types.

Example Example: Describe instances with the specified tag key

The following command describes the instances with a **Stack** tag, regardless of the value of the tag.

```
aws ec2 describe-instances \
--filters Name=tag-key,Values=Stack
```

Example Example: Describe instances with the specified tag

The following command describes the instances with the tag **Stack=production**.

```
aws ec2 describe-instances \
--filters Name=tag:Stack,Values=production
```

Example Example: Describe instances with the specified tag value

The following command describes the instances with a tag with the value **production**, regardless of the tag key.

```
aws ec2 describe-instances \
--filters Name=tag-value,Values=production
```

Example Example: Describe all EC2 resources with the specified tag

The following command describes all EC2 resources with the tag **Stack=Test**.

```
aws ec2 describe-tags \
--filters Name=key,Values=Stack Name=value,Values=Test
```

Adding tags to a resource using CloudFormation

With Amazon EC2 resource types, you specify tags using either a `Tags` or `TagSpecifications` property.

The following examples add the tag **Stack=Production** to [AWS::EC2::Instance](#) using its `Tags` property.

Example Example: Tags in YAML

```
Tags:
- Key: "Stack"
  Value: "Production"
```

Example Example: Tags in JSON

```
"Tags": [
{
```

```
        "Key": "Stack",
        "Value": "Production"
    }
]
```

The following examples add the tag **Stack=Production** to [AWS::EC2::LaunchTemplate](#) [LaunchTemplateData](#) using its TagSpecifications property.

Example Example: TagSpecifications in YAML

```
TagSpecifications:
- ResourceType: "instance"
  Tags:
    - Key: "Stack"
      Value: "Production"
```

Example Example: TagSpecifications in JSON

```
"TagSpecifications": [
{
    "ResourceType": "instance",
    "Tags": [
        {
            "Key": "Stack",
            "Value": "Production"
        }
    ]
}]
```

Amazon EC2 service quotas

Amazon EC2 provides different *resources* that you can use. These resources include images, instances, volumes, and snapshots. When you create your AWS account, we set default quotas (also referred to as limits) on these resources on a per-Region basis. For example, there is a maximum number of instances that you can launch in a Region. So if you were to launch an instance in the US West (Oregon) Region, for example, the request must not cause your usage to exceed your maximum number of instances in that Region.

The Amazon EC2 console provides limit information for the resources managed by the Amazon EC2 and Amazon VPC consoles. You can request an increase for many of these limits. Use the limit information that we provide to manage your AWS infrastructure. Plan to request any limit increases in advance of the time that you'll need them.

For more information, see [Amazon EC2 endpoints and quotas](#) in the [Amazon Web Services General Reference](#). For information about Amazon EBS quotas, see [Amazon EBS quotas \(p. 1188\)](#).

Viewing your current limits

Use the **Limits** page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-Region basis.

To view your current limits

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select a Region.

Ohio ▾	
US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
<hr/>	
Africa (Cape Town)	af-south-1
<hr/>	
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Osaka-Local)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
<hr/>	
Canada (Central)	ca-central-1
<hr/>	
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Milan)	eu-south-1
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
<hr/>	
Middle East (Bahrain)	me-south-1
<hr/>	
South America (São Paulo)	sa-east-1

3. From the navigation pane, choose **Limits**.
4. Locate the resource in the list. You can use the search fields to filter the list by resource name or resource group. The **Current limit** column displays the current maximum for the resource for your account.

Requesting an increase

Use the **Limits** page in the Amazon EC2 console to request an increase in your Amazon EC2 or Amazon VPC resources, on a per-Region basis.

Alternatively, request an increase using Service Quotas. For more information, see [Requesting a quota increase](#) in the *Service Quotas User Guide*.

To request an increase using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select a Region.
3. From the navigation pane, choose **Limits**.
4. Select the resource in the list, and choose **Request limit increase**.
5. Complete the required fields on the limit increase form and choose **Submit**. We'll respond to you using the contact method that you specified.

Limits on email sent using port 25

Amazon EC2 restricts traffic on port 25 of all instances by default. You can request that this restriction be removed. For more information, see [How do I remove the restriction on port 25 from my EC2 instance?](#) in the AWS Knowledge Center.

Amazon EC2 usage reports

AWS provides a free reporting tool called AWS Cost Explorer that enables you to analyze the cost and usage of your EC2 instances and the usage of your Reserved Instances. You can view data up to the last 13 months, and forecast how much you are likely to spend for the next three months. You can use Cost Explorer to see patterns in how much you spend on AWS resources over time, identify areas that need further inquiry, and see trends that you can use to understand your costs. You also can specify time ranges for the data, and view time data by day or by month.

Here's an example of some of the questions that you can answer when using Cost Explorer:

- How much am I spending on instances of each instance type?
- How many instance hours are being used by a particular department?
- How is my instance usage distributed across Availability Zones?
- How is my instance usage distributed across AWS accounts?
- How well am I using my Reserved Instances?
- Are my Reserved Instances helping me save money?

For more information about working with reports in Cost Explorer, including saving reports, see [Analyzing your costs with Cost Explorer](#).

Troubleshooting EC2 instances

The following documentation can help you troubleshoot problems that you might have with your instance.

Contents

- [Troubleshooting instance launch issues \(p. 1244\)](#)
- [Troubleshooting connecting to your instance \(p. 1246\)](#)
- [Troubleshooting stopping your instance \(p. 1254\)](#)
- [Troubleshooting terminating \(shutting down\) your instance \(p. 1255\)](#)
- [Troubleshooting instances with failed status checks \(p. 1256\)](#)
- [Troubleshooting an unreachable instance \(p. 1277\)](#)
- [Booting from the wrong volume \(p. 1280\)](#)
- [Using EC2Rescue for Linux \(p. 1281\)](#)
- [Sending a diagnostic interrupt \(for advanced users\) \(p. 1291\)](#)

For additional help with Windows instances, see [Troubleshooting Windows Instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Troubleshooting instance launch issues

The following issues prevent you from launching an instance.

Launch Issues

- [Instance limit exceeded \(p. 1244\)](#)
- [Insufficient instance capacity \(p. 1245\)](#)
- [Instance terminates immediately \(p. 1245\)](#)

Instance limit exceeded

Description

You get the `InstanceLimitExceeded` error when you try to launch a new instance or restart a stopped instance.

Cause

If you get an `InstanceLimitExceeded` error when you try to launch a new instance or restart a stopped instance, you have reached the limit on the number of instances that you can launch in a Region. When you create your AWS account, we set default limits on the number of instances you can run on a per-Region basis.

Solution

You can request an instance limit increase on a per-region basis. For more information, see [Amazon EC2 service quotas \(p. 1241\)](#).

Insufficient instance capacity

Description

You get the `InsufficientInstanceCapacity` error when you try to launch a new instance or restart a stopped instance.

Cause

If you get an `InsufficientInstanceCapacity` error when you try to launch an instance or restart a stopped instance, AWS does not currently have enough available On-Demand capacity to service your request.

Solution

To resolve the issue, try the following:

- Wait a few minutes and then submit your request again; capacity can shift frequently.
- Submit a new request with a reduced number of instances. For example, if you're making a single request to launch 15 instances, try making 3 requests for 5 instances, or 15 requests for 1 instance instead.
- If you're launching an instance, submit a new request without specifying an Availability Zone.
- If you're launching an instance, submit a new request using a different instance type (which you can resize at a later stage). For more information, see [Changing the instance type \(p. 294\)](#).
- If you are launching instances into a cluster placement group, you can get an insufficient capacity error. For more information, see [Placement group rules and limitations \(p. 872\)](#).
- Try creating an On-Demand Capacity Reservation, which enables you to reserve Amazon EC2 capacity for any duration. For more information, see [On-Demand Capacity Reservations \(p. 474\)](#).
- Try purchasing Reserved Instances, which are a long-term capacity reservation. For more information, see [Amazon EC2 Reserved Instances](#).

Instance terminates immediately

Description

Your instance goes from the pending state to the terminated state immediately after restarting it.

Cause

The following are a few reasons why an instance might immediately terminate:

- You've reached your EBS volume limit.
- An EBS snapshot is corrupt.
- The root EBS volume is encrypted and you do not have permissions to access the KMS key for decryption.
- The instance store-backed AMI that you used to launch the instance is missing a required part (an `image.part.xx` file).

Solution

You can use the Amazon EC2 console or AWS Command Line Interface to get the termination reason.

To get the termination reason using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and select the instance.
3. In the **Description** tab, note the reason next to the **State transition reason** label.

To get the termination reason using the AWS Command Line Interface

1. Use the [describe-instances](#) command and specify the instance ID.

```
aws ec2 describe-instances --instance-id instance_id
```

2. Review the JSON response returned by the command and note the values in the **StateReason** response element.

The following code block shows an example of a **StateReason** response element.

```
"StateReason": {  
    "Message": "Client.VolumeLimitExceeded: Volume limit exceeded",  
    "Code": "Server.InternalError"  
},
```

To address the issue

Take one of the following actions depending on the termination reason you noted:

- If the reason is **Client.VolumeLimitExceeded: Volume limit exceeded**, you have reached your EBS volume limit. For more information, see [Instance volume limits \(p. 1209\)](#). To submit a request to increase your Amazon EBS volume limit, complete the AWS Support Center [Create Case](#) form. For more information, see [Amazon EC2 service quotas \(p. 1241\)](#).
- If the reason is **Client.InternalError: Client error on launch**, that typically indicates that the root volume is encrypted and that you do not have permissions to access the KMS key for decryption. To get permissions to access the required KMS key, add the appropriate KMS permissions to your IAM user. For more information, see [Using Key Policies in AWS KMS](#) in the [AWS Key Management Service Developer Guide](#).

Troubleshooting connecting to your instance

The following information can help you troubleshoot issues with connecting to your instance. For additional help with Windows instances, see [Troubleshooting Windows Instances](#) in the [Amazon EC2 User Guide for Windows Instances](#).

Connection problems and errors

- [Common causes for connection issues \(p. 1247\)](#)
- [Error connecting to your instance: Connection timed out \(p. 1247\)](#)
- [Error: unable to load key ... Expecting: ANY PRIVATE KEY \(p. 1249\)](#)
- [Error: User key not recognized by server \(p. 1250\)](#)
- [Error: Permission denied or connection closed by \[instance\] port 22 \(p. 1251\)](#)
- [Error: Unprotected private key file \(p. 1252\)](#)
- [Error: Private key must begin with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----" \(p. 1253\)](#)

- [Error: Server refused our key or No supported authentication methods available \(p. 1253\)](#)
- [Cannot ping instance \(p. 1253\)](#)
- [Error: Server unexpectedly closed network connection \(p. 1254\)](#)

Common causes for connection issues

We recommend that you begin troubleshooting by checking some common causes for issues connecting to your instance.

Verify the user name for your instance

You can connect to your instance using the user name for your user account or the default user name for the AMI that you used to launch your instance.

- **Get the user name for your user account.**

For more information about how to create a user account, see [Managing user accounts on your Amazon Linux instance \(p. 620\)](#).

- **Get the default user name for the AMI that you used to launch your instance:**

- For Amazon Linux 2 or the Amazon Linux AMI, the user name is `ec2-user`.
- For a CentOS AMI, the user name is `centos`.
- For a Debian AMI, the user name is `admin`.
- For a Fedora AMI, the user name is `ec2-user` or `fedora`.
- For a RHEL AMI, the user name is `ec2-user` or `root`.
- For a SUSE AMI, the user name is `ec2-user` or `root`.
- For an Ubuntu AMI, the user name is `ubuntu`.
- Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.

Verify that your security group rules allow traffic

Make sure your security group rules allow inbound traffic from your public IPv4 address on the proper port. For steps to verify, see [Error connecting to your instance: Connection timed out \(p. 1247\)](#)

Verify that your instance is ready

After you launch an instance, it can take a few minutes for the instance to be ready so that you can connect to it. Check your instance to make sure it is running and has passed its status checks.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. Verify that your **Instance State** is running and your **Status Checks** have passed.

Verify the general prerequisites for connecting to your instance

For more information, see [General prerequisites for connecting to your instance \(p. 563\)](#).

Error connecting to your instance: Connection timed out

If you try to connect to your instance and get an error message `Network error: Connection timed out` or `Error connecting to [instance], reason: -> Connection timed out: connect`, try the following:

- Check your security group rules. You need a security group rule that allows inbound traffic from your public IPv4 address on the proper port.
 1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 2. In the navigation pane, choose **Instances**, and then select your instance.
 3. In the **Description** tab at the bottom of the console page, next to **Security groups**, select **view inbound rules** to display the list of rules that are in effect for the selected instance.
 4. For Linux instances: When you select **view inbound rules**, a window will appear that displays the port(s) to which traffic is allowed. Verify that there is a rule that allows traffic from your computer to port 22 (SSH).

For Windows instances: When you select **view inbound rules**, a window will appear that displays the port(s) to which traffic is allowed. Verify that there is a rule that allows traffic from your computer to port 3389 (RDP).

Each time you restart your instance, a new IP address (and host name) will be assigned. If your security group has a rule that allows inbound traffic from a single IP address, this address may not be static if your computer is on a corporate network or if you are connecting through an internet service provider (ISP). Instead, specify the range of IP addresses used by client computers. If your security group does not have a rule that allows inbound traffic as described in the previous step, add a rule to your security group. For more information, see [Authorizing Network Access to Your Instances \(p. 983\)](#).

For more information about Security Group rules, see [Security Group Rules](#) in the *Amazon VPC User Guide*.

- Check the route table for the subnet. You need a route that sends all traffic destined outside the VPC to the internet gateway for the VPC.
 1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 2. In the navigation pane, choose **Instances**, and then select your instance.
 3. In the **Description** tab, write down the values of **VPC ID** and **Subnet ID**.
 4. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
 5. In the navigation pane, choose **Internet Gateways**. Verify that there is an internet gateway attached to your VPC. Otherwise, choose **Create Internet Gateway** to create an internet gateway. Select the internet gateway, and then choose **Attach to VPC** and follow the directions to attach it to your VPC.
 6. In the navigation pane, choose **Subnets**, and then select your subnet.
 7. On the **Route Table** tab, verify that there is a route with `0.0.0.0/0` as the destination and the internet gateway for your VPC as the target. If you're connecting to your instance using its IPv6 address, verify that there is a route for all IPv6 traffic (`::/0`) that points to the internet gateway. Otherwise, do the following:
 - a. Choose the ID of the route table (rtb-xxxxxxx) to navigate to the route table.
 - b. On the **Routes** tab, choose **Edit routes**. Choose **Add route**, use `0.0.0.0/0` as the destination and the internet gateway as the target. For IPv6, choose **Add route**, use `::/0` as the destination and the internet gateway as the target.
 - c. Choose **Save routes**.
- Check the network access control list (ACL) for the subnet. The network ACLs must allow inbound and outbound traffic from your local IP address on the proper port. The default network ACL allows all inbound and outbound traffic.
 1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
 2. In the navigation pane, choose **Subnets** and select your subnet.
 3. On the **Description** tab, find **Network ACL**, and choose its ID (acl-xxxxxxx).

4. Select the network ACL. For **Inbound Rules**, verify that the rules allow traffic from your computer. Otherwise, delete or modify the rule that is blocking traffic from your computer.
 5. For **Outbound Rules**, verify that the rules allow traffic to your computer. Otherwise, delete or modify the rule that is blocking traffic to your computer.
- If your computer is on a corporate network, ask your network administrator whether the internal firewall allows inbound and outbound traffic from your computer on port 22 (for Linux instances) or port 3389 (for Windows instances).

If you have a firewall on your computer, verify that it allows inbound and outbound traffic from your computer on port 22 (for Linux instances) or port 3389 (for Windows instances).

- Check that your instance has a public IPv4 address. If not, you can associate an Elastic IP address with your instance. For more information, see [Elastic IP addresses \(p. 781\)](#).
- Check the CPU load on your instance; the server may be overloaded. AWS automatically provides data such as Amazon CloudWatch metrics and instance status, which you can use to see how much CPU load is on your instance and, if necessary, adjust how your loads are handled. For more information, see [Monitoring your instances using CloudWatch \(p. 715\)](#).
 - If your load is variable, you can automatically scale your instances up or down using [Auto Scaling](#) and [Elastic Load Balancing](#).
 - If your load is steadily growing, you can move to a larger instance type. For more information, see [Changing the instance type \(p. 294\)](#).

To connect to your instance using an IPv6 address, check the following:

- Your subnet must be associated with a route table that has a route for IPv6 traffic (`::/0`) to an internet gateway.
- Your security group rules must allow inbound traffic from your local IPv6 address on the proper port (22 for Linux and 3389 for Windows).
- Your network ACL rules must allow inbound and outbound IPv6 traffic.
- If you launched your instance from an older AMI, it may not be configured for DHCPv6 (IPv6 addresses are not automatically recognized on the network interface). For more information, see [Configure IPv6 on Your Instances](#) in the *Amazon VPC User Guide*.
- Your local computer must have an IPv6 address, and must be configured to use IPv6.

Error: unable to load key ... Expecting: ANY PRIVATE KEY

If you try to connect to your instance and get the error message, `unable to load key ...` `Expecting: ANY PRIVATE KEY`, the file in which the private key is stored is incorrectly configured. If the private key file ends in `.pem`, it might still be incorrectly configured. A possible cause for an incorrectly configured private key file is a missing certificate.

If the private key file is incorrectly configured, follow these steps to resolve the error

1. Create a new key pair. For more information, see [Option 1: Create a key pair using Amazon EC2 \(p. 986\)](#).
2. Add the new key pair to your instance. For more information, see [Connecting to your Linux instance if you lose your private key \(p. 994\)](#).
3. Connect to your instance using the new key pair.

Error: User key not recognized by server

If you use SSH to connect to your instance

- Use ssh -vvv to get triple verbose debugging information while connecting:

```
ssh -vvv -i path/my-key-pair.pem my-instance-user-
name@ec2-203-0-113-25.compute-1.amazonaws.com
```

The following sample output demonstrates what you might see if you were trying to connect to your instance with a key that was not recognized by the server:

```
open/ANT/myusername/.ssh/known_hosts).
debug2: bits set: 504/1024
debug1: ssh_rsa_verify: signature correct
debug2: kex_derive_keys
debug2: set_newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug2: set_newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: Roaming not allowed by server
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug2: service_accept: ssh-userauth
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug2: key: boguspem.pem ((nil))
debug1: Authentications that can continue: publickey
debug3: start over, passed a different list publickey
debug3: preferred gssapi-keyex,gssapi-with-mic,publickey,keyboard-interactive,password
debug3: authmethod_lookup publickey
debug3: remaining preferred: keyboard-interactive,password
debug3: authmethod_is_enabled publickey
debug1: Next authentication method: publickey
debug1: Trying private key: boguspem.pem
debug1: read PEM private key done: type RSA
debug3: sign_and_send_pubkey: RSA 9c:4c:bc:0c:d0:5c:c7:92:6c:8e:9b:16:e4:43:d8:b2
debug2: we sent a publickey packet, wait for reply
debug1: Authentications that can continue: publickey
debug2: we did not send a packet, disable method
debug1: No more authentication methods to try.
Permission denied (publickey).
```

If you use PuTTY to connect to your instance

- Verify that your private key (.pem) file has been converted to the format recognized by PuTTY (.ppk). For more information about converting your private key, see [Connecting to your Linux instance from Windows using PuTTY \(p. 578\)](#).

Note

In PuTTYgen, load your private key file and select **Save Private Key** rather than **Generate**.

- Verify that you are connecting with the appropriate user name for your AMI. Enter the user name in the **Host name** box in the **PuTTY Configuration** window.
 - For Amazon Linux 2 or the Amazon Linux AMI, the user name is `ec2-user`.
 - For a CentOS AMI, the user name is `centos`.
 - For a Debian AMI, the user name is `admin`.
 - For a Fedora AMI, the user name is `ec2-user` or `fedora`.
 - For a RHEL AMI, the user name is `ec2-user` or `root`.

- For a SUSE AMI, the user name is `ec2-user` or `root`.
- For an Ubuntu AMI, the user name is `ubuntu`.
- Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.
- Verify that you have an inbound security group rule to allow inbound traffic to the appropriate port. For more information, see [Authorizing Network Access to Your Instances \(p. 983\)](#).

Error: Permission denied or connection closed by [instance] port 22

If you connect to your instance using SSH and get any of the following errors, `Host key not found in [directory]`, `Permission denied (publickey)`, `Authentication failed, permission denied`, or `Connection closed by [instance] port 22`, verify that you are connecting with the appropriate user name for your AMI *and* that you have specified the proper private key (`.pem`) file for your instance.

The appropriate user names are as follows:

- For Amazon Linux 2 or the Amazon Linux AMI, the user name is `ec2-user`.
- For a CentOS AMI, the user name is `centos`.
- For a Debian AMI, the user name is `admin`.
- For a Fedora AMI, the user name is `ec2-user` or `fedora`.
- For a RHEL AMI, the user name is `ec2-user` or `root`.
- For a SUSE AMI, the user name is `ec2-user` or `root`.
- For an Ubuntu AMI, the user name is `ubuntu`.
- Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.

For example, to use an SSH client to connect to an Amazon Linux instance, use the following command:

```
ssh -i /path/my-key-pair.pem my-instance-user-name@ec2-203-0-113-25.compute-1.amazonaws.com
```

Confirm that you are using the private key file that corresponds to the key pair that you selected when you launched the instance.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select your instance. In the **Description** tab, verify the value of **Key pair name**.
3. If you did not specify a key pair when you launched the instance, you can terminate the instance and launch a new instance, ensuring that you specify a key pair. If this is an instance that you have been using but you no longer have the `.pem` file for your key pair, you can replace the key pair with a new one. For more information, see [Connecting to your Linux instance if you lose your private key \(p. 994\)](#).

If you generated your own key pair, ensure that your key generator is set up to create RSA keys. DSA keys are not accepted.

If you get a `Permission denied (publickey)` error and none of the above applies (for example, you were able to connect previously), the permissions on the home directory of your instance may have been changed. Permissions for `/home/my-instance-user-name/.ssh/authorized_keys` must be limited to the owner only.

To verify the permissions on your instance

1. Stop your instance and detach the root volume. For more information, see [Stop and start your instance \(p. 587\)](#) and [Detaching an Amazon EBS volume from a Linux instance \(p. 1058\)](#).
2. Launch a temporary instance in the same Availability Zone as your current instance (use a similar or the same AMI as you used for your current instance), and attach the root volume to the temporary instance. For more information, see [Attaching an Amazon EBS volume to an instance \(p. 1042\)](#).
3. Connect to the temporary instance, create a mount point, and mount the volume that you attached. For more information, see [Making an Amazon EBS volume available for use on Linux \(p. 1046\)](#).
4. From the temporary instance, check the permissions of the `/home/my-instance-user-name/` directory of the attached volume. If necessary, adjust the permissions as follows:

```
[ec2-user ~]$ chmod 600 mount_point/home/my-instance-user-name/.ssh/authorized_keys
```

```
[ec2-user ~]$ chmod 700 mount_point/home/my-instance-user-name/.ssh
```

```
[ec2-user ~]$ chmod 700 mount_point/home/my-instance-user-name
```

5. Unmount the volume, detach it from the temporary instance, and re-attach it to the original instance. Ensure that you specify the correct device name for the root volume; for example, `/dev/xvda`.
6. Start your instance. If you no longer require the temporary instance, you can terminate it.

Error: Unprotected private key file

Your private key file must be protected from read and write operations from any other users. If your private key can be read or written to by anyone but you, then SSH ignores your key and you see the following warning message below.

```
@@@@@@@  
@     WARNING: UNPROTECTED PRIVATE KEY FILE!          @  
@  
Permissions 0777 for '.ssh/my_private_key.pem' are too open.  
It is required that your private key files are NOT accessible by others.  
This private key will be ignored.  
bad permissions: ignore key: .ssh/my_private_key.pem  
Permission denied (publickey).
```

If you see a similar message when you try to log in to your instance, examine the first line of the error message to verify that you are using the correct public key for your instance. The above example uses the private key `.ssh/my_private_key.pem` with file permissions of 0777, which allow anyone to read or write to this file. This permission level is very insecure, and so SSH ignores this key. To fix the error, execute the following command, substituting the path for your private key file.

```
[ec2-user ~]$ chmod 0400 .ssh/my_private_key.pem
```

Error: Private key must begin with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----"

If you use a third-party tool, such as **ssh-keygen**, to create an RSA key pair, it generates the private key in the OpenSSH key format. When you connect to your instance, if you use the private key in the OpenSSH format to decrypt the password, you'll get the error **Private key must begin with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----".**

To resolve the error, the private key must be in the PEM format. Use the following command to create the private key in the PEM format:

```
ssh-keygen -m PEM
```

Error: Server refused our key or No supported authentication methods available

If you use PuTTY to connect to your instance and get either of the following errors, Error: Server refused our key or Error: No supported authentication methods available, verify that you are connecting with the appropriate user name for your AMI. Type the user name in **User name** in the **PuTTY Configuration** window.

The appropriate user names are as follows:

- For Amazon Linux 2 or the Amazon Linux AMI, the user name is `ec2-user`.
- For a CentOS AMI, the user name is `centos`.
- For a Debian AMI, the user name is `admin`.
- For a Fedora AMI, the user name is `ec2-user` or `fedora`.
- For a RHEL AMI, the user name is `ec2-user` or `root`.
- For a SUSE AMI, the user name is `ec2-user` or `root`.
- For an Ubuntu AMI, the user name is `ubuntu`.
- Otherwise, if `ec2-user` and `root` don't work, check with the AMI provider.

You should also verify that your private key (.pem) file has been correctly converted to the format recognized by PuTTY (.ppk). For more information about converting your private key, see [Connecting to your Linux instance from Windows using PuTTY \(p. 578\)](#).

Cannot ping instance

The `ping` command is a type of ICMP traffic — if you are unable to ping your instance, ensure that your inbound security group rules allow ICMP traffic for the Echo Request message from all sources, or from the computer or instance from which you are issuing the command.

If you are unable to issue a `ping` command from your instance, ensure that your outbound security group rules allow ICMP traffic for the Echo Request message to all destinations, or to the host that you are attempting to ping.

`Ping` commands can also be blocked by a firewall or time out due to network latency or hardware issues. You should consult your local network or system administrator for help with further troubleshooting.

Error: Server unexpectedly closed network connection

If you are connecting to your instance with PuTTY and you receive the error "Server unexpectedly closed network connection," verify that you have enabled keepalives on the Connection page of the PuTTY Configuration to avoid being disconnected. Some servers disconnect clients when they do not receive any data within a specified period of time. Set the Seconds between keepalives to 59 seconds.

If you still experience issues after enabling keepalives, try to disable Nagle's algorithm on the Connection page of the PuTTY Configuration.

Troubleshooting stopping your instance

If you have stopped your Amazon EBS-backed instance and it appears stuck in the `stopping` state, there may be an issue with the underlying host computer.

There is no cost for any instance usage while an instance is not in the `running` state.

Force the instance to stop using either the console or the AWS CLI.

- To force the instance to stop using the console, select the stuck instance, and choose **Actions, Instance State, Stop, and Yes, Forcefully Stop**.
- To force the instance to stop using the AWS CLI, use the `stop-instances` command and the `--force` option as follows:

```
aws ec2 stop-instances --instance-ids i-0123ab456c789d01e --force
```

If, after 10 minutes, the instance has not stopped, post a request for help in the [Amazon EC2 forum](#). To help expedite a resolution, include the instance ID, and describe the steps that you've already taken. Alternatively, if you have a support plan, create a technical support case in the [Support Center](#).

Creating a replacement instance

To attempt to resolve the problem while you are waiting for assistance from the [Amazon EC2 forum](#) or the [Support Center](#), create a replacement instance. Create an AMI of the stuck instance, and launch a new instance using the new AMI.

To create a replacement instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select the stuck instance.
3. Choose **Actions, Image, Create Image**.
4. In the **Create Image** dialog box, fill in the following fields, and then choose **Create Image**:
 - a. Specify a name and description for the AMI.
 - b. Choose **No reboot**.

For more information, see [Creating a Linux AMI from an instance \(p. 124\)](#).

5. Launch a new instance from the AMI and verify that the new instance is working.
6. Select the stuck instance, and choose **Actions, Instance State, Terminate**. If the instance also gets stuck terminating, Amazon EC2 automatically forces it to terminate within a few hours.

To create a replacement instance using the CLI

1. Create an AMI from the stuck instance using the [create-image](#) (AWS CLI) command and the --no-reboot option as follows:

```
aws ec2 create-image --instance-id i-0123ab456c789d01e --name "AMI" --description "AMI for replacement instance" --no-reboot
```

2. Launch a new instance from the AMI using the [run-instances](#) (AWS CLI) command as follows:

```
aws ec2 run-instances --image-id ami-1a2b3c4d --count 1 --instance-type c3.large --key-name MyKeyPair --security-groups MySecurityGroup
```

3. Verify that the new instance is working.
4. Terminate the stuck instance using the [terminate-instances](#) (AWS CLI) command as follows:

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

If you are unable to create an AMI from the instance as described in the previous procedures, you can set up a replacement instance as follows:

(Alternate) To create a replacement instance using the console

1. Select the instance and choose **Description**, **Block devices**. Select each volume and write down its volume ID. Be sure to note which volume is the root volume.
2. In the navigation pane, choose **Volumes**. Select each volume for the instance, and choose **Actions**, **Create Snapshot**.
3. In the navigation pane, choose **Snapshots**. Select the snapshot that you just created, and choose **Actions**, **Create Volume**.
4. Launch an instance with the same operating system as the stuck instance. Note the volume ID and device name of its root volume.
5. In the navigation pane, choose **Instances**, select the instance that you just launched, choose **Actions**, **Instance State**, and then choose **Stop**.
6. In the navigation pane, choose **Volumes**, select the root volume of the stopped instance, and choose **Actions**, **Detach Volume**.
7. Select the root volume that you created from the stuck instance, choose **Actions**, **Attach Volume**, and attach it to the new instance as its root volume (using the device name that you wrote down). Attach any additional non-root volumes to the instance.
8. In the navigation pane, choose **Instances** and select the replacement instance. Choose **Actions**, **Instance State**, **Start**. Verify that the instance is working.
9. Select the stuck instance, choose **Actions**, **Instance State**, **Terminate**. If the instance also gets stuck terminating, Amazon EC2 automatically forces it to terminate within a few hours.

Troubleshooting terminating (shutting down) your instance

You are not billed for any instance usage while an instance is not in the `running` state. In other words, when you terminate an instance, you stop incurring charges for that instance as soon as its state changes to `shutting-down`.

Delayed instance termination

If your instance remains in the shutting-down state longer than a few minutes, it might be delayed due to shutdown scripts being run by the instance.

Another possible cause is a problem with the underlying host computer. If your instance remains in the shutting-down state for several hours, Amazon EC2 treats it as a stuck instance and forcibly terminates it.

If it appears that your instance is stuck terminating and it has been longer than several hours, post a request for help to the [Amazon EC2 forum](#). To help expedite a resolution, include the instance ID and describe the steps that you've already taken. Alternatively, if you have a support plan, create a technical support case in the [Support Center](#).

Terminated instance still displayed

After you terminate an instance, it remains visible for a short while before being deleted. The state shows as terminated. If the entry is not deleted after several hours, contact Support.

Instances automatically launched or terminated

Generally, the following behaviors mean that you've used Amazon EC2 Auto Scaling or EC2 Fleet to scale your computing resources automatically based on criteria that you've defined:

- You terminate an instance and a new instance launches automatically.
- You launch an instance and one of your instances terminates automatically.
- You stop an instance and it terminates and a new instance launches automatically.

To stop automatic scaling, see the [Amazon EC2 Auto Scaling User Guide](#) or [Launching instances using an EC2 Fleet \(p. 524\)](#).

Troubleshooting instances with failed status checks

The following information can help you troubleshoot issues if your instance fails a status check. First determine whether your applications are exhibiting any problems. If you verify that the instance is not running your applications as expected, review the status check information and the system logs.

Contents

- [Review status check information \(p. 1257\)](#)
- [Retrieve the system logs \(p. 1257\)](#)
- [Troubleshooting system log errors for Linux-based instances \(p. 1258\)](#)
- [Out of memory: kill process \(p. 1259\)](#)
- [ERROR: mmu_update failed \(Memory management update failed\) \(p. 1260\)](#)
- [I/O error \(block device failure\) \(p. 1260\)](#)
- [I/O ERROR: neither local nor remote disk \(Broken distributed block device\) \(p. 1261\)](#)
- [request_module: runaway loop modprobe \(Looping legacy kernel modprobe on older Linux versions\) \(p. 1262\)](#)
- ["FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" \(Kernel and AMI mismatch\) \(p. 1263\)](#)

- "FATAL: Could not load /lib/modules" or "BusyBox" (Missing kernel modules) (p. 1263)
- ERROR Invalid kernel (EC2 incompatible kernel) (p. 1265)
- fsck: No such file or directory while trying to open... (File system not found) (p. 1266)
- General error mounting filesystems (failed mount) (p. 1267)
- VFS: Unable to mount root fs on unknown-block (Root filesystem mismatch) (p. 1269)
- Error: Unable to determine major/minor number of root device... (Root file system/device mismatch) (p. 1270)
- XENBUS: Device with no driver... (p. 1271)
- ... days without being checked, check forced (File system check required) (p. 1271)
- fsck died with exit status... (Missing device) (p. 1272)
- GRUB prompt (grubdom>) (p. 1273)
- Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address) (p. 1275)
- Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration) (p. 1276)
- XENBUS: Timeout connecting to devices (Xenbus timeout) (p. 1277)

Review status check information

To investigate impaired instances using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. In the details pane, choose **Status Checks** to see the individual results for all **System Status Checks** and **Instance Status Checks**.

If a system status check has failed, you can try one of the following options:

- Create an instance recovery alarm. For more information, see [Create alarms that stop, terminate, reboot, or recover an instance \(p. 738\)](#).
- If you changed the instance type to an instance built on the [Nitro System \(p. 205\)](#), status checks fail if you migrated from an instance that does not have the required ENA and NVMe drivers. For more information, see [Compatibility for resizing instances \(p. 295\)](#).
- For an instance using an Amazon EBS-backed AMI, stop and restart the instance.
- For an instance using an instance-store backed AMI, terminate the instance and launch a replacement.
- Wait for Amazon EC2 to resolve the issue.
- Post your issue to the [Amazon EC2 forum](#).
- If your instance is in an Auto Scaling group, the Amazon EC2 Auto Scaling service automatically launches a replacement instance. For more information, see [Health Checks for Auto Scaling Instances](#) in the [Amazon EC2 Auto Scaling User Guide](#).
- Retrieve the system log and look for errors.

Retrieve the system logs

If an instance status check fails, you can reboot the instance and retrieve the system logs. The logs may reveal an error that can help you troubleshoot the issue. Rebooting clears unnecessary information from the logs.

To reboot an instance and retrieve the system log

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**, and select your instance.
3. Choose **Actions, Instance State, Reboot**. It may take a few minutes for your instance to reboot.
4. Verify that the problem still exists; in some cases, rebooting may resolve the problem.
5. When the instance is in the `running` state, choose **Actions, Instance Settings, Get System Log**.
6. Review the log that appears on the screen, and use the list of known system log error statements below to troubleshoot your issue.
7. If your experience differs from our check results, or if you are having an issue with your instance that our checks did not detect, choose **Submit feedback** on the **Status Checks** tab to help us improve our detection tests.
8. If your issue is not resolved, you can post your issue to the [Amazon EC2 forum](#).

Troubleshooting system log errors for Linux-based instances

For Linux-based instances that have failed an instance status check, such as the instance reachability check, verify that you followed the steps above to retrieve the system log. The following list contains some common system log errors and suggested actions you can take to resolve the issue for each error.

Memory Errors

- [Out of memory: kill process \(p. 1259\)](#)
- [ERROR: mmu_update failed \(Memory management update failed\) \(p. 1260\)](#)

Device Errors

- [I/O error \(block device failure\) \(p. 1260\)](#)
- [I/O ERROR: neither local nor remote disk \(Broken distributed block device\) \(p. 1261\)](#)

Kernel Errors

- [request_module: runaway loop modprobe \(Looping legacy kernel modprobe on older Linux versions\) \(p. 1262\)](#)
- ["FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" \(Kernel and AMI mismatch\) \(p. 1263\)](#)
- ["FATAL: Could not load /lib/modules" or "BusyBox" \(Missing kernel modules\) \(p. 1263\)](#)
- [ERROR Invalid kernel \(EC2 incompatible kernel\) \(p. 1265\)](#)

File System Errors

- [fsck: No such file or directory while trying to open... \(File system not found\) \(p. 1266\)](#)
- [General error mounting filesystems \(failed mount\) \(p. 1267\)](#)
- [VFS: Unable to mount root fs on unknown-block \(Root filesystem mismatch\) \(p. 1269\)](#)
- [Error: Unable to determine major/minor number of root device... \(Root file system/device mismatch\) \(p. 1270\)](#)
- [XENBUS: Device with no driver... \(p. 1271\)](#)
- [... days without being checked, check forced \(File system check required\) \(p. 1271\)](#)

- [fsck died with exit status... \(Missing device\) \(p. 1272\)](#)

Operating System Errors

- [GRUB prompt \(grubdom>\) \(p. 1273\)](#)
- [Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. \(Hard-coded MAC address\) \(p. 1275\)](#)
- [Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. \(SELinux misconfiguration\) \(p. 1276\)](#)
- [XENBUS: Timeout connecting to devices \(Xenbus timeout\) \(p. 1277\)](#)

Out of memory: kill process

An out-of-memory error is indicated by a system log entry similar to the one shown below.

```
[115879.769795] Out of memory: kill process 20273 (httpd) score 1285879
or a child
[115879.769795] Killed process 1917 (php-cgi) vsz:467184kB, anon-
rss:101196kB, file-rss:204kB
```

Potential cause

Exhausted memory

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Do one of the following:</p> <ul style="list-style-type: none">• Stop the instance, and modify the instance to use a different instance type, and start the instance again. For example, a larger or a memory-optimized instance type.• Reboot the instance to return it to a non-impaired status. The problem will probably occur again unless you change the instance type.
Instance store-backed	<p>Do one of the following:</p> <ul style="list-style-type: none">• Terminate the instance and launch a new instance, specifying a different instance type. For example, a larger or a memory-optimized instance type.• Reboot the instance to return it to an unimpaired status. The problem will probably occur again unless you change the instance type.

ERROR: mmu_update failed (Memory management update failed)

Memory management update failures are indicated by a system log entry similar to the following:

```
...
Press `ESC' to enter the menu... 0  [H[J  Booting 'Amazon Linux 2011.09
(2.6.35.14-95.38.amzn1.i686)'

root (hd0)

Filesystem type is ext2fs, using whole disk

kernel /boot/vmlinuz-2.6.35.14-95.38.amzn1.i686 root=LABEL=/ console=hvc0 LANG=
en_US.UTF-8 KEYTABLE=us

initrd /boot/initramfs-2.6.35.14-95.38.amzn1.i686.img

ERROR: mmu_update failed with rc=-22
```

Potential cause

Issue with Amazon Linux

Suggested action

Post your issue to the [Developer Forums](#) or contact [AWS Support](#).

I/O error (block device failure)

An input/output error is indicated by a system log entry similar to the following example:

```
[9943662.053217] end_request: I/O error, dev sde, sector 52428288
[9943664.191262] end_request: I/O error, dev sde, sector 52428168
[9943664.191285] Buffer I/O error on device md0, logical block 209713024
[9943664.191297] Buffer I/O error on device md0, logical block 209713025
[9943664.191304] Buffer I/O error on device md0, logical block 209713026
[9943664.191310] Buffer I/O error on device md0, logical block 209713027
[9943664.191317] Buffer I/O error on device md0, logical block 209713028
[9943664.191324] Buffer I/O error on device md0, logical block 209713029
[9943664.191332] Buffer I/O error on device md0, logical block 209713030
[9943664.191339] Buffer I/O error on device md0, logical block 209713031
[9943664.191581] end_request: I/O error, dev sde, sector 52428280
[9943664.191590] Buffer I/O error on device md0, logical block 209713136
[9943664.191597] Buffer I/O error on device md0, logical block 209713137
[9943664.191767] end_request: I/O error, dev sde, sector 52428288
[9943664.191970] end_request: I/O error, dev sde, sector 52428288
[9943664.192143] end_request: I/O error, dev sde, sector 52428288
[9943664.192949] end_request: I/O error, dev sde, sector 52428288
[9943664.193112] end_request: I/O error, dev sde, sector 52428288
[9943664.193266] end_request: I/O error, dev sde, sector 52428288
...
```

Potential causes

Instance type	Potential cause
Amazon EBS-backed	A failed Amazon EBS volume
Instance store-backed	A failed physical drive

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Stop the instance.2. Detach the volume.3. Attempt to recover the volume. <p>Note It's good practice to snapshot your Amazon EBS volumes often. This dramatically decreases the risk of data loss as a result of failure.</p> <ol style="list-style-type: none">4. Re-attach the volume to the instance.5. Start the instance.
Instance store-backed	<p>Terminate the instance and launch a new instance.</p> <p>Note Data cannot be recovered. Recover from backups.</p> <p>Note It's a good practice to use either Amazon S3 or Amazon EBS for backups. Instance store volumes are directly tied to single host and single disk failures.</p>

I/O ERROR: neither local nor remote disk (Broken distributed block device)

An input/output error on the device is indicated by a system log entry similar to the following example:

```
...
block drbd1: Local IO failed in request_timer_fn. Detaching...
Aborting journal on device drbd1-8.
block drbd1: IO ERROR: neither local nor remote disk
Buffer I/O error on device drbd1, logical block 557056
lost page write due to I/O error on drbd1
```

JBD2: I/O error detected when updating journal superblock for drbd1-8.

Potential causes

Instance type	Potential cause
Amazon EBS-backed	A failed Amazon EBS volume
Instance store-backed	A failed physical drive

Suggested action

Terminate the instance and launch a new instance.

For an Amazon EBS-backed instance you can recover data from a recent snapshot by creating an image from it. Any data added after the snapshot cannot be recovered.

request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)

This condition is indicated by a system log similar to the one shown below. Using an unstable or old Linux kernel (for example, 2.6.16-xenU) can cause an interminable loop condition at startup.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1  
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007  
  
BIOS-provided physical RAM map:  
  
Xen: 0000000000000000 - 000000026700000 (usable)  
  
OMB HIGHMEM available.  
...  
  
request_module: runaway loop modprobe binfmt-464c  
request_module: runaway loop modprobe binfmt-464c
```

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use a newer kernel, either GRUB-based or static, using one of the following options: Option 1: Terminate the instance and launch a new instance, specifying the <code>-kernel</code> and <code>-ramdisk</code> parameters.

For this instance type	Do this
	Option 2: 1. Stop the instance. 2. Modify the kernel and ramdisk attributes to use a newer kernel. 3. Start the instance.
Instance store-backed	Terminate the instance and launch a new instance, specifying the <code>-kernel</code> and <code>-ramdisk</code> parameters.

"FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" (Kernel and AMI mismatch)

This condition is indicated by a system log similar to the one shown below.

```

Linux version 2.6.16.33-xenU (root@dom0-0-50-45-1-a4-ee.z-2.aes0.internal)
(gcc version 4.1.1 20070105 (Red Hat 4.1.1-52)) #2 SMP Wed Aug 15 17:27:36 SAST 2007
...
FATAL: kernel too old
Kernel panic - not syncing: Attempted to kill init!
  
```

Potential causes

Incompatible kernel and userland

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Modify the configuration to use a newer kernel. 3. Start the instance.
Instance store-backed	Use the following procedure: 1. Create an AMI that uses a newer kernel. 2. Terminate the instance. 3. Start a new instance from the AMI you created.

"FATAL: Could not load /lib/modules" or "BusyBox" (Missing kernel modules)

This condition is indicated by a system log similar to the one shown below.

Amazon Elastic Compute Cloud
User Guide for Linux Instances
"FATAL: Could not load /lib/modules"
or "BusyBox" (Missing kernel modules)

```

[    0.370415] Freeing unused kernel memory: 1716k freed
Loading, please wait...
WARNING: Couldn't open directory /lib/modules/2.6.34-4-virtual: No such file or directory
FATAL: Could not open /lib/modules/2.6.34-4-virtual/modules.dep.temp for writing: No such
      file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or directory
      Couldn't get a file descriptor referring to the console
Begin: Loading essential drivers... ...
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or directory
Done.
Begin: Running /scripts/init-premount ...
Done.
Begin: Mounting root file system... ...
Begin: Running /scripts/local-top ...
Done.
Begin: Waiting for root file system... ...
Done.
Gave up waiting for root device. Common problems:
 - Boot args (cat /proc/cmdline)
   - Check rootdelay= (did the system wait long enough?)
   - Check root= (did the system wait for the right device?)
 - Missing modules (cat /proc/modules; ls /dev)
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or directory
ALERT! /dev/sda1 does not exist. Dropping to a shell!

BusyBox v1.13.3 (Ubuntu 1:1.13.3-1ubuntu5) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs)

```

Potential causes

One or more of the following conditions can cause this problem:

- Missing ramdisk
- Missing correct modules from ramdisk
- Amazon EBS root volume not correctly attached as /dev/sda1

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none"> 1. Select corrected ramdisk for the Amazon EBS volume. 2. Stop the instance. 3. Detach the volume and repair it. 4. Attach the volume to the instance. 5. Start the instance. 6. Modify the AMI to use the corrected ramdisk.
Instance store-backed	<p>Use the following procedure:</p>

For this instance type	Do this
	<ol style="list-style-type: none">1. Terminate the instance and launch a new instance with the correct ramdisk.2. Create a new AMI with the correct ramdisk.

ERROR Invalid kernel (EC2 incompatible kernel)

This condition is indicated by a system log similar to the one shown below.

```
...
root (hd0)

Filesystem type is ext2fs, using whole disk
kernel /vmlinuz root=/dev/sda1 ro
initrd /initrd.img

ERROR Invalid kernel: elf_xen_note_check: ERROR: Will only load images
built for the generic loader or Linux images
xc_dom_parse_image returned -1

Error 9: Unknown boot failure

Booting 'Fallback'

root (hd0)

Filesystem type is ext2fs, using whole disk
kernel /vmlinuz.old root=/dev/sda1 ro

Error 15: File not found
```

Potential causes

One or both of the following conditions can cause this problem:

- Supplied kernel is not supported by GRUB
- Fallback kernel does not exist

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Stop the instance.2. Replace with working kernel.3. Install a fallback kernel.4. Modify the AMI by correcting the kernel.
Instance store-backed	Use the following procedure:

For this instance type	Do this
	<ol style="list-style-type: none">1. Terminate the instance and launch a new instance with the correct kernel.2. Create an AMI with the correct kernel.3. (Optional) Seek technical assistance for data recovery using AWS Support.

fsck: No such file or directory while trying to open... (File system not found)

This condition is indicated by a system log similar to the one shown below.

```
Welcome to Fedora
Press 'I' to enter interactive startup.
Setting clock : Wed Oct 26 05:52:05 EDT 2011 [ OK ]

Starting udev: [ OK ]

Setting hostname localhost: [ OK ]

No devices found
Setting up Logical Volume Management: File descriptor 7 left open
  No volume groups found
[ OK ]

Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1: clean, 82081/1310720 files, 2141116/2621440 blocks
[/sbin/fsck.ext3 (1) -- /mnt/dbbackups] fsck.ext3 -a /dev/sdh
fsck.ext3: No such file or directory while trying to open /dev/sdh

/dev/sdh:
The superblock could not be read or does not describe a correct ext2
filesystem. If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
  e2fsck -b 8193 <device>

[FAILED]

*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue):
```

Potential causes

- A bug exists in ramdisk filesystem definitions /etc/fstab
- Misconfigured filesystem definitions in /etc/fstab
- Missing/failed drive

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Stop the instance, detach the root volume, repair/modify /etc/fstab the volume, attach the volume to the instance, and start the instance.2. Fix ramdisk to include modified /etc/fstab (if applicable).3. Modify the AMI to use a newer ramdisk. <p>The sixth field in the fstab defines availability requirements of the mount – a nonzero value implies that an fsck will be done on that volume and <i>must</i> succeed. Using this field can be problematic in Amazon EC2 because a failure typically results in an interactive console prompt that is not currently available in Amazon EC2. Use care with this feature and read the Linux man page for fstab.</p>
Instance store-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Terminate the instance and launch a new instance.2. Detach any errant Amazon EBS volumes and the reboot instance.3. (Optional) Seek technical assistance for data recovery using AWS Support.

General error mounting filesystems (failed mount)

This condition is indicated by a system log similar to the one shown below.

```
Loading xenblk.ko module
xen-vbd: registered block device major 8

Loading ehci-hcd.ko module
Loading ohci-hcd.ko module
Loading uhci-hcd.ko module
USB Universal Host Controller Interface driver v3.0

Loading mbcache.ko module
Loading jbd.ko module
Loading ext3.ko module
Creating root device.
Mounting root filesystem.
kjournald starting. Commit interval 5 seconds

EXT3-fs: mounted filesystem with ordered data mode.

Setting up other filesystems.
Setting up new root fs
```

```
no fstab.sys, mounting internal defaults
Switching to new root and running init.
unmounting old /dev
unmounting old /proc
unmounting old /sys
mountall:/proc: unable to mount: Device or resource busy
mountall:/proc/self/mountinfo: No such file or directory
mountall: root filesystem isn't mounted
init: mountall main process (221) terminated with status 1

General error mounting filesystems.
A maintenance shell will now be started.
CONTROL-D will terminate this shell and re-try.
Press enter for maintenance
(or type Control-D to continue):
```

Potential causes

Instance type	Potential cause
Amazon EBS-backed	<ul style="list-style-type: none">Detached or failed Amazon EBS volume.Corrupted filesystem.Mismatched ramdisk and AMI combination (such as Debian ramdisk with a SUSE AMI).
Instance store-backed	<ul style="list-style-type: none">A failed drive.A corrupted file system.A mismatched ramdisk and combination (for example, a Debian ramdisk with a SUSE AMI).

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">Stop the instance.Detach the root volume.Attach the root volume to a known working instance.Run filesystem check (<code>fsck -a /dev/...</code>).Fix any errors.Detach the volume from the known working instance.Attach the volume to the stopped instance.Start the instance.Recheck the instance status.
Instance store-backed	<p>Try one of the following:</p> <ul style="list-style-type: none">Start a new instance.(Optional) Seek technical assistance for data recovery using AWS Support.

VFS: Unable to mount root fs on unknown-block (Root filesystem mismatch)

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1  
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007  
...  
Kernel command line: root=/dev/sda1 ro 4  
...  
Registering block device major 8  
...  
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(8,1)
```

Potential causes

Instance type	Potential cause
Amazon EBS-backed	<ul style="list-style-type: none">Device not attached correctly.Root device not attached at correct device point.Filesystem not in expected format.Use of legacy kernel (such as 2.6.16-XenU).A recent kernel update on your instance (faulty update, or an update bug)
Instance store-backed	Hardware device failure.

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Do one of the following:</p> <ul style="list-style-type: none">Stop and then restart the instance.Modify root volume to attach at the correct device point, possible /dev/sda1 instead of /dev/sda.Stop and modify to use modern kernel.Refer to the documentation for your Linux distribution to check for known update bugs. Change or reinstall the kernel.
Instance store-backed	Terminate the instance and launch a new instance using a modern kernel.

Error: Unable to determine major/minor number of root device... (Root file system/device mismatch)

This condition is indicated by a system log similar to the one shown below.

```
...
XENBUS: Device with no driver: device/vif/0
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

Potential causes

- Missing or incorrectly configured virtual block device driver
- Device enumeration clash (sda versus xvda or sda instead of sda1)
- Incorrect choice of instance kernel

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Stop the instance.2. Detach the volume.3. Fix the device mapping problem.4. Start the instance.5. Modify the AMI to address device mapping issues.
Instance store-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Create a new AMI with the appropriate fix (map block device correctly).2. Terminate the instance and launch a new instance from the AMI you created.

XENBUS: Device with no driver...

This condition is indicated by a system log similar to the one shown below.

```
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

Potential causes

- Missing or incorrectly configured virtual block device driver
- Device enumeration clash (sda versus xvda)
- Incorrect choice of instance kernel

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Stop the instance.2. Detach the volume.3. Fix the device mapping problem.4. Start the instance.5. Modify the AMI to address device mapping issues.
Instance store-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Create an AMI with the appropriate fix (map block device correctly).2. Terminate the instance and launch a new instance using the AMI you created.

... days without being checked, check forced (File system check required)

This condition is indicated by a system log similar to the one shown below.

```
...
Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1 has gone 361 days without being checked, check forced
```

Potential causes

Filesystem check time passed; a filesystem check is being forced.

Suggested actions

- Wait until the filesystem check completes. A filesystem check can take a long time depending on the size of the root filesystem.
- Modify your filesystems to remove the filesystem check (fsck) enforcement using tune2fs or tools appropriate for your filesystem.

fsck died with exit status... (Missing device)

This condition is indicated by a system log similar to the one shown below.

```
Cleaning up ifupdown....
Loading kernel modules...done.
...
Activating lvm and md swap...done.
Checking file systems...fsck from util-linux-ng 2.16.2
/sbin/fsck.xfs: /dev/sdh does not exist
fsck died with exit status 8
[31mfailed (code 8).[39;49m
```

Potential causes

- Ramdisk looking for missing drive
- Filesystem consistency check forced
- Drive failed or detached

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Try one or more of the following to resolve the issue:</p> <ul style="list-style-type: none">• Stop the instance, attach the volume to an existing running instance.• Manually run consistency checks.• Fix ramdisk to include relevant utilities.• Modify filesystem tuning parameters to remove consistency requirements (not recommended).
Instance store-backed	<p>Try one or more of the following to resolve the issue:</p>

For this instance type	Do this
	<ul style="list-style-type: none"> Rebundle ramdisk with correct tooling. Modify file system tuning parameters to remove consistency requirements (not recommended). Terminate the instance and launch a new instance. (Optional) Seek technical assistance for data recovery using AWS Support.

GRUB prompt (grubdom>)

This condition is indicated by a system log similar to the one shown below.

```
GNU GRUB version 0.97 (629760K lower / 0K upper memory)

[ Minimal BASH-like line editing is supported. For
the first word, TAB lists possible command
completions. Anywhere else TAB lists the possible
completions of a device/filename. ]

grubdom>
```

Potential causes

Instance type	Potential causes
Amazon EBS-backed	<ul style="list-style-type: none"> Missing GRUB configuration file. Incorrect GRUB image used, expecting GRUB configuration file at a different location. Unsupported filesystem used to store your GRUB configuration file (for example, converting your root file system to a type that is not supported by an earlier version of GRUB).
Instance store-backed	<ul style="list-style-type: none"> Missing GRUB configuration file. Incorrect GRUB image used, expecting GRUB configuration file at a different location. Unsupported filesystem used to store your GRUB configuration file (for example, converting your root file system to a type that is not supported by an earlier version of GRUB).

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Option 1: Modify the AMI and relaunch the instance:

For this instance type	Do this
	<ol style="list-style-type: none"> 1. Modify the source AMI to create a GRUB configuration file at the standard location (/boot/grub/menu.lst). 2. Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary. 3. Pick the appropriate GRUB image, (hd0-1st drive or hd00 – 1st drive, 1st partition). 4. Terminate the instance and launch a new one using the AMI that you created. <p>Option 2: Fix the existing instance:</p> <ol style="list-style-type: none"> 1. Stop the instance. 2. Detach the root filesystem. 3. Attach the root filesystem to a known working instance. 4. Mount filesystem. 5. Create a GRUB configuration file. 6. Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary. 7. Detach filesystem. 8. Attach to the original instance. 9. Modify kernel attribute to use the appropriate GRUB image (1st disk or 1st partition on 1st disk). 10. Start the instance.
Instance store-backed	<p>Option 1: Modify the AMI and relaunch the instance:</p> <ol style="list-style-type: none"> 1. Create the new AMI with a GRUB configuration file at the standard location (/boot/grub/menu.lst). 2. Pick the appropriate GRUB image, (hd0-1st drive or hd00 – 1st drive, 1st partition). 3. Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary. 4. Terminate the instance and launch a new instance using the AMI you created. <p>Option 2: Terminate the instance and launch a new instance, specifying the correct kernel.</p> <p>Note To recover data from the existing instance, contact AWS Support.</p>

Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address)

This condition is indicated by a system log similar to the one shown below.

```
...
Bringing up loopback interface: [ OK ]
Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring.
[FAILED]
Starting auditd: [ OK ]
```

Potential causes

There is a hardcoded interface MAC in the AMI configuration

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Do one of the following:</p> <ul style="list-style-type: none">• Modify the AMI to remove the hardcoding and relaunch the instance.• Modify the instance to remove the hardcoded MAC address. <p>OR</p> <p>Use the following procedure:</p> <ol style="list-style-type: none">1. Stop the instance.2. Detach the root volume.3. Attach the volume to another instance and modify the volume to remove the hardcoded MAC address.4. Attach the volume to the original instance.5. Start the instance.
Instance store-backed	<p>Do one of the following:</p> <ul style="list-style-type: none">• Modify the instance to remove the hardcoded MAC address.• Terminate the instance and launch a new instance.

Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration)

This condition is indicated by a system log similar to the one shown below.

```
audit(1313445102.626:2): enforcing=1 old_enforcing=0 auid=4294967295
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now.
Kernel panic - not syncing: Attempted to kill init!
```

Potential causes

SELinux has been enabled in error:

- Supplied kernel is not supported by GRUB
- Fallback kernel does not exist

Suggested actions

For this instance type	Do this
Amazon EBS-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Stop the failed instance.2. Detach the failed instance's root volume.3. Attach the root volume to another running Linux instance (later referred to as a recovery instance).4. Connect to the recovery instance and mount the failed instance's root volume.5. Disable SELinux on the mounted root volume. This process varies across Linux distributions; for more information, consult your OS-specific documentation. <p>Note On some systems, you disable SELinux by setting <code>SELINUX=disabled</code> in the <code>/mount_point/etc/sysconfig/selinux</code> file, where <code>mount_point</code> is the location that you mounted the volume on your recovery instance.</p> <ol style="list-style-type: none">6. Unmount and detach the root volume from the recovery instance and reattach it to the original instance.7. Start the instance.
Instance store-backed	<p>Use the following procedure:</p> <ol style="list-style-type: none">1. Terminate the instance and launch a new instance.

For this instance type	Do this
	2. (Optional) Seek technical assistance for data recovery using AWS Support .

XENBUS: Timeout connecting to devices (Xenbus timeout)

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
XENBUS: Timeout connecting to devices!
...
Kernel panic - not syncing: No init found. Try passing init= option to kernel.
```

Potential causes

- The block device is not connected to the instance
- This instance is using an old instance kernel

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following: <ul style="list-style-type: none">• Modify the AMI and instance to use a modern kernel and relaunch the instance.• Reboot the instance.
Instance store-backed	Do one of the following: <ul style="list-style-type: none">• Terminate the instance.• Modify the AMI to use a modern kernel, and launch a new instance using this AMI.

Troubleshooting an unreachable instance

You can use the following methods to troubleshoot an unreachable instance.

Contents

- [Instance reboot \(p. 1278\)](#)
- [Instance console output \(p. 1278\)](#)
- [Capture a screenshot of an unreachable instance \(p. 1278\)](#)
- [Instance recovery when a host computer fails \(p. 1279\)](#)

Instance reboot

The ability to reboot instances that are otherwise unreachable is valuable for both troubleshooting and general instance management.

Just as you can reset a computer by pressing the reset button, you can reset EC2 instances using the Amazon EC2 console, CLI, or API. For more information, see [Reboot your instance \(p. 603\)](#).

Warning

For Windows instances, this operation performs a hard reboot that might result in data corruption.

Instance console output

Console output is a valuable tool for problem diagnosis. It is especially useful for troubleshooting kernel problems and service configuration issues that could cause an instance to terminate or become unreachable before its SSH daemon can be started.

For Linux/Unix, the instance console output displays the exact console output that would normally be displayed on a physical monitor attached to a computer. The console output returns buffered information that was posted shortly after an instance transition state (start, stop, reboot, and terminate). The posted output is not continuously updated; only when it is likely to be of the most value.

For Windows instances, the instance console output includes the last three system event log errors.

You can optionally retrieve the latest serial console output at any time during the instance lifecycle. This option is only supported on [Instances built on the Nitro System \(p. 205\)](#). It is not supported through the Amazon EC2 console.

Note

Only the most recent 64 KB of posted output is stored, which is available for at least 1 hour after the last posting.

Only the instance owner can access the console output. You can retrieve the console output for your instances using the console or the command line.

To get console output using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances**, and select the instance.
3. Choose **Actions, Instance Settings, Get System Log**.

To get console output using the command line

You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [get-console-output](#) (AWS CLI)
- [Get-EC2ConsoleOutput](#) (AWS Tools for Windows PowerShell)

For more information about common system log errors, see [Troubleshooting system log errors for Linux-based instances \(p. 1258\)](#).

Capture a screenshot of an unreachable instance

If you are unable to reach your instance via SSH or RDP, you can capture a screenshot of your instance and view it as an image. The image can provide visibility as to the status of the instance, and allows

for quicker troubleshooting. You can generate screenshots while the instance is running or after it has crashed. There is no data transfer cost for this screenshot. The image is generated in JPG format and is no larger than 100 kb. This feature is not supported when the instance is using an NVIDIA GRID driver, is on bare metal instances (instances of type `*.metal`), or is powered by Arm-based Graviton or Graviton 2 processors. This feature is available in the following Regions:

- US East (N. Virginia) Region
- US East (Ohio) Region
- US West (Oregon) Region
- US West (N. California) Region
- Europe (Ireland) Region
- Europe (Frankfurt) Region
- Asia Pacific (Tokyo) Region
- Asia Pacific (Seoul) Region
- Asia Pacific (Singapore) Region
- Asia Pacific (Sydney) Region
- South America (São Paulo) Region
- Asia Pacific (Mumbai) Region
- Canada (Central) Region
- Europe (London) Region
- Europe (Paris) Region

To access the instance console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances**.
3. Select the instance to capture.
4. Choose **Actions, Instance Settings**.
5. Choose **Get Instance Screenshot**.

Right-click on the image to download and save it.

To capture a screenshot using the command line

You can use one of the following commands. The returned content is base64-encoded. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [get-console-screenshot](#) (AWS CLI)
- [GetConsoleScreenshot](#) (Amazon EC2 Query API)

Instance recovery when a host computer fails

If there is an unrecoverable issue with the hardware of an underlying host computer, AWS may schedule an instance stop event. You are notified of such an event ahead of time by email.

To recover an Amazon EBS-backed instance running on a host computer that failed

1. Back up any important data on your instance store volumes to Amazon EBS or Amazon S3.
2. Stop the instance.
3. Start the instance.

4. Restore any important data.

For more information, see [Stop and start your instance \(p. 587\)](#).

To recover an instance store-backed instance running on a host computer that failed

1. Create an AMI from the instance.
2. Upload the image to Amazon S3.
3. Back up important data to Amazon EBS or Amazon S3.
4. Terminate the instance.
5. Launch a new instance from the AMI.
6. Restore any important data to the new instance.

For more information, see [Creating an instance store-backed Linux AMI \(p. 127\)](#).

Booting from the wrong volume

In some situations, you may find that a volume other than the volume attached to /dev/xvda or /dev/sda has become the root volume of your instance. This can happen when you have attached the root volume of another instance, or a volume created from the snapshot of a root volume, to an instance with an existing root volume.

This is due to how the initial ramdisk in Linux works. It chooses the volume defined as / in the /etc/fstab, and in some distributions, this is determined by the label attached to the volume partition. Specifically, you find that your /etc/fstab looks something like the following:

```
LABEL=/ / ext4 defaults,noatime 1 1
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

If you check the label of both volumes, you see that they both contain the / label:

```
[ec2-user ~]$ sudo e2label /dev/xvda1
/
[ec2-user ~]$ sudo e2label /dev/xvdf1
/
```

In this example, you could end up having /dev/xvdf1 become the root device that your instance boots to after the initial ramdisk runs, instead of the /dev/xvda1 volume from which you had intended to boot. To solve this, use the same **e2label** command to change the label of the attached volume that you do not want to boot from.

In some cases, specifying a UUID in /etc/fstab can resolve this. However, if both volumes come from the same snapshot, or the secondary is created from a snapshot of the primary volume, they share a UUID.

```
[ec2-user ~]$ sudo blkid
/dev/xvda1: LABEL="/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"
PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
/dev/xvdf1: LABEL="old/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"
PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
```

To change the label of an attached ext4 volume

1. Use the **e2label** command to change the label of the volume to something other than `/`.

```
[ec2-user ~]$ sudo e2label /dev/xvdf1 old/
```

2. Verify that the volume has the new label.

```
[ec2-user ~]$ sudo e2label /dev/xvdf1  
old/
```

To change the label of an attached xfs volume

- Use the **xfs_admin** command to change the label of the volume to something other than `/`.

```
[ec2-user ~]$ sudo xfs_admin -L old/ /dev/xvdf1  
writing all SBs  
new label = "old/"
```

After changing the volume label as shown, you should be able to reboot the instance and have the proper volume selected by the initial ramdisk when the instance boots.

Important

If you intend to detach the volume with the new label and return it to another instance to use as the root volume, you must perform the above procedure again and change the volume label back to its original value. Otherwise, the other instance does not boot because the ramdisk is unable to find the volume with the label `/`.

Using EC2Rescue for Linux

EC2Rescue for Linux is an easy-to-use, open-source tool that can be run on an Amazon EC2 Linux instance to diagnose and troubleshoot common issues using its library of over 100 modules. A few generalized use cases for EC2Rescue for Linux include gathering syslog and package manager logs, collecting resource utilization data, and diagnosing/remediating known problematic kernel parameters and common OpenSSH issues.

If you are using a Windows instance, see [EC2Rescue for Windows Server](#).

Contents

- [Installing EC2Rescue for Linux \(p. 1281\)](#)
- [Working with EC2Rescue for Linux \(p. 1285\)](#)
- [Developing EC2Rescue modules \(p. 1287\)](#)

Installing EC2Rescue for Linux

The EC2Rescue for Linux tool can be installed on an Amazon EC2 Linux instance that meets the following prerequisites.

Prerequisites

- Supported operating systems:
 - Amazon Linux 2

- Amazon Linux 2016.09+
- SUSE Linux Enterprise Server 12+
- RHEL 7+
- Ubuntu 16.04+
- Software requirements:
 - Python 2.7.9+ or 3.2+

If your system has the required Python version, you can install the standard build. Otherwise, you can install the bundled build, which includes a minimal copy of Python.

To install the standard build

1. From a working Linux instance, download the [EC2Rescue for Linux](#) tool:

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz
```

2. (Optional) Before proceeding, you can optionally verify the signature of the EC2Rescue for Linux installation file. For more information, see [\(Optional\) Verify the signature of EC2Rescue for Linux \(p. 1282\)](#).
3. Download the sha256 hash file:

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz.sha256
```

4. Verify the integrity of the tarball:

```
sha256sum -c ec2rl.tgz.sha256
```

5. Unpack the tarball:

```
tar -xvf ec2rl.tgz
```

6. Verify the installation by listing out the help file:

```
cd ec2rl-<version_number>
./ec2rl help
```

To install the bundled build

For a link to the download and a list of limitations, see [EC2Rescue for Linux](#) on github.

(Optional) Verify the signature of EC2Rescue for Linux

The following is the recommended process of verifying the validity of the EC2Rescue for Linux package for Linux-based operating systems.

When you download an application from the internet, we recommend that you authenticate the identity of the software publisher and check that the application has not been altered or corrupted after it was published. This protects you from installing a version of the application that contains a virus or other malicious code.

If, after running the steps in this topic, you determine that the software for EC2Rescue for Linux is altered or corrupted, do not run the installation file. Instead, contact Amazon Web Services.

EC2Rescue for Linux files for Linux-based operating systems are signed using GnuPG, an open-source implementation of the Pretty Good Privacy (OpenPGP) standard for secure digital signatures. GnuPG (also known as GPG) provides authentication and integrity checking through a digital signature. AWS publishes a public key and signatures that you can use to verify the downloaded EC2Rescue for Linux package. For more information about PGP and GnuPG (GPG), see <http://www.gnupg.org>.

The first step is to establish trust with the software publisher. Download the public key of the software publisher, check that the owner of the public key is who they claim to be, and then add the public key to your keyring. Your keyring is a collection of known public keys. After you establish the authenticity of the public key, you can use it to verify the signature of the application.

Tasks

- [Install the GPG tools \(p. 1283\)](#)
- [Authenticate and import the public key \(p. 1283\)](#)
- [Verify the signature of the package \(p. 1284\)](#)

Install the GPG tools

If your operating system is Linux or Unix, the GPG tools may already be installed. To test whether the tools are installed on your system, enter `gpg2` at a command prompt. If the GPG tools are installed, you see a GPG command prompt. If the GPG tools are not installed, you see an error stating that the command cannot be found. You can install the GnuPG package from a repository.

To install GPG tools on Debian-based Linux

- From a terminal, run the following command:

```
apt-get install gnupg2
```

To install GPG tools on Red Hat-based Linux

- From a terminal, run the following command:

```
yum install gnupg2
```

Authenticate and import the public key

The next step in the process is to authenticate the EC2Rescue for Linux public key and add it as a trusted key in your GPG keyring.

To authenticate and import the EC2Rescue for Linux public key

1. At a command prompt, use the following command to obtain a copy of our public GPG build key:

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.key
```

2. At a command prompt in the directory where you saved `ec2rl.key`, use the following command to import the EC2Rescue for Linux public key into your keyring:

```
gpg2 --import ec2rl.key
```

The command returns results similar to the following:

```
gpg: /home/ec2-user/.gnupg/trustdb.gpg: trustdb created
gpg: key 2FAE2A1C: public key "ec2autodiag@amazon.com <EC2 Rescue for Linux>" imported
gpg: Total number processed: 1
gpg:          imported: 1  (RSA: 1)
```

Verify the signature of the package

After you've installed the GPG tools, authenticated and imported the EC2Rescue for Linux public key, and verified that the EC2Rescue for Linux public key is trusted, you are ready to verify the signature of the EC2Rescue for Linux installation script.

To verify the EC2Rescue for Linux installation script signature

- At a command prompt, run the following command to download the signature file for the installation script:

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz.sig
```

- Verify the signature by running the following command at a command prompt in the directory where you saved ec2rl.tgz.sig and the EC2Rescue for Linux installation file. Both files must be present.

```
gpg2 --verify ./ec2rl.tgz.sig
```

The output should look something like the following:

```
gpg: Signature made Thu 12 Jul 2018 01:57:51 AM UTC using RSA key ID 6991ED45
gpg: Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E528 BCC9 0DBF 5AFA 0F6C C36A F780 4843 2FAE 2A1C
Subkey fingerprint: 966B 0D27 85E9 AEEC 1146 7A9D 8851 1153 6991 ED45
```

If the output contains the phrase `Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"`, it means that the signature has successfully been verified, and you can proceed to run the EC2Rescue for Linux installation script.

If the output includes the phrase `BAD signature`, check whether you performed the procedure correctly. If you continue to get this response, contact Amazon Web Services and do not run the installation file that you downloaded previously.

The following are details about the warnings that you might see:

- WARNING: This key is not certified with a trusted signature! There is no indication that the signature belongs to the owner.** This refers to your personal level of trust in your belief that you possess an authentic public key for EC2Rescue for Linux. In an ideal world, you would visit an Amazon Web Services office and receive the key in person. However, more often you download it from a website. In this case, the website is an Amazon Web Services website.
- gpg2: no ultimately trusted keys found.** This means that the specific key is not "ultimately trusted" by you (or by other people whom you trust).

For more information, see <http://www.gnupg.org>.

Working with EC2Rescue for Linux

The following are common tasks you can perform to get started using this tool.

Tasks

- [Running EC2Rescue for Linux \(p. 1285\)](#)
- [Uploading the results \(p. 1285\)](#)
- [Creating backups \(p. 1286\)](#)
- [Getting help \(p. 1286\)](#)

Running EC2Rescue for Linux

You can run EC2Rescue for Linux as shown in the following examples.

Example Example: Run all modules

To run all modules, run EC2Rescue for Linux with no options:

```
./ec2rl run
```

Some modules require root access. If you are not a root user, use **sudo** to run these modules as follows:

```
sudo ./ec2rl run
```

Example Example: Run a specific module

To run only specific modules, use the **--only-modules** parameter:

```
./ec2rl run --only-modules=module_name --arguments
```

For example, this command runs the **dig** module to query the `amazon.com` domain:

```
./ec2rl run --only-modules=dig --domain=amazon.com
```

Example Example: View the results

You can view the results in `/var/tmp/ec2rl`:

```
cat /var/tmp/ec2rl/logfile_location
```

For example, view the log file for the **dig** module:

```
cat /var/tmp/ec2rl/2017-05-11T15_39_21.893145/mod_out/run/dig.log
```

Uploading the results

If AWS Support has requested the results or to share the results from an S3 bucket, upload them using the EC2Rescue for Linux CLI tool. The output of the EC2Rescue for Linux commands should provide the commands that you need to use.

Example Example: Upload results to AWS Support

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/2017-05-11T15_39_21.893145 --support-url="URLProvidedByAWSSupport"
```

Example Example: Upload results to an S3 bucket

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/2017-05-11T15_39_21.893145 --presigned-url="YourPresignedS3URL"
```

For more information about generating pre-signed URLs for Amazon S3, see [Uploading Objects Using Pre-Signed URLs](#).

Creating backups

Create a backup for your instance, one or more volumes, or a specific device ID using the following commands.

Example Example: Back up an instance using an Amazon Machine Image (AMI)

```
./ec2rl run --backup=ami
```

Example Example: Back up all volumes associated with the instance

```
./ec2rl run --backup=allvolumes
```

Example Example: Back up a specific volume

```
./ec2rl run --backup=volumeID
```

Getting help

EC2Rescue for Linux includes a help file that gives you information and syntax for each available command.

Example Example: Display the general help

```
./ec2rl help
```

Example Example: List the available modules

```
./ec2rl list
```

Example Example: Display the help for a specific module

```
./ec2rl help module_name
```

For example, use the following command to show the help file for the **dig** module:

```
./ec2rl help dig
```

Developing EC2Rescue modules

Modules are written in YAML, a data serialization standard. A module's YAML file consists of a single document, representing the module and its attributes.

Adding module attributes

The following table lists the available module attributes.

Attribute	Description
name	The name of the module. The name should be less than or equal to 18 characters in length.
version	The version number of the module.
title	A short, descriptive title for the module. This value should be less than or equal to 50 characters in length.
helptext	<p>The extended description of the module. Each line should be less than or equal to 75 characters in length. If the module consumes arguments, required or optional, include them in the helptext value.</p> <p>For example:</p> <div style="border: 1px solid black; padding: 5px;"><pre>helptext: !!str Collect output from ps for system analysis Consumes --times= for number of times to repeat Consumes --period= for time period between repetition</pre></div>
placement	<p>The stage in which the module should be run. Supported values:</p> <ul style="list-style-type: none">• prediagnostic• run• postdiagnostic
language	<p>The language that the module code is written in. Supported values:</p> <ul style="list-style-type: none">• bash• python <p>Note Python code must be compatible with both Python 2.7.9+ and Python 3.2+.</p>
remediation	Indicates whether the module supports remediation. Supported values are <code>True</code> or <code>False</code> .

Attribute	Description
	The module defaults to <code>False</code> if this is absent, making it an optional attribute for those modules that do not support remediation.
<code>content</code>	The entirety of the script code.
<code>constraint</code>	The name of the object containing the constraint values.
<code>domain</code>	A descriptor of how the module is grouped or classified. The set of included modules uses the following domains: <ul style="list-style-type: none"> • <code>application</code> • <code>net</code> • <code>os</code> • <code>performance</code>
<code>class</code>	A descriptor of the type of task performed by the module. The set of included modules uses the following classes: <ul style="list-style-type: none"> • <code>collect</code> (collects output from programs) • <code>diagnose</code> (pass/fail based on a set of criteria) • <code>gather</code> (copies files and writes to specific file)
<code>distro</code>	The list of Linux distributions that this module supports. The set of included modules uses the following distributions: <ul style="list-style-type: none"> • <code>alami</code> (Amazon Linux) • <code>rhel</code> • <code>ubuntu</code> • <code>suse</code>
<code>required</code>	The required arguments that the module is consuming from the CLI options.
<code>optional</code>	The optional arguments that the module can use.
<code>software</code>	The software executables used in the module. This attribute is intended to specify software that is not installed by default. The EC2Rescue for Linux logic ensures that these programs are present and executable before running the module.
<code>package</code>	The source software package for an executable. This attribute is intended to provide extended details on the package with the software, including a URL for downloading or getting further information.

Attribute	Description
sudo	Indicates whether root access is required to run the module. You do not need to implement sudo checks in the module script. If the value is true, then the EC2Rescue for Linux logic only runs the module when the executing user has root access.
perfimpact	Indicates whether the module can have significant performance impact upon the environment in which it is run. If the value is true and the --perfimpact=true argument is not present, then the module is skipped.
parallelexclusive	Specifies a program that requires mutual exclusivity. For example, all modules specifying "bpf" run in a serial manner.

Adding environment variables

The following table lists the available environment variables.

Environment Variable	Description
EC2RL_CALLPATH	The path to ec2rl.py. This path can be used to locate the lib directory and use vendored Python modules.
EC2RL_WORKDIR	The main tmp directory for the diagnostic tool. Default value: /var/tmp/ec2rl.
EC2RL_RUNDIR	The directory where all output is stored. Default value: /var/tmp/ec2rl/<date×tamp>.
EC2RL_GATHEREDDIR	The root directory for placing gathered module data. Default value: /var/tmp/ec2rl/<date×tamp>/mod_out/gathered/.
EC2RL_NET_DRIVER	The driver in use for the first, alphabetically ordered, non-virtual network interface on the instance. Examples: <ul style="list-style-type: none"> • xen_netfront • ixgbevf • ena
EC2RL_SUDO	True if EC2Rescue for Linux is running as root; otherwise, false.

Environment Variable	Description
EC2RL_VIRT_TYPE	<p>The virtualization type as provided by the instance metadata.</p> <p>Examples:</p> <ul style="list-style-type: none"> • default-hvm • default-paravirtual
EC2RL_INTERFACES	An enumerated list of interfaces on the system. The value is a string containing names, such as eth0, eth1, etc. This is generated via the <code>functions.bash</code> and is only available for modules that have sourced it.

Using YAML syntax

The following should be noted when constructing your module YAML files:

- The triple hyphen (---) denotes the explicit start of a document.
- The `!ec2rlcore.module` tag tells the YAML parser which constructor to call when creating the object from the data stream. You can find the constructor inside the `module.py` file.
- The `!!str` tag tells the YAML parser to not attempt to determine the type of data, and instead interpret the content as a string literal.
- The pipe character (|) tells the YAML parser that the value is a literal-style scalar. In this case, the parser includes all whitespace. This is important for modules because indentation and newline characters are kept.
- The YAML standard indent is two spaces, which can be seen in the following examples. Ensure that you maintain standard indentation (for example, four spaces for Python) for your script and then indent the entire content two spaces inside the module file.

Example modules

Example one (`mod.d/ps.yaml`):

```

--- !ec2rlcore.module.Module
# Module document. Translates directly into an almost-complete Module object
name: !!str ps
path: !!str
version: !!str 1.0
title: !!str Collect output from ps for system analysis
helptext: !!str |
    Collect output from ps for system analysis
    Requires --times= for number of times to repeat
    Requires --period= for time period between repetition
placement: !!str run
package:
  - !!str
language: !!str bash
content: !!str |
  #!/bin/bash
  error_trap()
  {
      printf "%0.s=" {1..80}
      echo -e "\nERROR: \"$BASH_COMMAND\" exited with an error on line ${BASH_LINENO[0]}"
  }

```

```
        exit 0
    }
trap error_trap ERR

# read-in shared function
source functions.bash
echo "I will collect ps output from this $EC2RL_DISTRO box for $times times every $period
seconds."
for i in $(seq 1 $times); do
    ps auxww
    sleep $period
done
constraint:
requires_ec2: !!str False
domain: !!str performance
class: !!str collect
distro: !!str alami ubuntu rhel suse
required: !!str period times
optional: !!str
software: !!str
sudo: !!str False
perfimpact: !!str False
parallelexclusive: !!str
```

Sending a diagnostic interrupt (for advanced users)

Warning

Diagnostic interrupts are intended for use by advanced users. Incorrect usage could negatively impact your instance. Sending a diagnostic interrupt to an instance could trigger an instance to crash and reboot, which could lead to the loss of data.

You can send a diagnostic interrupt to an unreachable or unresponsive Linux instance to manually trigger a *kernel panic*.

Linux operating systems typically crash and reboot when a kernel panic occurs. The specific behavior of the operating system depends on its configuration. A kernel panic can also be used to cause the instance's operating system kernel to perform tasks, such as generating a crash dump file. You can then use the information in the crash dump file to conduct root cause analysis and debug the instance.

The crash dump data is generated locally by the operating system on the instance itself.

Before sending a diagnostic interrupt to your instance, we recommend that you consult the documentation for your operating system and then make the necessary configuration changes.

Contents

- [Supported instance types \(p. 1291\)](#)
- [Prerequisites \(p. 1292\)](#)
- [Sending a diagnostic interrupt \(p. 1294\)](#)

Supported instance types

Diagnostic interrupt is supported on all Nitro-based instance types, except A1. For more information, see [Instances built on the Nitro System \(p. 205\)](#).

Prerequisites

Before using a diagnostic interrupt, you must configure your instance's operating system. This ensures that it performs the actions that you need when a kernel panic occurs.

To configure Amazon Linux 2 to generate a crash dump when a kernel panic occurs

1. Connect to your instance.
2. Install **kexec** and **kdump**.

```
[ec2-user ~]$ sudo yum install kexec-tools -y
```

3. Configure the kernel to reserve an appropriate amount of memory for the secondary kernel. The amount of memory to reserve depends on the total available memory of your instance. Open the `/etc/default/grub` file using your preferred text editor, locate the line that starts with `GRUB_CMDLINE_LINUX_DEFAULT`, and then add the `crashkernel` parameter in the following format: `crashkernel=memory_to_reserve`. For example, to reserve 160MB, modify the `grub` file as follows:

```
GRUB_CMDLINE_LINUX_DEFAULT="crashkernel=160M console=tty0 console=ttyS0,115200n8
net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
rd.shell=0"
GRUB_TIMEOUT=0
GRUB_DISABLE_RECOVERY="true"
```

4. Save the changes and close the `grub` file.
5. Rebuild the GRUB2 configuration file.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

6. On instances based on Intel and AMD processors, the `send-diagnostic-interrupt` command sends an *unknown non-maskable interrupt* (NMI) to the instance. You must configure the kernel to crash when it receives the unknown NMI. Open the `/etc/sysctl.conf` file using your preferred text editor and add the following.

```
kernel.unknown_nmi_panic=1
```

7. Reboot and reconnect to your instance.
8. Verify that the kernel has been booted with the correct `crashkernel` parameter.

```
$ grep crashkernel /proc/cmdline
```

The following example output indicates successful configuration.

```
BOOT_IMAGE=/boot/vmlinuz-4.14.128-112.105.amzn2.x86_64 root=UUID=a1e1011e-
e38f-408e-878b-fed395b47ad6 ro crashkernel=160M console=tty0 console=ttyS0,115200n8
net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
rd.shell=0
```

9. Verify that the **kdump** service is running.

```
[ec2-user ~]$ systemctl status kdump.service
```

The following example output shows the result if the **kdump** service is running.

```
kdump.service - Crash recovery kernel arming
   Loaded: loaded (/usr/lib/systemd/system/kdump.service; enabled; vendor preset:
             enabled)
     Active: active (exited) since Fri 2019-05-24 23:29:13 UTC; 22s ago
       Process: 2503 ExecStart=/usr/bin/kdumpctl start (code=exited, status=0/SUCCESS)
    Main PID: 2503 (code=exited, status=0/SUCCESS)
```

Note

By default, the crash dump file is saved to `/var/crash/`. To change the location, modify the `/etc/kdump.conf` file using your preferred text editor.

To configure Amazon Linux to generate a crash dump when a kernel panic occurs

1. Connect to your instance.
2. Install **kexec** and **kdump**.

```
[ec2-user ~]$ sudo yum install kexec-tools -y
```

3. Configure the kernel to reserve an appropriate amount of memory for the secondary kernel. The amount of memory to reserve depends on the total available memory of your instance.

```
$ sudo grub --args="crashkernel=memory_to_reserve" --update-kernel=ALL
```

For example, to reserve 160MB for the crash kernel, use the following command.

```
$ sudo grub --args="crashkernel=160M" --update-kernel=ALL
```

4. On instances based on Intel and AMD processors, the `send-diagnostic-interrupt` command sends an *unknown non-maskable interrupt* (NMI) to the instance. You must configure the kernel to crash when it receives the unknown NMI. Open the `/etc/sysctl.conf` file using your preferred text editor and add the following.

```
kernel.unknown_nmi_panic=1
```

5. Reboot and reconnect to your instance.
6. Verify that the kernel has been booted with the correct `crashkernel` parameter.

```
$ grep crashkernel /proc/cmdline
```

The following example output indicates successful configuration.

```
root=LABEL=/ console=tty1 console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295
LANG=en_US.UTF-8 KEYTABLE=us crashkernel=160M
```

7. Verify that the **kdump** service is running.

```
[ec2-user ~]$ sudo service kdump status
```

If the service is running, the command returns the `Kdump is operational` response.

Note

By default, the crash dump file is saved to `/var/crash/`. To change the location, modify the `/etc/kdump.conf` file using your preferred text editor.

To configure SUSE Linux Enterprise, Ubuntu, or Red Hat Enterprise Linux

See the following websites:

- [SUSE Linux Enterprise](#)
- [Ubuntu](#)
- [Red Hat Enterprise Linux \(RHEL\)](#)

Note

On instances based on Intel and AMD processors, the `send-diagnostic-interrupt` command sends an *unknown non-maskable interrupt* (NMI) to the instance. You must configure the kernel to crash when it receives the unknown NMI. Add the following to your configuration file.

```
kernel.unknown_nmi_panic=1
```

Sending a diagnostic interrupt

After you have completed the necessary configuration changes, you can send a diagnostic interrupt to your instance using the AWS CLI or Amazon EC2 API.

To send a diagnostic interrupt to your instance (AWS CLI)

Use the `send-diagnostic-interrupt` command and specify the instance ID.

```
aws ec2 send-diagnostic-interrupt --instance-id i-1234567890abcdef0
```

Document history

The following table describes important additions to the Amazon EC2 documentation starting in 2019. We also update the documentation frequently to address the feedback that you send us.

update-history-change	update-history-description	update-history-date
Spot Instance vCPU limits	Spot Instance limits are now managed in terms of the number of vCPUs that your running Spot Instances are either using or will use pending the fulfillment of open requests.	October 1, 2020
Capacity Reservations in Local Zones	Capacity Reservations can now be created and used in Local Zones.	September 30, 2020
Amazon Data Lifecycle Manager	Amazon Data Lifecycle Manager policies can be configured with up to four schedules.	September 17, 2020
T4g instances (p. 1295)	New general purpose instances powered by AWS Graviton2 processors, which are based on 64-bit Arm Neoverse cores and custom silicon designed by AWS for optimized performance and cost.	September 14, 2020
Hibernation support for M5a and R5a	You can now hibernate your newly-launched instances running on M5a and R5a instance types.	August 28, 2020
Provisioned IOPS SSD (io2) volumes for Amazon EBS	Provisioned IOPS SSD (io2) volumes are designed to provide 99.999 percent volume durability with an AFR no higher than 0.001 percent.	August 24, 2020
Instance metadata provides instance location and placement information	New instance metadata fields under the <code>placement</code> category: Region, placement group name, partition number, host ID, and Availability Zone ID.	August 24, 2020
C5ad instances (p. 1295)	New compute optimized instances featuring second-generation AMD EYPC processors.	August 13, 2020
Wavelength Zones	A Wavelength Zone is an isolated zone in the carrier location where the Wavelength infrastructure is deployed.	August 6, 2020

Capacity Reservation groups	You can use AWS Resource Groups to create logical collections of Capacity Reservations, and then target instance launches into those groups.	July 29, 2020
C6gd, M6gd, and R6gd instances (p. 1295)	New general purpose instances powered by AWS Graviton2 processors, which are based on 64-bit Arm Neoverse cores and custom silicon designed by AWS for optimized performance and cost.	July 27, 2020
Fast snapshot restore	You can enable fast snapshot restore for snapshots that are shared with you.	July 21, 2020
C6g and R6g instances (p. 1295)	New general purpose instances powered by AWS Graviton2 processors, which are based on 64-bit Arm Neoverse cores and custom silicon designed by AWS for optimized performance and cost.	June 10, 2020
Bare metal instances for G4 (p. 1295)	New instances that provide your applications with direct access to the physical resources of the host server.	June 5, 2020
C5a instances (p. 1295)	New compute optimized instances featuring second-generation AMD EYPC processors.	June 4, 2020
Bring your own IPv6 addresses	You can bring part or all of your IPv6 address range from your on-premises network to your AWS account.	May 21, 2020
M6g instances (p. 1295)	New general purpose instances powered by AWS Graviton2 processors, which are based on 64-bit Arm Neoverse cores and custom silicon designed by AWS for optimized performance and cost.	May 11, 2020
Launch instances using a Systems Manager parameter	You can specify a AWS Systems Manager parameter instead of an AMI when you launch an instance.	May 5, 2020
Customize scheduled event notifications	You can customize scheduled event notifications to include tags in the email notification.	May 4, 2020

Amazon Linux 2 Kernel Live Patching	Kernel Live Patching for Amazon Linux 2 enables you to apply security vulnerability and critical bug patches to a running Linux kernel, without reboots or disruptions to running applications.	April 28, 2020
Amazon EBS Multi-Attach	You can now attach a single Provisioned IOPS SSD (io1) volume to up to 16 Nitro-based instances that are in the same Availability Zone.	February 14, 2020
Stop and start a Spot Instance	You can now stop your Spot Instances backed by Amazon EBS and start them at will, instead of relying on the stop interruption behavior.	January 13, 2020
Resource tagging (p. 1295)	You can tag egress-only internet gateways, local gateways, local gateway route tables, local gateway virtual interfaces, local gateway virtual interface groups, local gateway route table VPC associations, and local gateway route table virtual interface group associations.	January 10, 2020
Connect to your instance using Session Manager	You can start a Session Manager session with an instance from the Amazon EC2 console.	December 18, 2019
Inf1 instances (p. 1295)	New instances featuring AWS Inferentia, a machine learning inference chip designed to deliver high performance at a low cost.	December 3, 2019
Dedicated Hosts and host resource groups	Dedicated Hosts can now be used with host resource groups.	December 2, 2019
Dedicated Host sharing	You can now share your Dedicated Hosts across AWS accounts.	December 2, 2019
Default credit specification at the account level	You can set the default credit specification per burstable performance instance family at the account level per AWS Region.	November 25, 2019
Instance type discovery	You can find an instance type that meets your needs.	November 22, 2019

Dedicated Hosts (p. 1295)	You can now configure a Dedicated Host to support multiple instance types in an instance family.	November 21, 2019
Amazon EBS fast snapshot restores	You can enable fast snapshot restores on an EBS snapshot to ensure that EBS volumes created from the snapshot are fully-initialized at creation and instantly deliver all of their provisioned performance.	November 20, 2019
Instance Metadata Service Version 2	You can use Instance Metadata Service Version 2, which is a session-oriented method for requesting instance metadata.	November 19, 2019
Elastic Fabric Adapter (p. 1295)	Elastic Fabric Adapters can now be used with Intel MPI 2019 Update 6.	November 15, 2019
Queued purchases of Reserved Instances	You can queue the purchase of a Reserved Instance up to three years in advance.	October 4, 2019
G4 instances (p. 1295)	New instances featuring NVIDIA Tesla GPUs.	September 19, 2019
Diagnostic interrupt	You can send a diagnostic interrupt to an unreachable or unresponsive instance to trigger a kernel panic.	August 14, 2019
Capacity optimized allocation strategy	Using EC2 Fleet or Spot Fleet, you can now launch Spot Instances from Spot pools with optimal capacity for the number of instances that are launching.	August 12, 2019
On-Demand Capacity Reservation sharing	You can now share your Capacity Reservations across AWS accounts.	July 29, 2019
Elastic Fabric Adapter (p. 1295)	EFA now supports Open MPI 3.1.4 and Intel MPI 2019 Update 4.	July 26, 2019
Resource tagging (p. 1295)	You can tag launch templates on creation.	July 24, 2019
EC2 Instance Connect	EC2 Instance Connect is a simple and secure way to connect to your instances using Secure Shell (SSH).	June 27, 2019

Host recovery	Automatically restart your instances on a new host in the event of an unexpected hardware failure on a Dedicated Host.	June 5, 2019
Amazon EBS multi-volume snapshots	You can take exact point-in-time, data coordinated, and crash-consistent snapshots across multiple EBS volumes attached to an EC2 instance.	May 29, 2019
Resource tagging (p. 1295)	You can tag Dedicated Host Reservations.	May 27, 2019
Amazon EBS encryption by default	After you enable encryption by default in a Region, all new EBS volumes you create in the Region are encrypted using the default CMK for EBS encryption.	May 23, 2019
Resource tagging (p. 1295)	You can tag VPC endpoints, endpoint services, and endpoint service configurations.	May 13, 2019
Windows to Linux Replatforming Assistant for Microsoft SQL Server Databases	Move existing Microsoft SQL Server workloads from a Windows to a Linux operating system.	May 8, 2019
I3en instances (p. 1295)	New I3en instances can utilize up to 100 Gbps of network bandwidth.	May 8, 2019
Elastic Fabric Adapter	You can attach an Elastic Fabric Adapter to your instances to accelerate High Performance Computing (HPC) applications.	April 29, 2019
T3a instances (p. 1295)	New instances featuring AMD EYPC processors.	April 24, 2019
M5ad and R5ad instances (p. 1295)	New instances featuring AMD EYPC processors.	March 27, 2019
Resource tagging (p. 1295)	You can assign custom tags to your Dedicated Host Reservations to categorize them in different ways.	March 14, 2019
Bare metal instances for M5, M5d, R5, R5d, and z1d (p. 1295)	New instances that provide your applications with direct access to the physical resources of the host server.	February 13, 2019

History for previous years

The following table describes important additions to the Amazon EC2 documentation in 2018 and earlier years.

Feature	API version	Description	Release date
Partition placement groups	2016-11-15	Partition placement groups spread instances across logical partitions, ensuring that instances in one partition do not share underlying hardware with instances in other partitions. For more information, see Partition placement groups (p. 871) .	20 December 2018
p3dn.24xlarge instances	2016-11-15	New p3dn.24xlarge instances provide 100 Gbps of network bandwidth.	7 December 2018
Hibernate EC2 Linux instances	2016-11-15	You can hibernate a Linux instance if it's enabled for hibernation and it meets the hibernation prerequisites. For more information, see Hibernate your Linux instance (p. 591) .	28 November 2018
Amazon Elastic Inference Accelerators	2016-11-15	You can attach an Amazon EI accelerator to your instances to add GPU-powered acceleration to reduce the cost of running deep learning inference. For more information, see Amazon Elastic Inference (p. 691) .	28 November 2018
Instances featuring 100 Gbps of network bandwidth	2016-11-15	New C5n instances can utilize up to 100 Gbps of network bandwidth.	26 November 2018
Instances featuring Arm-based Processors	2016-11-15	New A1 instances deliver significant cost savings and are ideally suited for scale-out and Arm-based workloads.	26 November 2018
Spot console recommends a fleet of instances	2016-11-15	The Spot console recommends a fleet of instances based on Spot best practice (instance diversification) to meet the minimum hardware specifications (vCPUs, memory, and storage) for your application need. For more information, see Creating a Spot Fleet request (p. 391) .	20 November 2018
New EC2 Fleet request type: instant	2016-11-15	EC2 Fleet now supports a new request type, instant, that you can use to synchronously provision capacity across instance types and purchase models. The instant request returns the launched instances in the API response, and takes no further action, enabling you to control if and when instances are launched. For more information, see EC2 Fleet request types (p. 527) .	14 November 2018
Instances featuring AMD EYPC processors	2016-11-15	New general purpose (M5a) and memory optimized instances (R5a) offer lower-priced options for microservices, small to medium	6 November 2018

Feature	API version	Description	Release date
		databases, virtual desktops, development and test environments, business applications, and more.	
Spot savings information	2016-11-15	You can view the savings made from using Spot Instances for a single Spot Fleet or for all Spot Instances. For more information, see Savings from purchasing Spot Instances (p. 365) .	5 November 2018
Console support for optimizing CPU options	2016-11-15	When you launch an instance, you can optimize the CPU options to suit specific workloads or business needs using the Amazon EC2 console. For more information, see Optimizing CPU options (p. 632) .	31 October 2018
Console support for creating a launch template from an instance	2016-11-15	You can create a launch template using an instance as the basis for a new launch template using the Amazon EC2 console. For more information, see Creating a launch template (p. 506) .	30 October 2018
On-Demand Capacity Reservations	2016-11-15	You can reserve capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. This allows you to create and manage capacity reservations independently from the billing discounts offered by Reserved Instances (RI). For more information, see On-Demand Capacity Reservations (p. 474) .	25 October 2018
Bring Your Own IP Addresses (BYOIP)	2016-11-15	You can bring part or all of your public IPv4 address range from your on-premises network to your AWS account. After you bring the address range to AWS, it appears in your account as an address pool. You can create an Elastic IP address from your address pool and use it with your AWS resources. For more information, see Bring your own IP addresses (BYOIP) in Amazon EC2 (p. 775) .	23 October 2018
g3s.xlarge instances	2016-11-15	Expands the range of the accelerated-computing G3 instance family with the introduction of g3s.xlarge instances.	11 October 2018
Dedicated Host tag on create and console support	2016-11-15	You can tag your Dedicated Hosts on creation, and you can manage your Dedicated Host tags using the Amazon EC2 console. For more information, see Allocating Dedicated Hosts (p. 443) .	08 October 2018
High memory instances	2016-11-15	These instances are purpose-built to run large in-memory databases. They offer bare metal performance with direct access to host hardware. For more information, see Memory optimized instances (p. 260) .	27 September 2018
f1.4xlarge instances	2016-11-15	Expands the range of the accelerated-computing F1 instance family with the introduction of f1.4xlarge instances.	25 September 2018

Feature	API version	Description	Release date
Console support for scheduled scaling for Spot Fleet	2016-11-15	Increase or decrease the current capacity of the fleet based on the date and time. For more information, see Scale Spot Fleet using scheduled scaling (p. 418) .	20 September 2018
T3 instances	2016-11-15	T3 instances are the next generation burstable general-purpose instance type that provide a baseline level of CPU performance with the ability to burst CPU usage at any time for as long as required. For more information, see Burstable performance instances (p. 219) .	21 August 2018
Allocation strategies for EC2 Fleets	2016-11-15	You can specify whether On-Demand capacity is fulfilled by price (lowest price first) or priority (highest priority first). You can specify the number of Spot pools across which to allocate your target Spot capacity. For more information, see Allocation strategies for Spot Instances (p. 527) .	26 July 2018
Allocation strategies for Spot Fleets	2016-11-15	You can specify whether On-Demand capacity is fulfilled by price (lowest price first) or priority (highest priority first). You can specify the number of Spot pools across which to allocate your target Spot capacity. For more information, see Allocation strategy for Spot Instances (p. 359) .	26 July 2018
R5 and R5d instances	2016-11-15	R5 and R5d instances are ideally suited for high-performance databases, distributed in-memory caches, and in-memory analytics. R5d instances come with NVMe instance store volumes. For more information, see Memory optimized instances (p. 260) .	25 July 2018
z1d instances	2016-11-15	These instances are designed for applications that require high per-core performance with a large amount of memory, such as electronic design automation (EDA) and relational databases. These instances come with NVME instance store volumes. For more information, see Memory optimized instances (p. 260) .	25 July 2018
Automate snapshot lifecycle	2016-11-15	You can use Amazon Data Lifecycle Manager to automate creation and deletion of snapshots for your EBS volumes. For more information, see Automating the Amazon EBS snapshot lifecycle (p. 1097) .	12 July 2018
Launch template CPU options	2016-11-15	When you create a launch template using the command line tools, you can optimize the CPU options to suit specific workloads or business needs. For more information, see Creating a launch template (p. 506) .	11 July 2018

Feature	API version	Description	Release date
Tag Dedicated Hosts	2016-11-15	You can tag your Dedicated Hosts. For more information, see Tagging Dedicated Hosts (p. 453) .	3 July 2018
i3.metal instances	2016-11-15	i3.metal instances provide your applications with direct access to the physical resources of the host server, such as processors and memory. For more information, see Storage optimized instances (p. 272) .	17 May 2018
Get latest console output	2016-11-15	You can retrieve the latest console output for some instance types when you use the get-console-output AWS CLI command.	9 May 2018
Optimize CPU options	2016-11-15	When you launch an instance, you can optimize the CPU options to suit specific workloads or business needs. For more information, see Optimizing CPU options (p. 632) .	8 May 2018
EC2 Fleet	2016-11-15	You can use EC2 Fleet to launch a group of instances across different EC2 instance types and Availability Zones, and across On-Demand Instance, Reserved Instance, and Spot Instance purchasing models. For more information, see Launching instances using an EC2 Fleet (p. 524) .	2 May 2018
On-Demand Instances in Spot Fleets	2016-11-15	You can include a request for On-Demand capacity in your Spot Fleet request to ensure that you always have instance capacity. For more information, see How Spot Fleet works (p. 358) .	2 May 2018
Tag EBS snapshots on creation	2016-11-15	You can apply tags to snapshots during creation. For more information, see Creating Amazon EBS snapshots (p. 1063) .	2 April 2018
Change placement groups	2016-11-15	You can move an instance in or out of a placement group, or change its placement group. For more information, see Changing the placement group for an instance (p. 879) .	1 March 2018
Longer resource IDs	2016-11-15	You can enable the longer ID format for more resource types. For more information, see Resource IDs (p. 1223) .	9 February 2018
Network performance improvements	2016-11-15	Instances outside of a cluster placement group can now benefit from increased bandwidth when sending or receiving network traffic between other instances or Amazon S3. For more information, see Networking and storage features (p. 205) .	24 January 2018
Tag Elastic IP addresses	2016-11-15	You can tag your Elastic IP addresses. For more information, see Tagging an Elastic IP address (p. 784) .	21 December 2017

Feature	API version	Description	Release date
Amazon Linux 2	2016-11-15	Amazon Linux 2 is a new version of Amazon Linux. It provides a high performance, stable, and secure foundation for your applications. For more information, see Amazon Linux (p. 175) .	13 December 2017
Amazon Time Sync Service	2016-11-15	You can use the Amazon Time Sync Service to keep accurate time on your instance. For more information, see Setting the time for your Linux instance (p. 628) .	29 November 2017
T2 Unlimited	2016-11-15	T2 Unlimited instances can burst above the baseline for as long as required. For more information, see Burstable performance instances (p. 219) .	29 November 2017
Launch templates	2016-11-15	A launch template can contain all or some of the parameters to launch an instance, so that you don't have to specify them every time you launch an instance. For more information, see Launching an instance from a launch template (p. 504) .	29 November 2017
Spread placement	2016-11-15	Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other. For more information, see Spread placement groups (p. 871) .	29 November 2017
H1 instances	2016-11-15	H1 instances are designed for high-performance big data workloads. For more information, see Storage optimized instances (p. 272) .	28 November 2017
M5 instances	2016-11-15	M5 instances are the next generation of general purpose compute instances. They provide a balance of compute, memory, storage, and network resources.	28 November 2017
Spot Instance hibernation	2016-11-15	The Spot service can hibernate Spot Instances in the event of an interruption. For more information, see Hibernating interrupted Spot Instances (p. 428) .	28 November 2017
Spot Fleet target tracking	2016-11-15	You can set up target tracking scaling policies for your Spot Fleet. For more information, see Scale Spot Fleet using a target tracking policy (p. 416) .	17 November 2017
Spot Fleet integrates with Elastic Load Balancing	2016-11-15	You can attach one or more load balancers to a Spot Fleet.	10 November 2017
X1e instances	2016-11-15	X1e instances are ideally suited for high-performance databases, in-memory databases, and other memory-intensive enterprise applications. For more information, see Memory optimized instances (p. 260) .	28 November 2017

Feature	API version	Description	Release date
C5 instances	2016-11-15	C5 instances are designed for compute-heavy applications. For more information, see Compute optimized instances (p. 253) .	6 November 2017
Merge and split Convertible Reserved Instances	2016-11-15	You can exchange (merge) two or more Convertible Reserved Instances for a new Convertible Reserved Instance. You can also use the modification process to split a Convertible Reserved Instance into smaller reservations. For more information, see Exchanging Convertible Reserved Instances (p. 342) .	6 November 2017
P3 instances	2016-11-15	P3 instances are the next generation of compute-optimized GPU instances. For more information, see Linux accelerated computing instances (p. 278) .	25 October 2017
Modify VPC tenancy	2016-11-15	You can change the instance tenancy attribute of a VPC from dedicated to default. For more information, see Changing the Tenancy of a VPC (p. 474) .	16 October 2017
Per second billing	2016-11-15	Amazon EC2 charges for Linux-based usage by the second, with a one-minute minimum charge.	2 October 2017
Stop on interruption	2016-11-15	You can specify whether Amazon EC2 should stop or terminate Spot Instances when they are interrupted. For more information, see Interruption behaviors (p. 427) .	18 September 2017
Tag NAT gateways	2016-11-15	You can tag your NAT gateway. For more information, see Tagging your resources (p. 1231) .	7 September 2017
Security group rule descriptions	2016-11-15	You can add descriptions to your security group rules. For more information, see Security group rules (p. 1000) .	31 August 2017
Recover Elastic IP addresses	2016-11-15	If you release an Elastic IP address for use in a VPC, you might be able to recover it. For more information, see Recovering an Elastic IP address (p. 787) .	11 August 2017
Tag Spot Fleet instances	2016-11-15	You can configure your Spot Fleet to automatically tag the instances that it launches.	24 July 2017
G3 instances	2016-11-15	G3 instances provide a cost-effective, high-performance platform for graphics applications using DirectX or OpenGL. G3 instances also provide NVIDIA GRID Virtual Workstation features, supporting 4 monitors with resolutions up to 4096x2160. For more information, see Linux accelerated computing instances (p. 278) .	13 July 2017

Feature	API version	Description	Release date
F1 instances	2016-11-15	F1 instances represent the next generation of accelerated computing instances. For more information, see Linux accelerated computing instances (p. 278) .	19 April 2017
Tag resources during creation	2016-11-15	You can apply tags to instances and volumes during creation. For more information, see Tagging your resources (p. 1231) . In addition, you can use tag-based resource-level permissions to control the tags that are applied. For more information see, Granting permission to tag resources during creation (p. 926) .	28 March 2017
I3 instances	2016-11-15	I3 instances represent the next generation of storage optimized instances. For more information, see Storage optimized instances (p. 272) .	23 February 2017
Perform modifications on attached EBS volumes	2016-11-15	With most EBS volumes attached to most EC2 instances, you can modify volume size, type, and IOPS without detaching the volume or stopping the instance. For more information, see Amazon EBS Elastic Volumes (p. 1109) .	13 February 2017
Attach an IAM role	2016-11-15	You can attach, detach, or replace an IAM role for an existing instance. For more information, see IAM roles for Amazon EC2 (p. 974) .	9 February 2017
Dedicated Spot Instances	2016-11-15	You can run Spot Instances on single-tenant hardware in a virtual private cloud (VPC). For more information, see Specifying a tenancy for your Spot Instances (p. 369) .	19 January 2017
IPv6 support	2016-11-15	You can associate an IPv6 CIDR with your VPC and subnets, and assign IPv6 addresses to instances in your VPC. For more information, see Amazon EC2 instance IP addressing (p. 759) .	1 December 2016
R4 instances	2016-09-15	R4 instances represent the next generation of memory optimized instances. R4 instances are well-suited for memory-intensive, latency-sensitive workloads such as business intelligence (BI), data mining and analysis, in-memory databases, distributed web scale in-memory caching, and applications performance real-time processing of unstructured big data. For more information, see Memory optimized instances (p. 260)	30 November 2016

Feature	API version	Description	Release date
New t2.xlarge and t2.2xlarge instance types	2016-09-15	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see Burstable performance instances (p. 219) .	30 November 2016
P2 instances	2016-09-15	P2 instances use NVIDIA Tesla K80 GPUs and are designed for general purpose GPU computing using the CUDA or OpenCL programming models. For more information, see Linux accelerated computing instances (p. 278) .	29 September 2016
m4.16xlarge instances	2016-04-01	Expands the range of the general-purpose M4 family with the introduction of m4.16xlarge instances, with 64 vCPUs and 256 GiB of RAM.	6 September 2016
Automatic scaling for Spot Fleet		You can now set up scaling policies for your Spot Fleet. For more information, see Automatic scaling for Spot Fleet (p. 414) .	1 September 2016
Elastic Network Adapter (ENA)	2016-04-01	You can now use ENA for enhanced networking. For more information, see Enhanced networking support (p. 813) .	28 June 2016
Enhanced support for viewing and modifying longer IDs	2016-04-01	You can now view and modify longer ID settings for other IAM users, IAM roles, or the root user. For more information, see Resource IDs (p. 1223) .	23 June 2016
Copy encrypted Amazon EBS snapshots between AWS accounts	2016-04-01	You can now copy encrypted EBS snapshots between AWS accounts. For more information, see Copying an Amazon EBS snapshot (p. 1068) .	21 June 2016
Capture a screenshot of an instance console	2015-10-01	You can now obtain additional information when debugging instances that are unreachable. For more information, see Capture a screenshot of an unreachable instance (p. 1278) .	24 May 2016
X1 instances	2015-10-01	Memory-optimized instances designed for running in-memory databases, big data processing engines, and high performance computing (HPC) applications. For more information, see Memory optimized instances (p. 260) .	18 May 2016
Two new EBS volume types	2015-10-01	You can now create Throughput Optimized HDD (st1) and Cold HDD (sc1) volumes. For more information, see Amazon EBS volume types (p. 1023) .	19 April 2016
Added new NetworkPacketsIn and NetworkPacketsOut metrics for Amazon EC2		Added new NetworkPacketsIn and NetworkPacketsOut metrics for Amazon EC2. For more information, see Instance metrics (p. 718) .	23 March 2016

Feature	API version	Description	Release date
CloudWatch metrics for Spot Fleet		You can now get CloudWatch metrics for your Spot Fleet. For more information, see CloudWatch metrics for Spot Fleet (p. 412) .	21 March 2016
Scheduled Instances	2015-10-01	Scheduled Reserved Instances (Scheduled Instances) enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified start time and duration. For more information, see Scheduled Reserved Instances (p. 347) .	13 January 2016
Longer resource IDs	2015-10-01	We're gradually introducing longer length IDs for some Amazon EC2 and Amazon EBS resource types. During the opt-in period, you can enable the longer ID format for supported resource types. For more information, see Resource IDs (p. 1223) .	13 January 2016
ClassicLink DNS support	2015-10-01	You can enable ClassicLink DNS support for your VPC so that DNS hostnames that are addressed between linked EC2-Classic instances and instances in the VPC resolve to private IP addresses and not public IP addresses. For more information, see Enabling ClassicLink DNS support (p. 898) .	11 January 2016
New t2.nano instance type	2015-10-01	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see Burstable performance instances (p. 219) .	15 December 2015
Dedicated hosts	2015-10-01	An Amazon EC2 Dedicated host is a physical server with instance capacity dedicated for your use. For more information, see Dedicated Hosts (p. 438) .	23 November 2015
Spot Instance duration	2015-10-01	You can now specify a duration for your Spot Instances. For more information, see Defining a duration for your Spot Instances (p. 369) .	6 October 2015
Spot Fleet modify request	2015-10-01	You can now modify the target capacity of your Spot Fleet request. For more information, see Modifying a Spot Fleet request (p. 401) .	29 September 2015
Spot Fleet diversified allocation strategy	2015-04-15	You can now allocate Spot Instances in multiple Spot pools using a single Spot Fleet request. For more information, see Allocation strategy for Spot Instances (p. 359) .	15 September 2015

Feature	API version	Description	Release date
Spot Fleet instance weighting	2015-04-15	You can now define the capacity units that each instance type contributes to your application's performance, and adjust the amount you are willing to pay for Spot Instances for each Spot pool accordingly. For more information, see Spot Fleet instance weighting (p. 361) .	31 August 2015
New reboot alarm action and new IAM role for use with alarm actions		Added the reboot alarm action and new IAM role for use with alarm actions. For more information, see Create alarms that stop, terminate, reboot, or recover an instance (p. 738) .	23 July 2015
New t2.large instance type		T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see Burstable performance instances (p. 219) .	16 June 2015
M4 instances		The next generation of general-purpose instances that provide a balance of compute, memory, and network resources. M4 instances are powered by a custom Intel 2.4 GHz Intel® Xeon® E5 2676v3 (Haswell) processor with AVX2.	11 June 2015
Spot Fleets	2015-04-15	You can manage a collection, or fleet, of Spot Instances instead of managing separate Spot Instance requests. For more information, see How Spot Fleet works (p. 358) .	18 May 2015
Migrate Elastic IP addresses to EC2-Classic	2015-04-15	You can migrate an Elastic IP address that you've allocated for use in EC2-Classic to be used in a VPC. For more information, see Migrating an Elastic IP Address from EC2-Classic (p. 889) .	15 May 2015
Importing VMs with multiple disks as AMIs	2015-03-01	The VM Import process now supports importing VMs with multiple disks as AMIs. For more information, see Importing a VM as an Image Using VM Import/Export in the <i>VM Import/Export User Guide</i> .	23 April 2015
New g2.8xlarge instance type		The new g2.8xlarge instance is backed by four high-performance NVIDIA GPUs, making it well suited for GPU compute workloads including large scale rendering, transcoding, machine learning, and other server-side workloads that require massive parallel processing power.	7 April 2015

Feature	API version	Description	Release date
D2 instances		<p>Next generation Amazon EC2 dense-storage instances that are optimized for applications requiring sequential access to large amount of data on direct attached instance storage. D2 instances are designed to offer best price/performance in the dense-storage family.</p> <p>Powered by 2.4 GHz Intel® Xeon® E5 2676v3 (Haswell) processors, D2 instances improve on HS1 instances by providing additional compute power, more memory, and Enhanced Networking. In addition, D2 instances are available in four instance sizes with 6TB, 12TB, 24TB, and 48TB storage options.</p> <p>For more information, see Storage optimized instances (p. 272).</p>	24 March 2015
Automatic recovery for EC2 instances		<p>You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. A recovered instance is identical to the original instance, including the instance ID, IP addresses, and all instance metadata.</p> <p>For more information, see Recover your instance (p. 612).</p>	12 January 2015
C4 instances		<p>Next-generation compute-optimized instances that provide very high CPU performance at an economical price. C4 instances are based on custom 2.9 GHz Intel® Xeon® E5-2666 v3 (Haswell) processors. With additional Turbo boost, the processor clock speed in C4 instances can reach as high as 3.5Ghz with 1 or 2 core turbo. Expanding on the capabilities of C3 compute-optimized instances, C4 instances offer customers the highest processor performance among EC2 instances. These instances are ideally suited for high-traffic web applications, ad serving, batch processing, video encoding, distributed analytics, high-energy physics, genome analysis, and computational fluid dynamics.</p> <p>For more information, see Compute optimized instances (p. 253).</p>	11 January 2015
ClassicLink	2014-10-01	<p>ClassicLink enables you to link your EC2-Classic instance to a VPC in your account. You can associate VPC security groups with the EC2-Classic instance, enabling communication between your EC2-Classic instance and instances in your VPC using private IP addresses. For more information, see ClassicLink (p. 892).</p>	7 January 2015

Feature	API version	Description	Release date
Spot Instance termination notices		<p>The best way to protect against Spot Instance interruption is to architect your application to be fault tolerant. In addition, you can take advantage of Spot Instance termination notices, which provide a two-minute warning before Amazon EC2 must terminate your Spot Instance.</p> <p>For more information, see Spot Instance interruption notices (p. 431).</p>	5 January 2015
DescribeVolumes pagination support	2014-09-01	<p>The <code>DescribeVolumes</code> API call now supports the pagination of results with the <code>MaxResults</code> and <code>NextToken</code> parameters. For more information, see DescribeVolumes in the <i>Amazon EC2 API Reference</i>.</p>	23 October 2014
T2 instances	2014-06-15	<p>T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see Burstable performance instances (p. 219).</p>	30 June 2014
New EC2 Service Limits page		<p>Use the EC2 Service Limits page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-region basis.</p>	19 June 2014
Amazon EBS General Purpose SSD Volumes	2014-05-01	<p>General Purpose SSD volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies, the ability to burst to 3,000 IOPS for extended periods of time, and a base performance of 3 IOPS/GiB. General Purpose SSD volumes can range in size from 1 GiB to 1 TiB. For more information, see General Purpose SSD (gp2) volumes (p. 1026).</p>	16 June 2014
Amazon EBS encryption	2014-05-01	<p>Amazon EBS encryption offers seamless encryption of EBS data volumes and snapshots, eliminating the need to build and maintain a secure key management infrastructure. EBS encryption enables data at rest security by encrypting your data using Amazon-managed keys. The encryption occurs on the servers that host EC2 instances, providing encryption of data as it moves between EC2 instances and EBS storage. For more information, see Amazon EBS encryption (p. 1121).</p>	21 May 2014

Feature	API version	Description	Release date
R3 instances	2014-02-01	<p>Next generation memory-optimized instances with the best price point per GiB of RAM and high performance. These instances are ideally suited for relational and NoSQL databases, in-memory analytics solutions, scientific computing, and other memory-intensive applications that can benefit from the high memory per vCPU, high compute performance, and enhanced networking capabilities of R3 instances.</p> <p>For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instance Types.</p>	9 April 2014
New Amazon Linux AMI release		Amazon Linux AMI 2014.03 is released.	27 March 2014
Amazon EC2 Usage Reports		Amazon EC2 Usage Reports is a set of reports that shows cost and usage data of your usage of EC2. For more information, see Amazon EC2 usage reports (p. 1243) .	28 January 2014
Additional M3 instances	2013-10-15	The M3 instance sizes <code>m3.medium</code> and <code>m3.large</code> are now supported. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instance Types .	20 January 2014
I2 instances	2013-10-15	These instances provide very high IOPS and support TRIM on Linux instances for better successive SSD write performance. I2 instances also support enhanced networking that delivers improved inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. For more information, see Storage optimized instances (p. 272) .	19 December 2013
Updated M3 instances	2013-10-15	The M3 instance sizes, <code>m3.xlarge</code> and <code>m3.2xlarge</code> now support instance store with SSD volumes.	19 December 2013
Importing Linux virtual machines	2013-10-15	The VM Import process now supports the importation of Linux instances. For more information, see the VM Import/Export User Guide .	16 December 2013
Resource-level permissions for RunInstances	2013-10-15	You can now create policies in AWS Identity and Access Management to control resource-level permissions for the Amazon EC2 RunInstances API action. For more information and example policies, see Identity and access management for Amazon EC2 (p. 919) .	20 November 2013

Feature	API version	Description	Release date
C3 instances	2013-10-15	<p>Compute-optimized instances that provide very high CPU performance at an economical price. C3 instances also support enhanced networking that delivers improved inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. These instances are ideally suited for high-traffic web applications, ad serving, batch processing, video encoding, distributed analytics, high-energy physics, genome analysis, and computational fluid dynamics.</p> <p>For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instance Types.</p>	14 November 2013
Launching an instance from the AWS Marketplace		You can now launch an instance from the AWS Marketplace using the Amazon EC2 launch wizard. For more information, see Launching an AWS Marketplace instance (p. 522) .	11 November 2013
G2 instances	2013-10-01	These instances are ideally suited for video creation services, 3D visualizations, streaming graphics-intensive applications, and other server-side workloads requiring massive parallel processing power. For more information, see Linux accelerated computing instances (p. 278) .	4 November 2013
New launch wizard		There is a new and redesigned EC2 launch wizard. For more information, see Launching an instance using the Launch Instance Wizard (p. 499) .	10 October 2013
Modifying Instance Types of Amazon EC2 Reserved Instances	2013-10-01	You can now modify the instance type of Linux Reserved Instances within the same family (for example, M1, M2, M3, C1). For more information, see Modifying Reserved Instances (p. 335) .	09 October 2013
New Amazon Linux AMI release		Amazon Linux AMI 2013.09 is released.	30 September 2013
Modifying Amazon EC2 Reserved Instances	2013-08-15	You can now modify Reserved Instances in a Region. For more information, see Modifying Reserved Instances (p. 335) .	11 September 2013
Assigning a public IP address	2013-07-15	You can now assign a public IP address when you launch an instance in a VPC. For more information, see Assigning a public IPv4 address during instance launch (p. 764) .	20 August 2013
Granting resource-level permissions	2013-06-15	Amazon EC2 supports new Amazon Resource Names (ARNs) and condition keys. For more information, see IAM policies for Amazon EC2 (p. 921) .	8 July 2013

Feature	API version	Description	Release date
Incremental Snapshot Copies	2013-02-01	You can now perform incremental snapshot copies. For more information, see Copying an Amazon EBS snapshot (p. 1068) .	11 June 2013
New Tags page		There is a new Tags page in the Amazon EC2 console. For more information, see Tagging your Amazon EC2 resources (p. 1229) .	04 April 2013
New Amazon Linux AMI release		Amazon Linux AMI 2013.03 is released.	27 March 2013
Additional EBS-optimized instance types	2013-02-01	The following instance types can now be launched as EBS-optimized instances: c1.xlarge , m2.2xlarge , m3.xlarge , and m3.2xlarge . For more information, see Amazon EBS-optimized instances (p. 1139) .	19 March 2013
Copy an AMI from one Region to another	2013-02-01	You can copy an AMI from one Region to another, enabling you to launch consistent instances in more than one AWS Region quickly and easily. For more information, see Copying an AMI (p. 163) .	11 March 2013
Launch instances into a default VPC	2013-02-01	Your AWS account is capable of launching instances into either EC2-Classic or a VPC, or only into a VPC, on a region-by-region basis. If you can launch instances only into a VPC, we create a default VPC for you. When you launch an instance, we launch it into your default VPC, unless you create a nondefault VPC and specify it when you launch the instance.	11 March 2013
High-memory cluster (cr1.8xlarge) instance type	2012-12-01	Have large amounts of memory coupled with high CPU and network performance. These instances are well suited for in-memory analytics, graph analysis, and scientific computing applications.	21 January 2013
High storage (hs1.8xlarge) instance type	2012-12-01	High storage instances provide very high storage density and high sequential read and write performance per instance. They are well-suited for data warehousing, Hadoop/MapReduce, and parallel file systems.	20 December 2012
EBS snapshot copy	2012-12-01	You can use snapshot copies to create backups of data, to create new Amazon EBS volumes, or to create Amazon Machine Images (AMIs). For more information, see Copying an Amazon EBS snapshot (p. 1068) .	17 December 2012

Feature	API version	Description	Release date
Updated EBS metrics and status checks for Provisioned IOPS SSD volumes	2012-10-01	Updated the EBS metrics to include two new metrics for Provisioned IOPS SSD volumes. For more information, see Amazon CloudWatch metrics for Amazon EBS (p. 1172) . Also added new status checks for Provisioned IOPS SSD volumes. For more information, see EBS volume status checks (p. 1051) .	20 November 2012
Linux Kernels		Updated AKI IDs; reorganized distribution kernels; updated PVOps section.	13 November 2012
M3 instances	2012-10-01	There are new M3 extra-large and M3 double-extra-large instance types. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instance Types .	31 October 2012
Spot Instance request status	2012-10-01	Spot Instance request status makes it easy to determine the state of your Spot requests.	14 October 2012
New Amazon Linux AMI release		Amazon Linux AMI 2012.09 is released.	11 October 2012
Amazon EC2 Reserved Instance Marketplace	2012-08-15	The Reserved Instance Marketplace matches sellers who have Amazon EC2 Reserved Instances that they no longer need with buyers who are looking to purchase additional capacity. Reserved Instances bought and sold through the Reserved Instance Marketplace work like any other Reserved Instances, except that they can have less than a full standard term remaining and can be sold at different prices.	11 September 2012
Provisioned IOPS SSD for Amazon EBS	2012-07-20	Provisioned IOPS SSD volumes deliver predictable, high performance for I/O intensive workloads, such as database applications, that rely on consistent and fast response times. For more information, see Amazon EBS volume types (p. 1023) .	31 July 2012
High I/O instances for Amazon EC2	2012-06-15	High I/O instances provides very high, low latency, disk I/O performance using SSD-based local instance storage.	18 July 2012

Feature	API version	Description	Release date
IAM roles on Amazon EC2 instances	2012-06-01	<p>IAM roles for Amazon EC2 provide:</p> <ul style="list-style-type: none"> • AWS access keys for applications running on Amazon EC2 instances. • Automatic rotation of the AWS access keys on the Amazon EC2 instance. • Granular permissions for applications running on Amazon EC2 instances that make requests to your AWS services. 	11 June 2012
Spot Instance features that make it easier to get started and handle the potential of interruption.		<p>You can now manage your Spot Instances as follows:</p> <ul style="list-style-type: none"> • Specify the amount you are willing to pay for Spot Instances using Auto Scaling launch configurations, and set up a schedule for specifying the amount you are willing to pay for Spot Instances. For more information, see Launching Spot Instances in Your Auto Scaling Group in the <i>Amazon EC2 Auto Scaling User Guide</i>. • Get notifications when instances are launched or terminated. • Use AWS CloudFormation templates to launch Spot Instances in a stack with AWS resources. 	7 June 2012
EC2 instance export and timestamps for status checks for Amazon EC2	2012-05-01	Added support for timestamps on instance status and system status to indicate the date and time that a status check failed.	25 May 2012
EC2 instance export, and timestamps in instance and system status checks for Amazon VPC	2012-05-01	<p>Added support for EC2 instance export to Citrix Xen, Microsoft Hyper-V, and VMware vSphere.</p> <p>Added support for timestamps in instance and system status checks.</p>	25 May 2012
Cluster Compute Eight Extra Large instances	2012-04-01	Added support for cc2.8xlarge instances in a VPC.	26 April 2012
AWS Marketplace AMIs	2012-04-01	Added support for AWS Marketplace AMIs.	19 April 2012
New Linux AMI release		Amazon Linux AMI 2012.03 is released.	28 March 2012
New AKI version		We've released AKI version 1.03 and AKIs for the AWS GovCloud (US) region.	28 March 2012
Medium instances, support for 64-bit on all AMIs, and a Java-based SSH Client	2011-12-15	Added support for a new instance type and 64-bit information. Added procedures for using the Java-based SSH client to connect to Linux instances.	7 March 2012

Feature	API version	Description	Release date
Reserved Instance pricing tiers	2011-12-15	Added a new section discussing how to take advantage of the discount pricing that is built into the Reserved Instance pricing tiers.	5 March 2012
Elastic Network Interfaces (ENIs) for EC2 instances in Amazon Virtual Private Cloud	2011-12-01	Added new section about elastic network interfaces (ENIs) for EC2 instances in a VPC. For more information, see Elastic network interfaces (p. 789) .	21 December 2011
New GRU Region and AKIs		Added information about the release of new AKIs for the SA-East-1 Region. This release deprecates the AKI version 1.01. AKI version 1.02 will continue to be backward compatible.	14 December 2011
New offering types for Amazon EC2 Reserved Instances	2011-11-01	You can choose from a variety of Reserved Instance offerings that address your projected use of the instance.	01 December 2011
Amazon EC2 instance status	2011-11-01	You can view additional details about the status of your instances, including scheduled events planned by AWS that might have an impact on your instances. These operational activities include instance reboots required to apply software updates or security patches, or instance retirements required where there are hardware issues. For more information, see Monitoring the status of your instances (p. 697) .	16 November 2011
Amazon EC2 Cluster Compute Instance Type		Added support for Cluster Compute Eight Extra Large (cc2.8xlarge) to Amazon EC2.	14 November 2011
New PDX Region and AKIs		Added information about the release of new AKIs for the new US-West 2 Region.	8 November 2011
Spot Instances in Amazon VPC	2011-07-15	Added information about the support for Spot Instances in Amazon VPC. With this update, users can launch Spot Instances a virtual private cloud (VPC). By launching Spot Instances in a VPC, users of Spot Instances can enjoy the benefits of Amazon VPC.	11 October 2011
New Linux AMI release		Added information about the release of Amazon Linux AMI 2011.09. This update removes the beta tag from the Amazon Linux AMI, supports the ability to lock the repositories to a specific version, and provides for notification when updates are available to installed packages including security updates.	26 September 2011

Feature	API version	Description	Release date
Simplified VM import process for users of the CLI tools	2011-07-15	The VM Import process is simplified with the enhanced functionality of <code>ImportInstance</code> and <code>ImportVolume</code> , which now will perform the upload of the images into Amazon EC2 after creating the import task. In addition, with the introduction of <code>ResumeImport</code> , users can restart an incomplete upload at the point the task stopped.	15 September 2011
Support for importing in VHD file format		VM Import can now import virtual machine image files in VHD format. The VHD file format is compatible with the Citrix Xen and Microsoft Hyper-V virtualization platforms. With this release, VM Import now supports RAW, VHD and VMDK (VMware ESX-compatible) image formats. For more information, see the VM Import/Export User Guide .	24 August 2011
Update to the Amazon EC2 VM Import Connector for VMware vCenter		Added information about the 1.1 version of the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). This update includes proxy support for Internet access, better error handling, improved task progress bar accuracy, and several bug fixes.	27 June 2011
Enabling Linux AMI to run user-provided kernels		Added information about the AKI version change from 1.01 to 1.02. This version updates the PVGRUB to address launch failures associated with t1.micro Linux instances. For more information, see Enabling Your Own Linux Kernels (p. 193) .	20 June 2011
Spot Instances Availability Zone pricing changes	2011-05-15	Added information about the Spot Instances Availability Zone pricing feature. In this release, we've added new Availability Zone pricing options as part of the information returned when you query for Spot Instance requests and Spot price history. These additions make it easier to determine the price required to launch a Spot Instance into a particular Availability Zone.	26 May 2011
AWS Identity and Access Management		Added information about AWS Identity and Access Management (IAM), which enables users to specify which Amazon EC2 actions a user can use with Amazon EC2 resources in general. For more information, see Identity and access management for Amazon EC2 (p. 919) .	26 April 2011
Enabling Linux AMI to run user-provided kernels		Added information about enabling a Linux AMI to use PVGRUB Amazon Kernel Image (AKI) to run a user-provided kernel. For more information, see Enabling Your Own Linux Kernels (p. 193) .	26 April 2011

Feature	API version	Description	Release date
Dedicated instances		Launched within your Amazon Virtual Private Cloud (Amazon VPC), Dedicated Instances are instances that are physically isolated at the host hardware level. Dedicated Instances let you take advantage of Amazon VPC and the AWS cloud, with benefits including on-demand elastic provisioning and pay only for what you use, while isolating your Amazon EC2 compute instances at the hardware level. For more information, see Dedicated Instances (p. 469) .	27 March 2011
Reserved Instances updates to the AWS Management Console		Updates to the AWS Management Console make it easier for users to view their Reserved Instances and purchase additional Reserved Instances, including Dedicated Reserved Instances. For more information, see Reserved Instances (p. 308) .	27 March 2011
New Amazon Linux reference AMI		The new Amazon Linux reference AMI replaces the CentOS reference AMI. Removed information about the CentOS reference AMI, including the section named Correcting Clock Drift for Cluster Instances on CentOS 5.4 AMI.	15 March 2011
Metadata information	2011-01-01	Added information about metadata to reflect changes in the 2011-01-01 release. For more information, see Instance metadata and user data (p. 658) and Instance metadata categories (p. 677) .	11 March 2011
Amazon EC2 VM Import Connector for VMware vCenter		Added information about the Amazon EC2 VM Import Connector for VMware vCenter virtual appliance (Connector). The Connector is a plug-in for VMware vCenter that integrates with VMware vSphere Client and provides a graphical user interface that you can use to import your VMware virtual machines to Amazon EC2.	3 March 2011
Force volume detachment		You can now use the AWS Management Console to force the detachment of an Amazon EBS volume from an instance. For more information, see Detaching an Amazon EBS volume from a Linux instance (p. 1058) .	23 February 2011
Instance termination protection		You can now use the AWS Management Console to prevent an instance from being terminated. For more information, see Enabling termination protection (p. 609) .	23 February 2011
Correcting Clock Drift for Cluster Instances on CentOS 5.4 AMI		Added information about how to correct clock drift for cluster instances running on Amazon's CentOS 5.4 AMI.	25 January 2011

Feature	API version	Description	Release date
VM Import	2010-11-15	Added information about VM Import, which allows you to import a virtual machine or volume into Amazon EC2. For more information, see the VM Import/Export User Guide .	15 December 2010
Basic monitoring for instances	2010-08-31	Added information about basic monitoring for EC2 instances.	12 December 2010
Filters and Tags	2010-08-31	Added information about listing, filtering, and tagging resources. For more information, see Listing and filtering your resources (p. 1224) and Tagging your Amazon EC2 resources (p. 1229) .	19 September 2010
Idempotent Instance Launch	2010-08-31	Added information about ensuring idempotency when running instances. For more information, see Ensuring Idempotency in the Amazon EC2 API Reference .	19 September 2010
Micro instances	2010-06-15	Amazon EC2 offers the t1.micro instance type for certain types of applications. For more information, see Burstable performance instances (p. 219) .	8 September 2010
AWS Identity and Access Management for Amazon EC2		Amazon EC2 now integrates with AWS Identity and Access Management (IAM). For more information, see Identity and access management for Amazon EC2 (p. 919) .	2 September 2010
Cluster instances	2010-06-15	Amazon EC2 offers cluster compute instances for high-performance computing (HPC) applications. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instance Types .	12 July 2010
Amazon VPC IP Address Designation	2010-06-15	Amazon VPC users can now specify the IP address to assign an instance launched in a VPC.	12 July 2010
Amazon CloudWatch Monitoring for Amazon EBS Volumes		Amazon CloudWatch monitoring is now automatically available for Amazon EBS volumes. For more information, see Amazon CloudWatch metrics for Amazon EBS (p. 1172) .	14 June 2010
High-memory extra large instances	2009-11-30	Amazon EC2 now supports a High-Memory Extra Large (m2.xlarge) instance type. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instance Types .	22 February 2010