
Amazon Elastic Compute Cloud

User Guide for Linux Instances



Amazon Elastic Compute Cloud: User Guide for Linux Instances

Copyright © 2018 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 Availability Zones	6
Region and Availability Zone Concepts	7
Available Regions	8
Regions and Endpoints	9
Describing Your Regions and Availability Zones	9
Specifying the Region for a Resource	11
Launching Instances in an Availability Zone	12
Migrating an Instance to Another Availability Zone	13
Root Device Volume	13
Root Device Storage Concepts	14
Choosing an AMI by Root Device Type	15
Determining the Root Device Type of Your Instance	16
Changing the Root Device Volume to Persist	16
Setting Up	19
Sign Up for AWS	19
Create an IAM User	19
Create a Key Pair	21
Create a Virtual Private Cloud (VPC)	23
Create a Security Group	23
Getting Started	26
Overview	26
Prerequisites	27
Step 1: Launch an Instance	27
Step 2: Connect to Your Instance	28
Step 3: Clean Up Your Instance	29
Next Steps	29
Best Practices	30
Tutorials	32
Install a LAMP Server (Amazon Linux 2)	32
Step 1: Prepare the LAMP Server	32
Step 2: Test Your LAMP Server	35
Step 3: Secure the Database Server	37
Step 4: (Optional) Install phpMyAdmin	38
Troubleshooting	40
Related Topics	40
Install a LAMP Server (Amazon Linux AMI)	41
Troubleshooting	40
Related Topics	40
Tutorial: Hosting a WordPress Blog	51
Prerequisites	52
Install WordPress	52
Next Steps	58
Help! My Public DNS Name Changed and now my Blog is Broken	59
Tutorial: Configure Apache Web Server on Amazon Linux 2 to Use SSL/TLS	59

Prerequisites	60
Step 1: Enable SSL/TLS on the Server	60
Step 2: Obtain a CA-signed Certificate	62
Step 3: Test and Harden the Security Configuration	67
Troubleshooting	69
Appendix: Let's Encrypt with Certbot on Amazon Linux 2	70
Tutorial: Increase the Availability of Your Application	74
Prerequisites	74
Scale and Load Balance Your Application	75
Test Your Load Balancer	76
Tutorial: Remotely Manage Your Instances	77
Grant Your User Account Access to Systems Manager	77
Install the SSH Agent	78
Send a Command Using the EC2 Console	78
Send a Command Using AWS Tools for Windows PowerShell	79
Send a Command Using the AWS CLI	80
Related Content	80
Amazon Machine Images	82
Using an AMI	82
Creating Your Own AMI	82
Buying, Sharing, and Selling AMIs	83
Deregistering Your AMI	83
Amazon Linux 2 and Amazon Linux AMI	83
AMI Types	83
Launch Permissions	84
Storage for the Root Device	84
Virtualization Types	86
Finding a Linux AMI	87
Finding a Linux AMI Using the Amazon EC2 Console	88
Finding an AMI Using the AWS CLI	88
Shared AMIs	89
Finding Shared AMIs	89
Making an AMI Public	91
Sharing an AMI with Specific AWS Accounts	92
Using Bookmarks	94
Guidelines for Shared Linux AMIs	94
Paid AMIs	98
Selling Your AMI	99
Finding a Paid AMI	99
Purchasing a Paid AMI	99
Getting the Product Code for Your Instance	100
Using Paid Support	100
Bills for Paid and Supported AMIs	101
Managing Your AWS Marketplace Subscriptions	101
Creating an Amazon EBS-Backed Linux AMI	101
Overview of Creating Amazon EBS-Backed AMIs	102
Creating a Linux AMI from an Instance	102
Creating a Linux AMI from a Snapshot	104
Creating an Instance Store-Backed Linux AMI	105
Overview of the Creation Process for Instance Store-Backed AMIs	105
Prerequisites	106
Setting Up the AMI Tools	106
Creating an AMI from an Instance Store-Backed Instance	109
Converting to an Amazon EBS-Backed AMI	116
AMI Tools Reference	119
AMIs with Encrypted Snapshots	136
AMI Scenarios Involving Encrypted EBS Snapshots	136

Copying an AMI	139
Permissions for Copying an Instance Store-Backed AMI	139
Cross-Region AMI Copy	140
Cross-Account AMI Copy	141
Encryption and AMI Copy	141
Copying an AMI	142
Stopping a Pending AMI Copy Operation	143
Deregistering Your Linux AMI	144
Cleaning Up Your Amazon EBS-Backed AMI	144
Cleaning Up Your Instance Store-Backed AMI	145
Amazon Linux	146
Connecting to an Amazon Linux Instance	146
Identifying Amazon Linux Images	146
AWS Command Line Tools	148
Package Repository	148
Extras Library (Amazon Linux 2)	150
Accessing Source Packages for Reference	151
cloud-init	151
Subscribing to Amazon Linux Notifications	153
Running Amazon Linux 2 as a Virtual Machine On-Premises	153
User Provided Kernels	157
HVM AMIs (GRUB)	157
Paravirtual AMIs (PV-GRUB)	158
Instances	163
Instance Types	163
Available Instance Types	164
Hardware Specifications	165
Virtualization Types	166
Networking and Storage Features	166
Instance Limits	168
T2 Instances	168
General Purpose Instances	190
Compute Optimized Instances	194
Memory Optimized Instances	197
Storage Optimized Instances	202
Accelerated Computing Instances	208
T1 Micro Instances	217
Changing the Instance Type	228
Optimizing CPU Options	232
Instance Purchasing Options	238
Determining the Instance Lifecycle	239
Reserved Instances	240
Scheduled Instances	274
Spot Instances	278
Dedicated Hosts	332
Dedicated Instances	345
Instance Lifecycle	350
Instance Launch	351
Instance Stop and Start (Amazon EBS-Backed Instances Only)	352
Instance Reboot	352
Instance Retirement	352
Instance Termination	353
Differences Between Reboot, Stop, and Terminate	353
Launch	354
Connect	397
Stop and Start	415
Reboot	418

Retire	419
Terminate	421
Recover	426
Configure Instances	427
Common Configuration Scenarios	427
Managing Software	428
Managing Users	432
Processor State Control	434
Setting the Time	439
Changing the Hostname	444
Setting Up Dynamic DNS	447
Running Commands at Launch	448
Instance Metadata and User Data	454
Identify Instances	469
Inspecting the System UUID	469
Inspecting the Instance Identity Document	470
Monitoring	471
Automated and Manual Monitoring	472
Automated Monitoring Tools	472
Manual Monitoring Tools	473
Best Practices for Monitoring	473
Monitoring the Status of Your Instances	474
Instance Status Checks	474
Scheduled Events	479
Monitoring Your Instances Using CloudWatch	483
Enable Detailed Monitoring	483
List Available Metrics	485
Get Statistics for Metrics	493
Graph Metrics	500
Create an Alarm	501
Create Alarms That Stop, Terminate, Reboot, or Recover an Instance	502
Automating Amazon EC2 with CloudWatch Events	511
Monitoring Memory and Disk Metrics	511
New CloudWatch Agent Available	511
CloudWatch Monitoring Scripts	512
Network and Security	521
Key Pairs	521
Creating a Key Pair Using Amazon EC2	523
Importing Your Own Public Key to Amazon EC2	523
Retrieving the Public Key for Your Key Pair on Linux	525
Retrieving the Public Key for Your Key Pair on Windows	525
Retrieving the Public Key for Your Key Pair From Your Instance	525
Verifying Your Key Pair's Fingerprint	526
Deleting Your Key Pair	526
Adding or Replacing a Key Pair for Your Instance	527
Connecting to Your Linux Instance if You Lose Your Private Key	528
Security Groups	531
Security Groups for EC2-Classic	531
Security Groups for EC2-VPC	531
Security Group Rules	532
Default Security Groups	534
Custom Security Groups	535
Working with Security Groups	535
Security Group Rules Reference	540
Controlling Access	545
Network Access to Your Instance	546
Amazon EC2 Permission Attributes	546

IAM and Amazon EC2	546
IAM Policies	548
IAM Roles	620
Network Access	628
Amazon VPC	630
Benefits of Using a VPC	631
Differences Between EC2-Classic and EC2-VPC	631
Sharing and Accessing Resources Between EC2-Classic and EC2-VPC	634
Instance Types Available Only in a VPC	636
Amazon VPC Documentation	636
Supported Platforms	637
ClassicLink	638
Migrating from EC2-Classic to a VPC	648
Instance IP Addressing	658
Private IPv4 Addresses and Internal DNS Hostnames	659
Public IPv4 Addresses and External DNS Hostnames	660
Elastic IP Addresses (IPv4)	661
Amazon DNS Server	661
IPv6 Addresses	661
IP Address Differences Between EC2-Classic and EC2-VPC	662
Working with IP Addresses for Your Instance	663
Multiple IP Addresses	667
Elastic IP Addresses	674
Elastic IP Address Basics	675
Elastic IP Address Differences for EC2-Classic and EC2-VPC	675
Working with Elastic IP Addresses	677
Using Reverse DNS for Email Applications	682
Elastic IP Address Limit	683
Network Interfaces	683
Network Interface Basics	684
IP Addresses Per Network Interface Per Instance Type	684
Scenarios for Network Interfaces	689
Best Practices for Configuring Network Interfaces	691
Working with Network Interfaces	692
Requester-Managed Network Interfaces	700
Placement Groups	701
Cluster Placement Groups	702
Spread Placement Groups	702
Placement Group Rules and Limitations	702
Creating a Placement Group	703
Launching Instances in a Placement Group	704
Changing the Placement Group for an Instance	704
Deleting a Placement Group	705
Network MTU	706
Jumbo Frames (9001 MTU)	706
Path MTU Discovery	706
Check the Path MTU Between Two Hosts	707
Check and Set the MTU on Your Linux Instance	707
Troubleshooting	708
Enhanced Networking	708
Enhanced Networking Types	709
Enabling Enhanced Networking on Your Instance	709
Enhanced Networking: Intel 82599 VF	709
Enhanced Networking: ENA	716
Troubleshooting ENA	723
Storage	730
Amazon EBS	731

Features of Amazon EBS	732
EBS Volumes	733
EBS Snapshots	792
EBS Optimization	812
EBS Encryption	819
EBS Volumes and NVMe	823
EBS Performance	824
EBS CloudWatch Events	840
Instance Store	847
Instance Store Lifetime	848
Instance Store Volumes	848
Add Instance Store Volumes	852
SSD Instance Store Volumes	854
Instance Store Swap Volumes	856
Optimizing Disk Performance	858
Amazon EFS	858
Prerequisites	859
Step 1: Create an EFS File System	859
Step 2: Mount the File System	860
Step 3: Test the File System	861
Step 4: Clean Up	861
Amazon S3	862
Amazon S3 and Amazon EC2	862
Instance Volume Limits	863
Linux-Specific Volume Limits	863
Windows-Specific Volume Limits	864
Instance Type Limits	864
Bandwidth versus Capacity	864
Device Naming	864
Available Device Names	865
Device Name Considerations	865
Block Device Mapping	866
Block Device Mapping Concepts	866
AMI Block Device Mapping	869
Instance Block Device Mapping	871
Using Public Data Sets	875
Public Data Set Concepts	875
Finding Public Data Sets	875
Creating a Public Data Set Volume from a Snapshot	876
Attaching and Mounting the Public Data Set Volume	876
Resources and Tags	878
Resource Locations	878
Resource IDs	879
Working with Longer IDs	880
Controlling Access to Longer ID Settings	883
Listing and Filtering Your Resources	884
Advanced Search	884
Listing Resources Using the Console	885
Filtering Resources Using the Console	886
Listing and Filtering Using the CLI and API	887
Tagging Your Resources	887
Tag Basics	888
Tagging Your Resources	889
Tag Restrictions	891
Tagging Your Resources for Billing	891
Working with Tags Using the Console	892
Working with Tags Using the CLI or API	895

Service Limits	896
Viewing Your Current Limits	897
Requesting a Limit Increase	897
Usage Reports	898
EC2Rescue for Linux	899
Installing EC2Rescue for Linux	899
Working with EC2Rescue for Linux	900
Getting Help	900
Running a Module	900
Uploading the Results	901
Creating Backups	901
Developing EC2Rescue Modules	902
Adding Module Attributes	902
Adding Environment Variables	904
Using YAML Syntax	905
Example Modules	906
Troubleshooting	907
Troubleshooting Launch Issues	907
Instance Limit Exceeded	907
Insufficient Instance Capacity	908
Instance Terminates Immediately	908
Connecting to Your Instance	909
Error connecting to your instance: Connection timed out	910
Error: User key not recognized by server	911
Error: Host key not found, Permission denied (publickey), or Authentication failed, permission denied	913
Error: Unprotected Private Key File	914
Error: Server refused our key or No supported authentication methods available	914
Error Using MindTerm on Safari Browser	915
Error Using macOS RDP Client	915
Cannot Ping Instance	915
Stopping Your Instance	916
Creating a Replacement Instance	916
Terminating Your Instance	917
Delayed Instance Termination	917
Terminated Instance Still Displayed	918
Automatically Launch or Terminate Instances	918
Instance Recovery Failures	918
Failed Status Checks	918
Initial Steps	919
Retrieving System Logs	920
Troubleshooting System Log Errors for Linux-Based Instances	920
Out of memory: kill process	921
ERROR: mmu_update failed (Memory management update failed)	922
I/O Error (Block Device Failure)	922
I/O ERROR: neither local nor remote disk (Broken distributed block device)	923
request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)	924
"FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" (Kernel and AMI mismatch)	925
"FATAL: Could not load /lib/modules" or "BusyBox" (Missing kernel modules)	926
ERROR Invalid kernel (EC2 incompatible kernel)	927
request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)	928
fsck: No such file or directory while trying to open... (File system not found)	929
General error mounting filesystems (Failed mount)	930
VFS: Unable to mount root fs on unknown-block (Root filesystem mismatch)	932

Error: Unable to determine major/minor number of root device... (Root file system/device mismatch)	933
XENBUS: Device with no driver...	934
... days without being checked, check forced (File system check required)	934
fsck died with exit status... (Missing device)	935
GRUB prompt (grubdom>)	936
Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address)	938
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration)	939
XENBUS: Timeout connecting to devices (Xenbus timeout)	940
Getting Console Output and Rebooting Instances	940
Instance Reboot	941
Instance Console Output	941
Capture a Screenshot of an Unreachable Instance	941
Instance Recovery When a Host Computer Fails	942
Booting from the Wrong Volume	943
Document History	945
AWS Glossary	964

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 \(p. 19\)](#)
- [Getting Started with Amazon EC2 Linux Instances \(p. 26\)](#)

Basics

- [Instances and AMIs \(p. 4\)](#)
- [Regions and Availability Zones \(p. 6\)](#)
- [Instance Types \(p. 163\)](#)
- [Tags \(p. 887\)](#)

Networking and Security

- [Amazon EC2 Key Pairs \(p. 521\)](#)
- [Security Groups \(p. 531\)](#)
- [Elastic IP Addresses \(p. 674\)](#)
- [Amazon EC2 and Amazon VPC \(p. 630\)](#)

Storage

- [Amazon EBS \(p. 731\)](#)
- [Instance Store \(p. 847\)](#)

Working with Linux Instances

- [Remote Management \(Run Command\)](#)
- [Tutorial: Install a LAMP Web Server on Amazon Linux 2 \(p. 32\)](#)
- [Tutorial: Configure Apache Web Server on Amazon Linux 2 to Use SSL/TLS \(p. 59\)](#)
- [Getting Started with AWS: Hosting a Web App for Linux](#)

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 [Elastic Load Balancing User Guide](#).

To monitor basic statistics for your instances and Amazon EBS volumes, use Amazon CloudWatch. For more information, see the [Amazon CloudWatch User Guide](#).

To automate actions, such as activating a Lambda function whenever a new Amazon EC2 instance starts, or invoking SSM Run Command whenever an event in another AWS service happens, use Amazon CloudWatch Events. For more information, see the [Amazon CloudWatch Events User Guide](#).

To monitor the calls made to the Amazon EC2 API for your account, including calls made by the AWS Management Console, command line tools, and other services, use AWS CloudTrail. For more information, see the [AWS CloudTrail 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 [Amazon Relational Database Service Developer Guide](#).

To import virtual machine (VM) images from your local environment into AWS and convert them into ready-to-use AMIs or instances, use VM Import/Export. For more information, see the [VM Import/Export 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 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.

Reserved Instances

Make a low, one-time, up-front payment for an instance, reserve it for a one- or three-year term, and pay a significantly lower hourly rate for these instances.

Spot Instances

Request unused EC2 instances, which can lower your costs significantly.

For a complete list of charges and specific 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 your [AWS Account Activity page](#). 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](#).

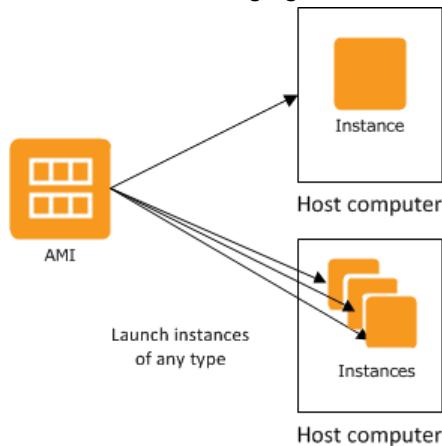
For an overview of Trusted Advisor, a service that helps you optimize the costs, security, and performance of your AWS environment, see [AWS Trusted Advisor](#).

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

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 \(p. 13\)](#).

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 \(p. 866\)](#). 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 \(p. 730\)](#).

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 [Controlling Access to Amazon EC2 Resources \(p. 545\)](#).
- 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 \(p. 531\)](#).
- 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 \(p. 95\)](#). For more information about sharing AMIs safely, see [Shared AMIs \(p. 89\)](#).

Stopping, Starting, and Terminating Instances

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 \(p. 350\)](#).

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 \(p. 84\)](#).

Regions and Availability Zones

Amazon EC2 is hosted in multiple locations world-wide. These locations are composed of regions and Availability Zones. Each *region* is a separate geographic area. Each region has multiple, isolated locations known as *Availability Zones*. Amazon EC2 provides you the ability to place resources, such as instances, and data in multiple locations. Resources aren't replicated across regions unless you do so specifically.

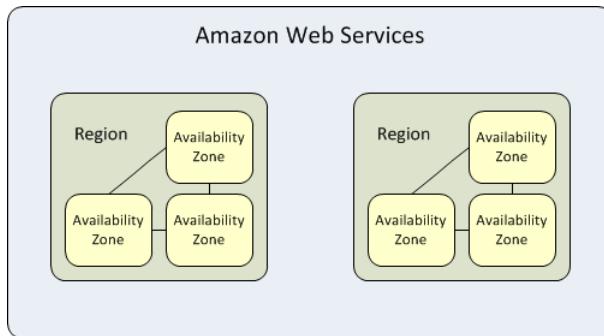
Amazon operates state-of-the-art, highly-available data centers. Although rare, failures can occur that affect the availability of instances that are in the same location. If you host all your instances in a single location that is affected by such a failure, none of your instances would be available.

Contents

- [Region and Availability Zone Concepts \(p. 7\)](#)
- [Available Regions \(p. 8\)](#)
- [Regions and Endpoints \(p. 9\)](#)
- [Describing Your Regions and Availability Zones \(p. 9\)](#)
- [Specifying the Region for a Resource \(p. 11\)](#)
- [Launching Instances in an Availability Zone \(p. 12\)](#)
- [Migrating an Instance to Another Availability Zone \(p. 13\)](#)

Region and Availability Zone Concepts

Each region is completely independent. Each Availability Zone is isolated, but the Availability Zones in a region are connected through low-latency links. The following diagram illustrates the relationship between regions and Availability Zones.



Amazon EC2 resources are either global, tied to a region, or tied to an Availability Zone. For more information, see [Resource Locations \(p. 878\)](#).

Regions

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

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

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 \(p. 139\)](#).

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

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.

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 \(p. 674\)](#).

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 identifiers for each account. For example, your Availability Zone us-east-1a might not be the same location as us-east-1a for another account. There's no way for you to coordinate Availability Zones between accounts.

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 customers. Therefore, your account might have a different number of available Availability Zones in a region than another account.

You can list the Availability Zones that are available to your account. For more information, see [Describing Your Regions and Availability Zones \(p. 9\)](#).

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) account provides access to the AWS GovCloud (US) region only. For more information, see [AWS GovCloud \(US\) Region](#).
- An Amazon AWS (China) account provides access to the China (Beijing) region only.

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) or China (Beijing).

Code	Name
us-east-1	US East (N. Virginia)
us-east-2	US East (Ohio)
us-west-1	US West (N. California)
us-west-2	US West (Oregon)
ca-central-1	Canada (Central)
eu-central-1	EU (Frankfurt)
eu-west-1	EU (Ireland)
eu-west-2	EU (London)
eu-west-3	EU (Paris)
ap-northeast-1	Asia Pacific (Tokyo)
ap-northeast-2	Asia Pacific (Seoul)
ap-northeast-3	Asia Pacific (Osaka-Local)

Code	Name
ap-southeast-1	Asia Pacific (Singapore)
ap-southeast-2	Asia Pacific (Sydney)
ap-south-1	Asia Pacific (Mumbai)
sa-east-1	South America (São Paulo)

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 and Availability Zones \(p. 9\)](#).

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 [Regions and Endpoints](#) in the *Amazon Web Services General Reference*.

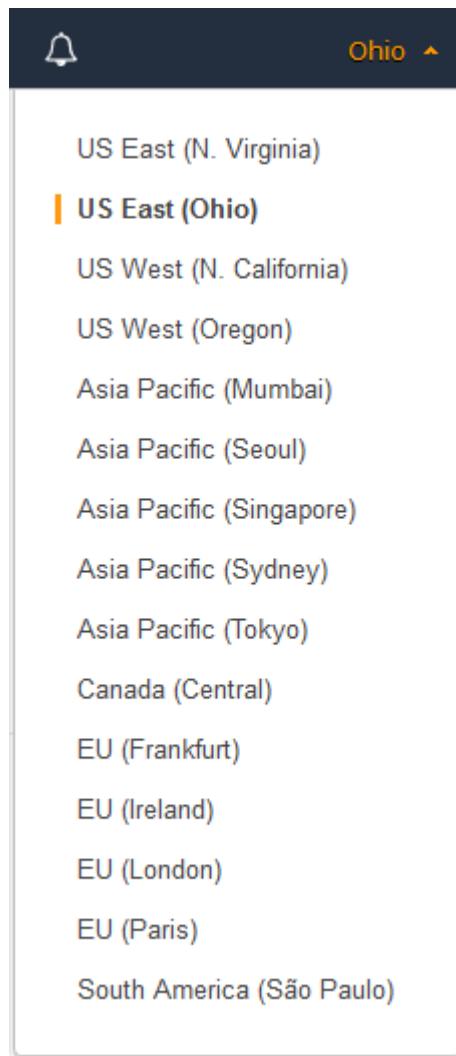
For more information about endpoints and protocols in AWS GovCloud (US), see [AWS GovCloud \(US\) Endpoints](#) in the *AWS GovCloud (US) User Guide*.

Describing Your Regions and Availability Zones

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

To find your regions and 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, Availability Zone Status**.

To find your regions and Availability Zones using the command line

1. [AWS CLI] Use the [describe-regions](#) command as follows to describe the regions for your account.

```
aws ec2 describe-regions
```

2. [AWS CLI] 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
```

3. [AWS Tools for Windows PowerShell] Use the [Get-EC2Region](#) command as follows to describe the regions for your account.

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

4. [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
```

Specifying the Region for a Resource

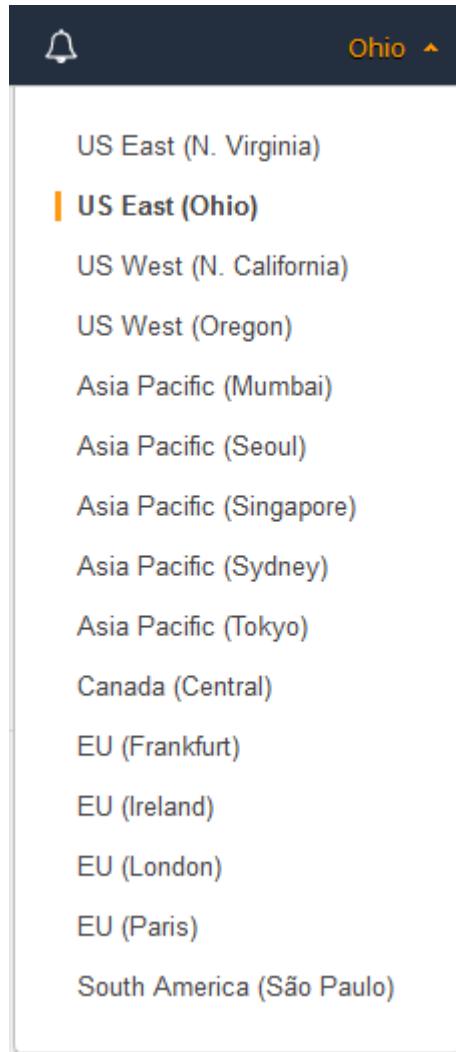
Every time you create an Amazon EC2 resource, you can specify the region for the resource. You can specify the region for a resource using the AWS Management Console or the command line.

Note

Some AWS resources might not be available in all regions and Availability Zones. Ensure that you can create the resources you need in the desired regions or Availability Zone before launching an instance in a specific Availability Zone.

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](#).

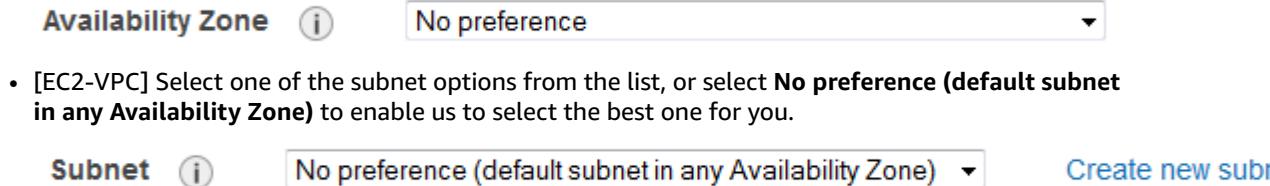
Launching Instances in an Availability Zone

When you launch an instance, select a region that puts your instances closer to specific customers, or meets the legal or other requirements 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 one for you. When you launch your initial instances, we recommend that you accept the default Availability Zone, because this enables us to select the best Availability Zone for you based on system health and available capacity. If you launch additional instances, only specify an Availability Zone if your new instances must be close to, or separated from, your running instances.

To specify an Availability Zone for your instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the dashboard, choose **Launch Instance**.
3. Follow the directions for the wizard. On the **Configure Instance Details** page, do the following:
 - [EC2-Classic] Select one of the Availability Zone options from the list, or select **No Preference** to enable us to select the best one for you.
 - [EC2-VPC] Select one of the subnet options from the list, or select **No preference (default subnet in any Availability Zone)** to enable us to select the best one for you.



To specify an Availability Zone for your instance using the AWS CLI

You can use the `run-instances` command with one of the following options:

- [EC2-Classic] `--placement`
- [EC2-VPC] `--subnet-id`

To specify an Availability Zone for your instance using the AWS Tools for Windows PowerShell

You can use the `New-EC2Instance` command with one of the following options:

- [EC2-Classic] –AvailabilityZone
- [EC2-VPC] –SubnetId

Migrating an Instance to Another Availability Zone

If you need to, you can migrate an instance from one Availability Zone to another. For example, if 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, you could migrate the instance to an Availability Zone where we can 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, and 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 the 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 \(p. 101\)](#)
 - [Creating an Instance Store-Backed Linux AMI \(p. 105\)](#)
 - [Creating an Amazon EBS-Backed Windows AMI](#)
2. [EC2-VPC] 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 the instances in your subnet so that you can move all instances in 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 \(p. 12\)](#).
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 and Reassociating with a Different Instance \(p. 679\)](#).
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 \(p. 268\)](#).
6. (Optional) Terminate the original instance. For more information, see [Terminating an Instance \(p. 422\)](#).

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.

For more information about the device names Amazon EC2 uses for your root volumes, see [Device Naming on Linux Instances \(p. 864\)](#).

Topics

- [Root Device Storage Concepts \(p. 14\)](#)
- [Choosing an AMI by Root Device Type \(p. 15\)](#)
- [Determining the Root Device Type of Your Instance \(p. 16\)](#)
- [Changing the Root Device Volume to Persist \(p. 16\)](#)

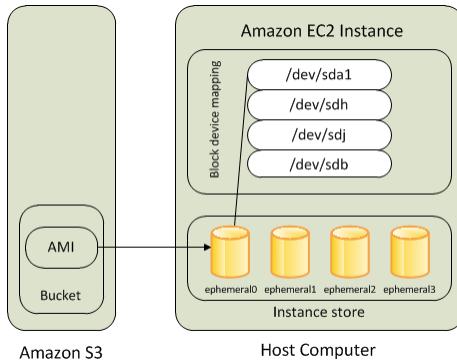
Root Device Storage Concepts

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 \(p. 84\)](#).

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).

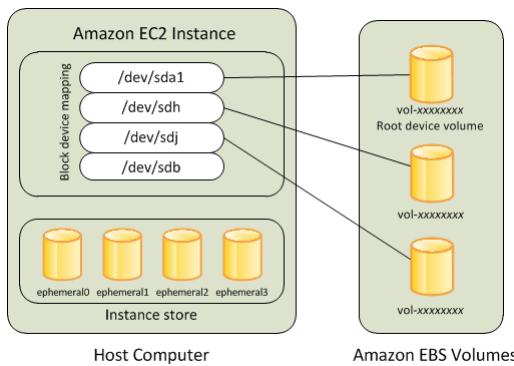


After an instance store-backed instance fails or terminates, it cannot be restored. If you plan to use Amazon EC2 instance store-backed instances, we highly recommend that you distribute the data on your instance stores across multiple Availability Zones. You should also back up critical data from your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

For more information, see [Amazon EC2 Instance Store \(p. 847\)](#).

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 \(p. 101\)](#).
- 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 \(p. 733\)](#).

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

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

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**, and select the instance.
3. Check the value of **Root device type** in the **Description** tab 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 Device Volume to Persist

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

Changing the Root Volume to Persist 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 device volume of an instance to persist at launch using the console

1. Open the Amazon EC2 console.
2. From the Amazon EC2 console 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, deselect **Delete On Termination** 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 of an Instance to Persist Using the AWS CLI

Using the AWS CLI, you can change the `DeleteOnTermination` attribute when you launch an instance or while the instance is running.

Example at Launch

Use the [run-instances](#) command to preserve the root volume by including a block device mapping that sets its `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  
    }  
  }  
]
```

You can confirm that `DeleteOnTermination` is `false` by using the [describe-instances](#) command and looking for the `BlockDeviceMappings` entry for the device in the command output, as shown here.

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

Example While the Instance is Running

Use the [modify-instance-attribute](#) command to preserve the root volume by including a block device mapping that sets its `DeleteOnTermination` attribute to `false`.

```
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  
    }  
  }  
]
```

Setting Up with Amazon EC2

If you've already signed up for Amazon Web Services (AWS), you can start using Amazon EC2 immediately. You can open the Amazon EC2 console, choose **Launch Instance**, and follow the steps in the launch wizard to launch your first instance.

If you haven't signed up for AWS yet, or if you need assistance launching your first instance, complete the following tasks to get set up to use Amazon EC2:

1. [Sign Up for AWS \(p. 19\)](#)
2. [Create an IAM User \(p. 19\)](#)
3. [Create a Key Pair \(p. 21\)](#)
4. [Create a Virtual Private Cloud \(VPC\) \(p. 23\)](#)
5. [Create a Security Group \(p. 23\)](#)

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://aws.amazon.com/>, and then choose **Create an AWS Account**.

Note

This might be unavailable in your browser if you previously signed into the AWS Management Console. In that case, choose **Sign in to a different account**, and then choose **Create a new AWS account**.

2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Note your AWS account number, because you'll need it for the next task.

Create an IAM User

Services in AWS, such as Amazon EC2, require that you provide credentials when you access them, so that the service can determine whether you have permission to access its resources. The console requires your password. You can create access keys for your AWS account to access the command line interface or API. However, we don't recommend that you access AWS using the credentials for your AWS account; we recommend that you use AWS Identity and Access Management (IAM) instead. Create an IAM user, and then add the user to an IAM group with administrative permissions or grant this user administrative permissions. You can then access AWS using a special URL and the credentials for the IAM user.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM console. If you aren't familiar with using the console, see [Working with the AWS Management Console](#) for an overview.

To create an IAM user for yourself and add the user to an Administrators group

1. Use your AWS account email address and password to sign in as the [AWS account root user](#) to the IAM console at <https://console.aws.amazon.com/iam/>.

Note

We strongly recommend that you adhere to the best practice of using the **Administrator** IAM user below and securely lock away the root user credentials. Sign in as the root user only to perform a few [account and service management tasks](#).

2. In the navigation pane of the console, choose **Users**, and then choose **Add user**.
3. For **User name**, type **Administrator**.
4. Select the check box next to **AWS Management Console access**, select **Custom password**, and then type the new user's password in the text box. You can optionally select **Require password reset** to force the user to create a new password the next time the user signs in.
5. Choose **Next: Permissions**.
6. On the **Set permissions** page, choose **Add user to group**.
7. Choose **Create group**.
8. In the **Create group** dialog box, for **Group name** type **Administrators**.
9. For **Filter policies**, select the check box for **AWS managed - job function**.
10. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.
11. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
12. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

You can use this same process to create more groups and users, and to give your users access to your AWS account resources. To learn about using policies to restrict users' permissions to specific AWS resources, go to [Access Management](#) and [Example Policies](#).

To sign in as this new IAM user, sign out of the AWS console, then use the following URL, where *your_aws_account_id* is your AWS account number without the hyphens (for example, if your AWS account number is 1234-5678-9012, your AWS account ID is 123456789012):

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

Enter the IAM user name (not your email address) and password that you just created. When you're signed in, the navigation bar displays "*your_user_name* @ *your_aws_account_id*".

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. From the IAM console, choose **Dashboard** in the navigation pane. From the dashboard, choose **Customize** and enter an alias such as your company name. To sign in after you create an account alias, use the following URL:

```
https://your_account_alias.signin.aws.amazon.com/console/
```

To verify the sign-in link for IAM users for your account, open the IAM console and check under **IAM users sign-in link** on the dashboard.

For more information about IAM, see [IAM and Amazon EC2 \(p. 546\)](#).

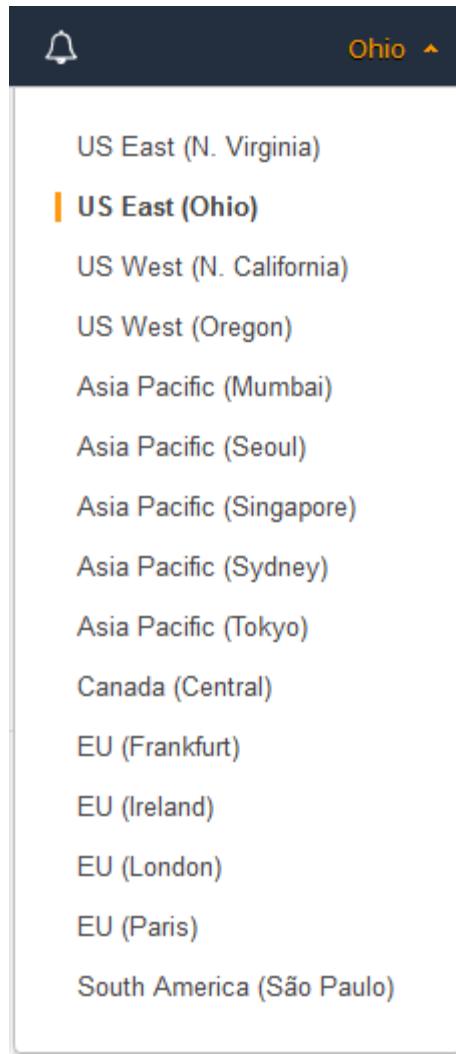
Create a Key Pair

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 Availability Zones \(p. 6\)](#).

To create a key pair

1. Sign in to AWS using the URL that you created in the previous section.
2. From the AWS dashboard, choose **EC2** to open the Amazon EC2 console.
3. From the navigation bar, select a region for the key pair. You can select any region that's available to you, regardless of your location. However, key pairs are specific to a region; for example, if you plan to launch an instance in the US East (Ohio) Region, you must create a key pair for the instance in the US East (Ohio) Region.



4. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.

Tip

The navigation pane is on the left side of the console. If you do not see the pane, it might be minimized; choose the arrow to expand the pane. You may have to scroll down to see the **Key Pairs** link.

 **NETWORK & SECURITY**

- Security Groups**
- Elastic IPs**
- Placement Groups**
- Key Pairs** 
- Network Interfaces**

5. Choose **Create Key Pair**.
6. Enter a name for the new key pair in the **Key pair name** field of the **Create Key Pair** dialog box, and then choose **Create**. Use a name that is easy for you to remember, such as your IAM user name, followed by `-key-pair`, plus the region name. For example, `me-key-pair-useast2`.
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 `.pem`. Save the private key file in a safe place.

Important

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

8. If you will use an SSH client on a Mac 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 your_user_name-key-pair-region_name.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 \(p. 914\)](#).

For more information, see [Amazon EC2 Key Pairs \(p. 521\)](#).

To connect to your instance using your key pair

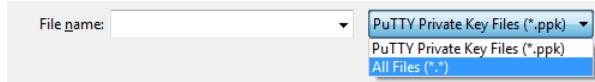
To connect to your Linux instance from a computer running Mac or Linux, you'll specify the `.pem` file to your SSH client with the `-i` option and the path to your private key. To connect to your Linux instance from a computer running Windows, you can use either MindTerm or PuTTY. If you plan to use PuTTY, you'll need to install it and use the following procedure to convert the `.pem` file to a `.ppk` file.

(Optional) To prepare to connect to a Linux instance from Windows using PuTTY

1. Download and install PuTTY from <http://www.chiark.greenend.org.uk/~sgtatham/putty/>. Be sure to install the entire suite.
2. Start PuTTYgen (for example, from the **Start** menu, choose **All Programs > PuTTY > PuTTYgen**).
3. Under **Type of key to generate**, choose **RSA**.



4. Choose **Load**. By default, PuTTYgen displays only files with the extension .ppk. To locate your .pem file, select the option to display files of all types.



5. Select the private key file that you created in the previous procedure and choose **Open**. Choose **OK** to dismiss the confirmation dialog box.
6. Choose **Save private key**. PuTTYgen displays a warning about saving the key without a passphrase. Choose **Yes**.
7. Specify the same name for the key that you used for the key pair. PuTTY automatically adds the .ppk file extension.

Create a Virtual Private Cloud (VPC)

Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. If you have a default VPC, you can skip this section and move to the next task, [Create a Security Group \(p. 23\)](#). To determine whether you have a default VPC, see [Supported Platforms in the Amazon EC2 Console \(p. 637\)](#). Otherwise, you can create a nondefault VPC in your account using the steps below.

Important

If your account supports EC2-Classic in a region, then you do not have a default VPC in that region. T2 instances must be launched into a VPC.

To create a nondefault VPC

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. From the navigation bar, select a region for the VPC. VPCs are specific to a region, so you should select the same region in which you created your key pair.
3. On the VPC dashboard, choose **Create VPC**.
4. On the **Step 1: Select a VPC Configuration** page, ensure that **VPC with a Single Public Subnet** is selected, and choose **Select**.
5. On the **Step 2: VPC with a Single Public Subnet** page, enter a friendly name for your VPC in the **VPC name** field. Leave the other default configuration settings, and choose **Create VPC**. On the confirmation page, choose **OK**.

For more information about Amazon VPC, see [What is Amazon VPC?](#) in the *Amazon VPC User Guide*.

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 Availability Zones \(p. 6\)](#).

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.

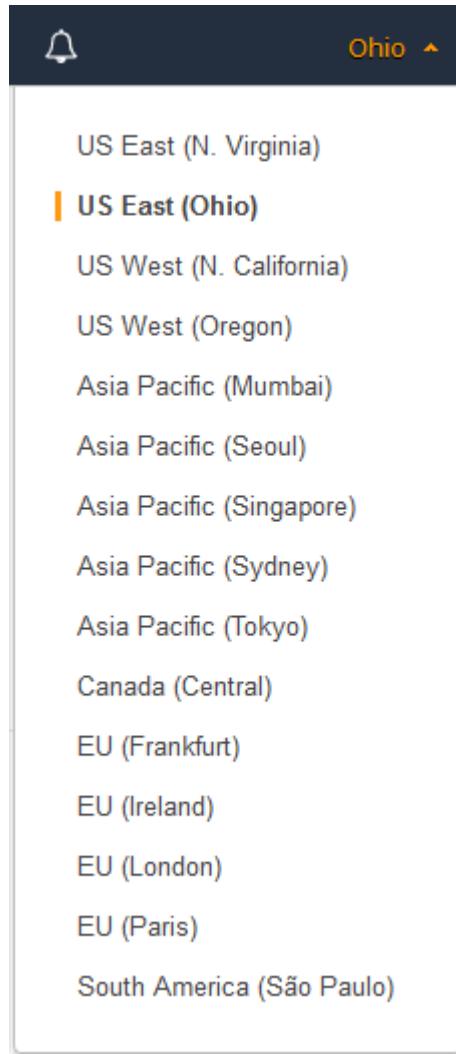
To create a security group with least privilege

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

Tip

Alternatively, you can use the Amazon VPC console to create a security group. However, the instructions in this procedure don't match the Amazon VPC console. Therefore, if you switched to the Amazon VPC console in the previous section, either switch back to the Amazon EC2 console and use these instructions, or use the instructions in [Set Up a Security Group for Your VPC](#) in the *Amazon VPC Getting Started Guide*.

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. Choose **Security Groups** in the navigation pane.

4. Choose **Create Security Group**.
5. Enter a name for the new security group and a description. Use a name that is easy for you to remember, such as your IAM user name, followed by `_SG_`, plus the region name. For example, `me_SG_uswest2`.
6. In the **VPC** list, select your VPC. If you have a default VPC, it's the one that is marked with an asterisk (*).

Note

If your account supports EC2-Classic, select the VPC that you created in the previous task.

7. On the **Inbound** tab, create the following rules (choose **Add Rule** for each new rule), and then choose **Create**:

- 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.

For more information, see [Amazon EC2 Security Groups for Linux Instances \(p. 531\)](#).

Getting Started with Amazon EC2 Linux Instances

Let's get started with Amazon Elastic Compute Cloud (Amazon EC2) by launching, connecting to, and using 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.

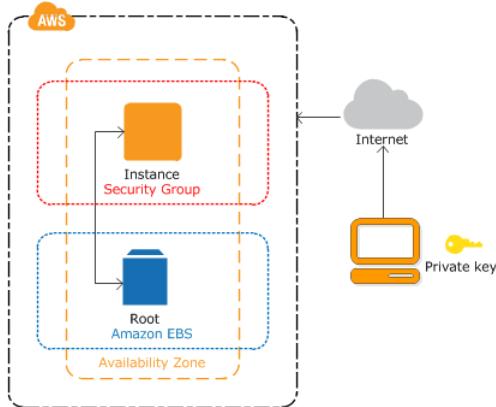
When you sign up for AWS, you can get started with Amazon EC2 using the [AWS Free Tier](#). If you created your AWS account less than 12 months ago, and have not already exceeded the free tier benefits for Amazon EC2, it will not cost you anything to complete this tutorial, because we help you select 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 \(p. 26\)](#)
- [Prerequisites \(p. 27\)](#)
- [Step 1: Launch an Instance \(p. 27\)](#)
- [Step 2: Connect to Your Instance \(p. 28\)](#)
- [Step 3: Clean Up Your Instance \(p. 29\)](#)
- [Next Steps \(p. 29\)](#)

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 \(p. 27\)](#)

2. Connect to Your Instance (p. 28)
3. Clean Up Your Instance (p. 29)

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 \(p. 19\)](#).

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](#).

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` type, which is selected by default. Notice that this instance type is eligible for the free tier.

Note

`T2 instances`, such as `t2.micro`, must be launched into a VPC. If your AWS account supports EC2-Classic and you do not have a VPC in the selected region, the launch wizard creates a VPC for you and you can continue to the next step. Otherwise, the **Review and Launch** button is disabled and you must choose **Next: Configure Instance Details** and follow the directions to select a subnet.

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.

Alternatively, you can create a new key pair. Select **Create a new key pair**, enter a name for the key pair, and then choose **Download Key Pair**. This is the only chance for you to save the private key file, so be sure to download it. 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.

Warning

Don't select the **Proceed without a key pair** option. 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 a Linux instance. In this procedure, you'll connect using your browser. Alternatively, you can connect using PuTTY or an SSH client. It's also assumed that you launched an Amazon Linux 2 instance, per the previous step, which has a specific user name. Other Linux distributions might use a different user name. For more information, see [Connecting to Your Linux Instance from Windows Using PuTTY \(p. 402\)](#) or [Connecting to Your Linux Instance Using SSH \(p. 397\)](#).

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. If you can't connect to your instance, see [Troubleshooting Connecting to Your Instance \(p. 909\)](#) for assistance.

To connect to your Linux instance using a web browser

1. You must have Java installed and enabled in the browser. If you don't have Java already, you can contact your system administrator to get it installed, or follow the steps outlined in the following pages: [Install Java](#) and [Enable Java in your web browser](#).
2. From the Amazon EC2 console, choose **Instances** in the navigation pane.
3. Select the instance, and then choose **Connect**.
4. Choose **A Java SSH client directly from my browser (Java required)**.
5. Amazon EC2 automatically detects the public DNS name of your instance and populates **Public DNS** for you. It also detects the key pair that you specified when you launched the instance. Complete the following, and then choose **Launch SSH Client**.
 - a. In **User name**, enter ec2-user.
 - b. In **Private key path**, enter the fully qualified path to your private key (.pem) file, including the key pair name.
 - c. (Optional) Choose **Store in browser cache** to store the location of the private key in your browser cache. This enables Amazon EC2 to detect the location of the private key in subsequent browser sessions, until you clear your browser's cache.

6. If necessary, choose **Yes** to trust the certificate, and choose **Run** to run the MindTerm client.
7. If this is your first time running MindTerm, a series of dialog boxes asks you to accept the license agreement, confirm setup for your home directory, and confirm setup of the known hosts directory. Confirm these settings.
8. A dialog prompts you to add the host to your set of known hosts. If you do not want to store the host key information on your local computer, choose **No**.

A window opens and you are connected to your instance.

Note

If you chose **No** in the previous step, you'll see the following message, which is expected:

Verification of server key disabled in this session.

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 \(p. 29\)](#).

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**.
3. Choose **Yes, 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 deleted.

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 [Tutorial: Remotely Manage Your Amazon EC2 Instances \(p. 77\)](#) and [Systems Manager Remote Management \(Run Command\)](#).
- 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 \(p. 747\)](#) and [Attaching an Amazon EBS Volume to an Instance \(p. 750\)](#).
- Install the LAMP stack. For more information, see [Tutorial: Install a LAMP Web Server on Amazon Linux 2 \(p. 32\)](#).

Best Practices for Amazon EC2

This checklist is intended to help you get the maximum benefit from and satisfaction with Amazon EC2.

Security and Network

- 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 \(p. 532\)](#).
- 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](#). For more information about updating your Windows instance, see [Updating Your Windows Instance](#) in the *Amazon EC2 User Guide for Windows Instances*.
- Launch your instances into a VPC instead of EC2-Classic. Note that if you created your AWS account after 2013-12-04, we automatically launch your instances into a VPC. For more information about the benefits, see [Amazon EC2 and Amazon Virtual Private Cloud \(p. 630\)](#).

Storage

- Understand the implications of the root device type for data persistence, backup, and recovery. For more information, see [Storage for the Root Device \(p. 84\)](#).
- 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 \(p. 424\)](#).
- 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.

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 \(p. 454\)](#) and [Tagging Your Amazon EC2 Resources \(p. 887\)](#).
- 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 Limits \(p. 896\)](#).

Backup and Recovery

- Regularly back up your EBS volumes using [Amazon EBS snapshots \(p. 792\)](#), and create an [Amazon Machine Image \(AMI\) \(p. 82\)](#) 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 \(p. 658\)](#).
- Monitor and respond to events. For more information, see [Monitoring Amazon EC2 \(p. 471\)](#).

- 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 \(p. 683\)](#). 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.

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 \(p. 32\)](#)
- [Tutorial: Install a LAMP Web Server with the Amazon Linux AMI \(p. 41\)](#)
- [Tutorial: Hosting a WordPress Blog with Amazon Linux \(p. 51\)](#)
- [Tutorial: Configure Apache Web Server on Amazon Linux 2 to Use SSL/TLS \(p. 59\)](#)
- [Tutorial: Increase the Availability of Your Application on Amazon EC2 \(p. 74\)](#)
- [Tutorial: Remotely Manage Your Amazon EC2 Instances \(p. 77\)](#)

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.

To set up a LAMP web server on Amazon Linux AMI, see [Tutorial: Install a LAMP Web Server with the Amazon Linux AMI \(p. 41\)](#).

Important

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.

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 \(p. 27\)](#). 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 [Setting Up with Amazon EC2 \(p. 19\)](#).

To prepare the LAMP server

1. [Connect to your instance \(p. 28\)](#).
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
```

Note

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 with 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 \(p. 41\)](#).

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
```

Note

You can view the current versions of these packages with 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. Under **Security groups**, choose **view inbound rules**.
 - d. You should see the following list of rules in your default security group:

Ports	Protocol	Source	launch-wizard- <i>N</i>
Security Groups associated with i-1234567890abcdef0			

22	tcp	0.0.0.0/0	#
----	-----	-----------	---

Using the procedures in [Adding Rules to a Security Group \(p. 537\)](#), 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 \(p. 537\)](#).

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 Apache Web Server on Amazon Linux to use SSL/TLS](#).

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 \(p. 35\)](#).

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-domxml.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

Note

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 `lamp-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.

Note

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 Apache Web Server on Amazon Linux to use SSL/TLS](#).

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:

The screenshot shows the phpMyAdmin login interface. At the top, there's a decorative sailboat graphic and the text "phpMyAdmin". Below that is the heading "Welcome to phpMyAdmin". Underneath, there's a "Language" dropdown menu set to "English". The main login area has "Log in" and a gear icon at the top left. It contains two input fields: "Username" with "root" typed in and "Password" with a masked password. At the bottom right is a "Go" button.

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 \(p. 32\)](#).

- **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 \(p. 537\)](#).

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 \(p. 406\)](#)
- [Transferring Files to Linux Instances from Linux Using SCP \(p. 399\)](#)
- [Tutorial: Hosting a WordPress Blog with Amazon Linux \(p. 51\)](#)

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.

To set up a LAMP web server on Amazon Linux 2, see [Tutorial: Install a LAMP Web Server on Amazon Linux 2 \(p. 32\)](#).

Important

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.

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 \(p. 27\)](#). 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 [Setting Up with Amazon EC2 \(p. 19\)](#).

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

1. [Connect to your instance \(p. 28\)](#).
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.

Note

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 \(p. 50\)](#)

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 php70 mysql56-server php70-mysqlnd
```

Note

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
```

To set up a LAMP web server on Amazon Linux 2, see [Tutorial: Install a LAMP Web Server on Amazon Linux 2 \(p. 32\)](#).

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.
 - Under **Security groups**, choose **view inbound rules**.
 - You should see the following list of rules in your default security group:

```
Security Groups associated with i-1234567890abcdef0
Ports      Protocol     Source      launch-wizard-N
22         tcp          0.0.0.0/0  #
```

Using the procedures in [Adding Rules to a Security Group \(p. 537\)](#), 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. 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 \(p. 537\)](#).

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.

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!



Note

This test page appears only when there is no content in `/var/www/html`. When you add content to the document root, your content appears at the public DNS address of your instance instead of this test page.

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 Apache Web Server on Amazon Linux to use SSL/TLS](#).

To test your LAMP web 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.

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 \(p. 44\)](#).

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 5.6.6

System	Linux ip-172-31-7-35 3.14.35-28.38.amzn1.x86_64 #1 SMP Wed Mar 11 22:50:37 UTC 2015 x86_64
Build Date	Mar 5 2015 23:26:53
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-5.6.d
Additional .ini files parsed	/etc/php-5.6.d/20-bz2.ini, /etc/php-5.6.d/20-calendar.ini, /etc/php-5.6.d/20-ctype.ini, /etc/php-5.6.d/20-cs ini, /etc/php-5.6.d/20-dom.ini, /etc/php-5.6.d/20-exif.ini, /etc/php-5.6.d/20-fileinfo.ini, /etc/php-5.6.d/20-ftp.ini, /etc/php-5.6.d/20-gettext.ini, /etc/php-5.6.d/20-iconv.ini, /etc/php-5.6.d/20-mysqlind.ini, /etc/php-5.6.d/20-pdo.ini, /etc/php-5.6.d/20-phar.ini, /etc/php-5.6.d/20-posix.ini, /etc/php-5.6.d/20-shmop.ini, /etc/php-5.6.d/20-simplexml.ini, /etc/php-5.6.d/20-sockets.ini, /etc/php-5.6.d/20-sqlite3.ini, /etc/php-5.6.d/20-sysvmsg.ini, /etc/php-5.6.d/20-sysvshm.ini, /etc/php-5.6.d/20-tokenizer.ini, /etc/php-5.6.d/20-xml.ini, /etc/php-5.6.d/20-xmlreader.ini, /etc/php-5.6.d/20-xsl.ini, /etc/php-5.6.d/20-zip.ini, /etc/php-5.6.d/30-mysql.ini, /etc/php-5.6.d/30-mysqli.ini, /etc/php-5.6.d/30-pdo_mysql.ini, /etc/php-5.6.d/30-pdo_sqlite.ini, /etc/php-5.6.d/30-wddx.ini, /etc/php-5.6.d/30-xmlreader.ini, /etc/php-5.6.d/40-json.ini, /etc/php-5.6.d/php.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,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 php70 mysql56-server php70-mysqlind
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-mysqlind.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
```

To 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.

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.

Note

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.

(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 Apache Web Server on Amazon Linux to use SSL/TLS](#).

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 php70-mbstring.x86_64 php70-zip.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 [To install and start the LAMP web server with the Amazon Linux AMI \(p. 41\)](#).

- **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 \(p. 537\)](#).

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 php70 mysql56-server php70-mysqlnd perl-DBD-MySQL56
```

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-MYSQL55
```

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 \(p. 406\)](#)
- [Transferring Files to Linux Instances from Linux Using SCP \(p. 399\)](#)
- [Tutorial: Hosting a WordPress Blog with Amazon Linux \(p. 51\)](#)

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 [Getting 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.

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 \(p. 41\)](#) for Amazon Linux AMI or [Tutorial: Install a LAMP Web Server on Amazon Linux 2 \(p. 32\)](#) 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 \(p. 537\)](#).

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 \(p. 674\)](#).

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 AMI:** `sudo service mysqld start`
 - **Amazon Linux 2:** `sudo systemctl start mariadb`
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.

Important

If you have not secured your database server yet, it is very important that you do so. For more information, see [To secure the database server \(p. 46\)](#).

```
[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 \(p. 53\) of To create a database user and database for your WordPress installation \(p. 52\)](#).

```
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 \(p. 53\) of To create a database user and database for your WordPress installation \(p. 52\)](#).

```
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 \(p. 53\) of To create a database user and database for your WordPress installation \(p. 52\)](#).

```
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 http://codex.wordpress.org/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', 'zsz._P=l/|y.Lq)Xjlkws1y5NJ76E6EJ.AV0pCKZZB,*-*r ?6OP  
$eJ@;+(ndLg');  
define('LOGGED_IN_KEY', 'ju}qwre3V*+8f_zOWF?{LlGsQ]Ye@2Jh^,8x>)Y |;(^[Iw]Pi  
+LG#A4R?7N`YB3');  
define('NONCE_KEY', 'P(g62HeZxEes/LnI^i=H,[XwK9I&[2s|:?ON}VJM%?;v2v)v+;  
+^9eXUahg@:Cj');  
define('AUTH_SALT', 'C$DpB4Hj[JK:{ql`sRVA:{:7yShy(9A@5wg+`JJVb1fk%_-  
Bx*M4(qc[Og%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?Zy4zUU4r99QQ_xGs2LTd%P; |  
_eits)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

1. 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`).
2. 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/
```

3. 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 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 \(p. 41\)](#)).

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
```

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

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

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

- **Amazon Linux AMI:** `sudo service httpd restart`
- **Amazon Linux 2:** `sudo systemctl restart httpd`

To run the WordPress installation script

1. Use the `chkconfig` command to ensure that the `httpd` and database services start at every system boot.
 - **Amazon Linux AMI:** `sudo chkconfig enable httpd && sudo chkconfig enable mysql`
 - **Amazon Linux 2:** `sudo systemctl enable httpd && sudo systemctl enable mariadb`
2. Verify that the database server is running.
 - **Amazon Linux AMI:** `sudo service mysqld status`
 - **Amazon Linux 2:** `sudo systemctl status mariadb`

If the database service is not running, start it.

- **Amazon Linux AMI:** `sudo service mysqld start`
 - **Amazon Linux 2:** `sudo systemctl start mariadb`
3. Verify that your Apache web server (`httpd`) is running.
- **Amazon Linux AMI:** `sudo service httpd status`
 - **Amazon Linux 2:** `sudo systemctl status httpd`

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

- **Amazon Linux AMI:** `sudo service httpd start`
 - **Amazon Linux 2:** `sudo systemctl start httpd`
4. In a web browser, enter 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 screen.

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

The screenshot shows the initial WordPress installation page. At the top, the classic 'W' logo is displayed next to the word 'WORDPRESS'. Below it, the word 'Welcome' is centered. A paragraph of text follows, stating: 'Welcome to the famous five minute WordPress installation process! You may want to browse the [ReadMe](#) documentation at your leisure. Otherwise, just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.' Underneath this, a section titled 'Information needed' is shown. It contains a note: 'Please provide the following information. Don't worry, you can always change these settings later.' Two input fields are visible: one for 'Site Title' containing 'Amazon EC2 WordPress Blog', and another for 'Username' which is currently empty. A small note below the empty username field says: 'Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods and the @ symbol.'

5. Enter the remaining installation information into the WordPress installation wizard.

Field	Value
Site Title	Enter a name for your WordPress site.
Username	Enter a name for your WordPress administrator. For security purposes, you should choose a unique name for this user, because it will be more difficult to exploit than the default user name, <code>admin</code> .

Field	Value
Password	Enter a strong password, and then enter it again to confirm. Do not reuse an existing password, and make sure to store this password in a safe place.
Your E-mail	Enter the email address you want to use for notifications.

6. Click **Install WordPress** to complete the installation.

Congratulations, you should now be able to log into your WordPress blog and start posting entries.

Next Steps

After you have tested your initial 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 http://codex.wordpress.org/Changing_The_Site_URL.

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 \(p. 82\)](#).

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 [Modifying the Size, IOPS, or Type of an EBS Volume on Linux \(p. 769\)](#).
- 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 \(p. 228\)](#).
- Add additional instances. For more information, see [Tutorial: Increase the Availability of Your Application on Amazon EC2 \(p. 74\)](#).

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 http://codex.wordpress.org/Installing_WordPress#Common_Installation_Problems. For information about making your WordPress blog more secure, go to http://codex.wordpress.org/Hardening_WordPress. For information about keeping your WordPress blog up-to-date, go to http://codex.wordpress.org/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 http://codex.wordpress.org/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 http://codex.wordpress.org/Changing_The_Site_URL and http://codex.wordpress.org/Installing_WordPress#Common_Installation_Problems for more information.

Tutorial: Configure Apache Web Server on Amazon Linux 2 to Use SSL/TLS

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 a single instance of Amazon Linux 2 running Apache web

server. If you plan to offer commercial-grade services, the [AWS Certificate Manager](#), not discussed here, is a good option.

Note

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 all versions of SSL by default and recommends disabling TLS version 1.0, as described below. Only TLS 1.1 and 1.2 may be safely enabled. For more information about the updated encryption standard, see [RFC 7568](#).

Important

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

The version of this tutorial supporting Amazon Linux 1 is no longer maintained, but you can still find it on the [Internet Archive](#).

Topics

- [Prerequisites \(p. 60\)](#)
- [Step 1: Enable SSL/TLS on the Server \(p. 60\)](#)
- [Step 2: Obtain a CA-signed Certificate \(p. 62\)](#)
- [Step 3: Test and Harden the Security Configuration \(p. 67\)](#)
- [Troubleshooting \(p. 69\)](#)
- [Appendix: Let's Encrypt with Certbot on Amazon Linux 2 \(p. 70\)](#)

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 \(p. 27\)](#).
- 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 \(p. 628\)](#).

- Install an Apache web server. For step-by-step instructions, see [Tutorial: Install a LAMP Web Server on Amazon Linux 2 \(p. 32\)](#). 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 SSL/TLS public key infrastructure (PKI) relies on the Domain Name System (DNS). If you plan 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 may use [Amazon Route 53](#).

Step 1: Enable SSL/TLS on the Server

This procedure takes you through the process of setting up SSL/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 SSL/TLS on a server

1. [Connect to your instance \(p. 28\)](#) 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
```

3. Now that your instance is current, add SSL/TLS support by installing the Apache module `mod_ssl`:

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

Later in this tutorial, you work with three important files that have been installed:

- `/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 SSL/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 SSL/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-----  
MIIEazCCA1OgAwIBAgICWxQwDQYJKoZIhvcNAQELBQAwgbExCzAJBgNVBAYTAi0t  
MRIwEAYDVQQIDA1Tb21lU3RhGUxETAPBgNVBAcMCFNvbWVDaXR5MRkwFwYDVQQK  
DBBTb21lT3JnYW5pemF0aW9uMR8wHQYDVQQLDbzTb21lT3JnYW5pemF0aW9uYWxV  
bmlOMRkwFwYDVQoQDDBpcC0xNzItMzEtMjAtMjM2MSQwIgYJKoZIhvcNAQkBFhVv  
...  
-----END CERTIFICATE-----
```

```
-----  
z5rRUE/XzxRLBZoowZpNWTXJkQ3uFYH6s/  
sBwtHpKKZMzOvDedREjNKAvk4ws6F0  
WanXWehT6FiSzvB4sTEXXJN2jdw8g  
+sHGnZ8zC0sclknYhHrCVD2vnBlZJKSzvak  
3ZazhBxtQSukFMOonWPP2a0DMMFGYUHod0BQE8sBJxg==  
-----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 SSL/TLS files with your own customized files, be sure that they are in PEM format.

4. [Reboot your instance](#) and reconnect to it.
5. Restart Apache.

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

Note

Make sure the TCP port 443 is accessible on your EC2 instance, as described above.

6. 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 SSL/TLS on your server. All data passing between the browser and server is now encrypted.

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

Step 2: Obtain a CA-signed Certificate

This section describes the process of generating a certificate signing request (CSR) from a private key, submitting the CSR to a certificate authority (CA), obtaining a signed host certificate, and configuring Apache to use it.

A self-signed SSL/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 [dmoztols.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 the automation of the certificate creation and renewal process. For more information about using Let's Encrypt as your CA, see [Appendix: Let's Encrypt with Certbot on Amazon Linux 2 \(p. 70\)](#).

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 2 is 2048 bits, which means that the existing autogenerated 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.

To obtain a CA-signed certificate

1. [Connect to your instance \(p. 28\)](#) and navigate to `/etc/pki/tls/private/`. This is the directory where the server's private key for SSL/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 sample key configurations. Any of the resulting keys work with your web server, but they vary in the degree and type of security that they implement.
 1. As a starting point, here is the command to create an RSA key resembling the default host key on your instance:

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key 2048
```

The resulting file, `custom.key`, is a 2048-bit RSA private key.

2. To create a stronger RSA key with a bigger modulus, use the following command:

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

The resulting file, `custom.key`, is a 4096-bit RSA private key.

3. To create a 4096-bit encrypted RSA key with password protection, use the following command:

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

This results in a 4096-bit RSA private key that has been 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 need to supply the password "abcde12345" over an SSH connection.

4. RSA cryptography can be relatively slow, because its security relies on the difficulty of factoring the product of two large prime numbers. However, it is possible to create keys for SSL/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. Here is an example:

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

The output in this case 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 information shown in the table below. 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 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 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 will receive a new host certificate signed by the CA. You may 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
```

Examine the output for the tell-tale lines described above. Do not use files ending with **.p7b**, **.p7c**, or similar extensions.

5. Remove or rename the old self-signed host certificate **localhost.crt** from the **/etc/pki/tls/certs** directory and place the new CA-signed certificate there (along with any intermediate certificates).

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 (**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/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 commands would be as follows:

```
[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 2 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. Because the file name of the new CA-signed host certificate (**custom.crt** in this example) probably differs from the old certificate, edit /etc/httpd/conf.d/ssl.conf and provide the correct path and file name using Apache's SSLCertificateFile directive:

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

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.

If you installed a custom private key (**custom.key** in this example), provide its path and file name using Apache's SSLCertificateKeyFile directive:

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

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

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

Step 3: Test and Harden the Security Configuration

After your SSL/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. The table below summarizes the report for a domain with settings identical to the default Apache configuration on Amazon Linux 2 and a default Certbot certificate:

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

The report shows that the configuration is mostly sound, with acceptable ratings for certificate, protocol support, key exchange, and cipher strength issues. The configuration also supports [Forward secrecy](#), a feature of protocols 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. However, the report also flags one serious vulnerability that is responsible for lowering the overall grade, and points to an additional potential problem:

- RC4 cipher support:** A cipher is the mathematical core of an encryption algorithm. RC4, a fast cipher used to encrypt SSL/TLS data-streams, is known to have several [serious weaknesses](#). The fix is to completely disable RC4 support. We also specify an explicit cipher order and an explicit list of forbidden ciphers.

In the configuration file `/etc/httpd/conf.d/ssl.conf`, 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:

- a. Support for forward secrecy
- b. Strength
- c. Speed
- d. Specific ciphers before cipher families
- e. Allowed ciphers before denied ciphers

The high-ranking ciphers have *ECDHE* in their names, for *Elliptic Curve Diffie-Hellman Ephemeral*. The term *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 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*.

Finally, uncomment the following line by removing the "#":

```
#SSLHonorCipherOrder 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 strong secure connection before falling back to allowed ciphers with lesser security.

2. **Future protocol support:** The configuration supports TLS versions 1.0 and 1.1, which are on a path to deprecation, with TLS version 1.2 recommended after June 2018. To future-proof the protocol support, 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
```

Then, 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 supported versions of TLS. 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.

Restart Apache after saving these changes to the edited configuration file.

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

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

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 strip the key of its encryption and password. Assuming you have a private encrypted RSA key called `custom.key` in the default directory, and that the passphrase on it is `abcde12345`, run the following commands on your EC2 instance to generate an unencrypted version of this 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 like these:

```
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.

Appendix: 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 webserver. 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 below in [To automate Certbot \(p. 73\)](#). For more information, consult the Certbot [User Guide](#) and [man page](#).



Certbot is a client utility for EFF's Let's Encrypt certificate service.

Certbot is not officially supported on Amazon Linux 2, but is available for download and functions correctly once installed. We recommend that you make the following backups to protect your data and avoid inconvenience:

- Before you begin, take a snapshot of your EBS root volume. This allows you to restore the original state of your EC2 instance. For information about creating EBS snapshots, see [Creating an Amazon EBS Snapshot \(p. 795\)](#).
- 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. Download EPEL with the following command:

```
[ec2-user ~]$ sudo wget -r --no-parent -A 'epel-release*.rpm' http://dl.fedoraproject.org/pub/epel/7/x86_64/Packages/e/
```

- b. Install the repository packages as follows:

```
[ec2-user ~]$ sudo rpm -Uvh dl.fedoraproject.org/pub/epel/7/x86_64/Packages/e/epel-release*.rpm
```

- c. Enable EPEL:

```
[ec2-user ~]$ sudo yum-config-manager --enable epel*
```

You can confirm that EPEL is enabled with the following command:

```
[ec2-user ~]$ sudo yum repolist all

...
!epel/x86_64
  x86_64
!epel-debuginfo/x86_64
  x86_64 - Debug
!epel-source/x86_64
  x86_64 - Source
!epel-testing/x86_64
  Testing - x86_64
!epel-testing-debuginfo/x86_64
  Testing - x86_64 - Debug
!epel-testing-source/x86_64
  Testing - x86_64 - Source
                                         Extra Packages for Enterprise Linux 7 -
                                         enabled: 12,184+105
                                         Extra Packages for Enterprise Linux 7 -
                                         enabled: 2,717
                                         Extra Packages for Enterprise Linux 7 -
                                         enabled: 0
                                         Extra Packages for Enterprise Linux 7 -
                                         enabled: 959+10
                                         Extra Packages for Enterprise Linux 7 -
                                         enabled: 142
                                         Extra Packages for Enterprise Linux 7 -
                                         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) to configure:

```
<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's 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. If you want to experiment with customized keys, you might start with [Using ECDSA certificates with Let's Encrypt](#).

1. Install Certbot packages and dependencies using the following command:

```
[ec2-user ~]$ sudo yum install -y certbot python2-certbot-apache
```

2. Run Certbot.
3. At the prompt "Enter email address (used for urgent renewal and security notices)," type a contact address and press Enter.
4. Agree to the Let's Encrypt Terms of Service at the prompt. Type "A" and press Enter to proceed:

```
Starting new HTTPS connection (1): acme-v01.api.letsencrypt.org
-----
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-v01.api.letsencrypt.org/directory
-----
(A)gree/(C)ancel: A
```

5. At the authorization for EFF to put you on their mailing list, type "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, type "1". If you want to accept only encrypted connections via HTTPS, type "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/example.com/fullchain.pem`
Your key file has been saved at:
`/etc/letsencrypt/live/example.com/privkey.pem`
Your cert will expire on **2018-05-28**. 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 \(p. 67\)](#).

Configure Automated Certificate Renewal

To automate Certbot

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.

1. Open `/etc/crontab` in a text editor and add a line similar to the following:

```
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.

Save the file when done.

2. Restart the cron daemon:

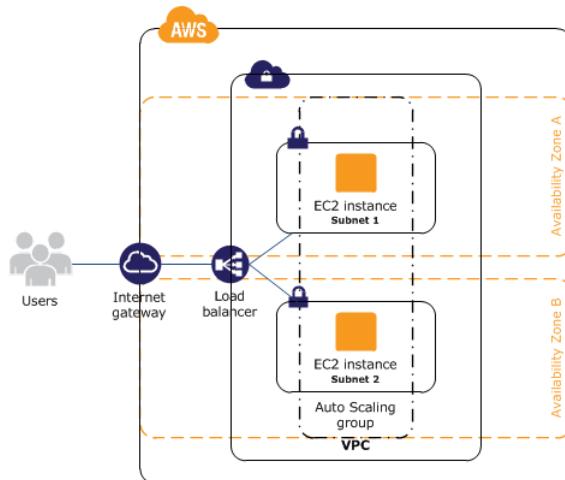
```
[ec2-user ~]$ sudo systemctl restart crond
```

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 \(p. 74\)](#)
- [Scale and Load Balance Your Application \(p. 75\)](#)
- [Test Your Load Balancer \(p. 76\)](#)

Prerequisites

This tutorial assumes that you have already done the following:

1. Created a virtual private cloud (VPC) with one public subnet in two or more Availability Zones. If you haven't done so, see [Create a Virtual Private Cloud \(VPC\) \(p. 23\)](#).

2. Launched an instance in the VPC.
3. Connected 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 \(p. 41\)](#).
4. Tested your application on your instance to ensure that your instance is configured correctly.
5. Created a custom Amazon Machine Image (AMI) from your instance. For more information, see [Creating an Amazon EBS-Backed Linux AMI \(p. 101\)](#) or [Creating an Instance Store-Backed Linux AMI \(p. 105\)](#).
6. (Optional) Terminated the instance if you no longer need it.
7. Created 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 \(p. 623\)](#).

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**, type 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**, type 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. On the **Choose AMI** page, select the **My AMIs** tab, and then select the AMI that you created in [Prerequisites \(p. 74\)](#).
13. On the **Choose Instance Type** page, select an instance type, and then choose **Next: Configure details**.
14. On the **Configure details** page, do the following:
 - a. For **Name**, type a name for your launch configuration (for example, `my-launch-config`).
 - b. For **IAM role**, select the IAM role that you created in [Prerequisites \(p. 74\)](#).
 - c. (Optional) If you need to run a startup script, expand **Advanced Details** and type the script in **User data**.
 - d. Choose **Skip to review**.
15. On the **Review** page, choose **Edit 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 your instances will have public IP addresses, you can optionally allow SSH traffic if you need to connect to the instances. When you are finished, choose **Review**.
16. On the **Review** page, choose **Create launch configuration**.
17. When prompted, select an existing key pair, create a new key pair, or proceed without a key pair. Select the acknowledgment check box, and then choose **Create launch configuration**.
18. 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, choose **Create an Auto Scaling group using this launch configuration**.
19. On the **Configure Auto Scaling group details** page, do the following:
 - a. For **Group name**, type a name for the Auto Scaling group. For example, `my-asg`.
 - b. 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.
 - c. Select your VPC from **Network** and your two public subnets from **Subnet**.
 - d. Under **Advanced Details**, select **Receive traffic from one or more load balancers**. Select your target group from **Target Groups**.
 - e. Choose **Next: Configure scaling policies**.
20. On the **Configure scaling policies** page, choose **Review**, as we will let Amazon EC2 Auto Scaling maintain the group at the specified size. Note that later on, you can manually scale this Auto Scaling group, configure the group to scale on a schedule, or configure the group to scale based on demand.
21. On the **Review** page, choose **Create Auto Scaling group**.
22. After the group is created, choose **Close**.

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 **Instances** tab. Initially, your instances are in the **Pending** state. When their states are **InService**, they are ready for use.
2. Verify that your instances are registered with the load balancer. From the **Target Groups** page, select your target group, and then choose the **Targets** tab. 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 test your load balancer as follows.
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.

Tutorial: Remotely Manage Your Amazon EC2 Instances

This tutorial shows you how to remotely manage an Amazon EC2 instance using Systems Manager Run Command from your local machine. This tutorial includes procedures for executing commands using the Amazon EC2 console, AWS Tools for Windows PowerShell, and the AWS Command Line Interface.

Note

With Run Command, you can also manage your servers and virtual machines (VMs) in your on-premises environment or in an environment provided by other cloud providers. For more information, see [Setting Up Systems Manager in Hybrid Environments](#).

Before You Begin

You must configure an AWS Identity and Access Management (IAM) instance profile role for Systems Manager. Attach an IAM role with the **AmazonEC2RoleforSSM** managed policy to an Amazon EC2 instance. This role enables the instance to communicate with the Systems Manager API. For more information about how to attach the role to an existing instance, see [Attaching an IAM Role to an Instance \(p. 625\)](#).

You must also configure your IAM user account for Systems Manager, as described in the next section.

Grant Your User Account Access to Systems Manager

Your user account must be configured to communicate with the SSM API. Use the following procedure to attach a managed AWS Identity and Access Management (IAM) policy to your user account that grants you full access to SSM API actions.

To create the IAM policy for your user account

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**. (If this is your first time using IAM, choose **Get Started**, and then choose **Create policy**.)
3. In the **Filter** field, type **AmazonSSMFullAccess** and press Enter.

4. Select the check box next to **AmazonSSMFullAccess** and then choose **Policy actions, Attach**.
5. On the **Attach Policy** page, choose your user account and then choose **Attach policy**.

Install the SSM Agent

SSM Agent processes Run Command requests and configures the instances that are specified in the request. The agent is installed by default on Windows AMIs starting in November 2016 and later, Amazon Linux AMIs starting with 2017.09, and all Amazon Linux 2 AMIs.

To install the agent on Linux, see [Installing and Configuring SSM Agent on Linux Instances](#) in the *AWS Systems Manager User Guide*.

To install the agent on Windows, see [Installing and Configuring SSM Agent on Windows Instances](#) in the *AWS Systems Manager User Guide*.

Send a Command Using the EC2 Console

Use the following procedure to list all services running on the instance by using Run Command from the Amazon EC2 console.

To execute a command using Run Command from the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Run Command**.
3. Choose **Run a command**.
4. For **Command document**, choose **AWS-RunPowerShellScript** for Windows instances, and **AWS-RunShellScript** for Linux instances.
5. For **Target instances**, choose the instance you created. If you don't see the instance, verify that you are currently in the same region as the instance you created. Also verify that you configured the IAM role and trust policies as described earlier.
6. For **Commands**, type **Get-Service** for Windows, or **ps aux** for Linux.
7. (Optional) For **Working Directory**, specify a path to the folder on your EC2 instances where you want to run the command.
8. (Optional) For **Execution Timeout**, specify the number of seconds the EC2Config service or SSM agent will attempt to run the command before it times out and fails.
9. For **Comment**, we recommend providing information that will help you identify this command in your list of commands.
10. For **Timeout (seconds)**, type the number of seconds that Run Command should attempt to reach an instance before it is considered unreachable and the command execution fails.
11. Choose **Run** to execute the command. Run Command displays a status screen. Choose **View result**.
12. To view the output, choose the command invocation for the command, choose the **Output** tab, and then choose **View Output**.

The screenshot shows the AWS Lambda console interface. At the top, there are two tabs: "Run a command" (selected) and "Actions". Below the tabs is a search bar labeled "Filter by attributes". A table lists several command entries:

	Command ID	Instance ID	Document name	Status	Requested date	Comments
<input checked="" type="checkbox"/>	65555b90-ee60-45...	i-8fd6aa30	AWS-RunPowerSh...	Success	October 21, 2015 at...	Listing
<input type="checkbox"/>	65555b90-ee60-45...	i-d583f76a	AWS-RunPowerSh...	Success	October 21, 2015 at...	Listing
<input type="checkbox"/>	65555b90-ee60-45...	i-8ed6aa31	AWS-RunPowerSh...	Success	October 21, 2015 at...	Listing
<input type="checkbox"/>	ca4b10c6-cee1-437...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	getting
<input type="checkbox"/>	561e5f4a-27d2-419...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	ipconfi

Below the table, the command and instance IDs are displayed: "Command ID: 65555b90-ee60-4520-9dc3-e42e94445469" and "Instance ID: i-8fd6aa30". Under the "Output" tab, the following parameters are shown:

Command ID	65555b90-ee60-4520-9dc3-e42e94445469	Instance...
Document name	AWS-RunPowerShellScript	Stat...
Date requested	October 21, 2015 at 3:56:59 PM UTC-7	Comm...
Output S3 bucket	run-command-test	Document parameters

For more examples of how to execute commands using Run Command, see [Executing Commands Using Systems Manager Run Command](#).

Send a Command Using AWS Tools for Windows PowerShell

Use the following procedure to list all services running on the instance by using Run Command from AWS Tools for Windows PowerShell.

To execute a command

1. On your local computer, download the latest version of [AWS Tools for Windows PowerShell](#).
2. Open **AWS Tools for Windows PowerShell** on your local computer and execute the following command to specify your credentials.

```
Set-AWSCredentials -AccessKey key -SecretKey key
```

3. Execute the following command to set the region for your PowerShell session. Specify the region where you created the instance in the previous procedure. This example uses the us-west-2 region.

```
Set-DefaultAWSRegion -Region us-west-2
```

4. Execute the following command to retrieve the services running on the instance.

```
Send-SSMCommand -InstanceId 'Instance-ID' -DocumentName AWS-RunPowerShellScript -Comment 'listing services on the instance' -Parameter @{'commands'=@('Get-Service')}
```

The command returns a command ID, which you will use to view the results.

5. The following command returns the output of the original Send-SSMCommand. The output is truncated after 2500 characters. To view the full list of services, specify an Amazon S3 bucket in the command using the -OutputS3BucketName *bucket_name* parameter.

```
Get-SSMCommandInvocation -CommandId Command-ID -Details $true | select -ExpandProperty  
CommandPlugins
```

For more examples of how to execute commands using Run Command with Tools for Windows PowerShell, see [Systems Manager Run Command Walkthrough Using the AWS Tools for Windows PowerShell](#).

Send a Command Using the AWS CLI

Use the following procedure to list all services running on the instance by using Run Command in the AWS CLI.

To execute a command

1. On your local computer, download the latest version of the [AWS Command Line Interface](#) (AWS CLI).
2. Open the AWS CLI on your local computer and execute the following command to specify your credentials and the region.

```
aws configure
```

3. The system prompts you to specify the following.

```
AWS Access Key ID [None]: key
AWS Secret Access Key [None]: key
Default region name [None]: region, for example us-east-1
Default output format [None]: ENTER
```

4. Execute the following command to retrieve the services running on the instance.

```
aws ssm send-command --document-name "AWS-RunShellScript" --comment "listing services"
--instance-ids "Instance-ID" --parameters commands="service --status-all" --region us-west-2 --output text
```

The command returns a command ID, which you will use to view the results.

5. The following command returns the output of the original Send-SSMCommand. The output is truncated after 2500 characters. To view the full list of services, you would need to specify an Amazon S3 bucket in the command using the --output-s3-bucket-name *bucket_name* parameter.

```
aws ssm list-command-invocations --command-id "command ID" --details
```

For more examples of how to execute commands using Run Command using the AWS CLI, see [Systems Manager Run Command Walkthrough Using the AWS CLI](#).

Related Content

For more information about Run Command and Systems Manager, see the following references.

- [AWS Systems Manager User Guide](#)
- [Amazon EC2 Systems Manager API Reference](#)

- [Systems Manager AWS Tools for PowerShell Cmdlet Reference](#)
- [Systems Manager AWS CLI Command Reference](#)
- [AWS SDKs](#)

Amazon Machine Images (AMI)

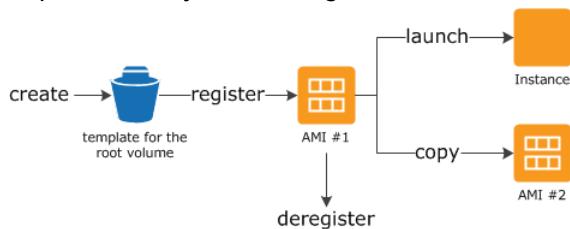
An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. You must specify a source AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration. You can use different AMIs to launch instances when you need instances with different configurations.

An AMI includes the following:

- A template for the root volume for 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

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 \(p. 83\)](#) and [Finding a Linux AMI \(p. 87\)](#).

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 \(p. 163\)](#).

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 information, see [Amazon EC2 Root Device Volume \(p. 13\)](#).

To create an Amazon EBS-backed AMI, see [Creating an Amazon EBS-Backed Linux AMI \(p. 101\)](#). To create an instance store-backed AMI, see [Creating an Instance Store-Backed Linux AMI \(p. 105\)](#).

To help categorize and manage your AMIs, you can assign custom *tags* to them. For more information, see [Tagging Your Amazon EC2 Resources \(p. 887\)](#).

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 \(p. 89\)](#).

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 \(p. 98\)](#).

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 \(p. 144\)](#).

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 \(p. 146\)](#).

AMI Types

You can select an AMI to use based on the following characteristics:

- Region (see [Regions and Availability Zones \(p. 6\)](#))
- Operating system
- Architecture (32-bit or 64-bit)
- [Launch Permissions \(p. 84\)](#)

- [Storage for the Root Device \(p. 84\)](#)

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 \(p. 89\)](#). Developers can charge for their AMIs. For more information, see [Paid AMIs \(p. 98\)](#).

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 \(p. 13\)](#).

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.

Characteristic	Amazon EBS-Backed AMI	Amazon Instance Store-Backed AMI
AMI creation/bundling	Uses a single command/call	Requires installation and use of AMI tools
Stopped state	Can be placed in stopped state where instance is not running, but 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 Device Volume to Persist \(p. 16\)](#).

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 \(p. 415\)](#).

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 \(p. 848\)](#).

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 \(p. 866\)](#).

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 \(p. 101\)](#).

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](#).

HVM AMIs

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.

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 \(p. 708\)](#) and [Linux Accelerated Computing Instances \(p. 208\)](#).

All instance types support HVM AMIs.

To find an HVM AMI, verify that the virtualization type of the AMI is set to `hvm`, using the console or the [describe-images](#) command.

PV AMIs

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 `menu.lst` file on your image. Paravirtual guests can run on host hardware that does not have explicit support for virtualization, but they cannot take advantage of special hardware extensions such as enhanced networking or GPU processing. 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 \(p. 157\)](#).

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.

The following AWS regions support PV instances: Asia Pacific (Tokyo), Asia Pacific (Singapore), Asia Pacific (Sydney), EU (Frankfurt), EU (Ireland), South America (São Paulo), US East (N. Virginia), US West (N. California), and US West (Oregon).

To find a PV AMI, verify that the virtualization type of the AMI is set to `paravirtual`, 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`) or 64-bit (`x86_64`)

- 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 \(p. 88\)](#)
- [Finding an AMI Using the AWS CLI \(p. 88\)](#)

Finding a Linux AMI Using the Amazon EC2 Console

You can find Linux AMIs using the Amazon EC2 console. You can search through all available AMIs using the **Images** page, or select from commonly used AMIs on the **Quick Start** tab when you use the console to launch an instance. AMI IDs are unique to each region.

To find a Linux AMI using the Choose AMI 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. From the console dashboard, choose **Launch Instance**.
4. On **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 **AWS Marketplace** or **Community AMIs** tab to find additional AMIs.

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 \(p. 84\)](#).
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 \(p. 357\)](#). 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 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"
```

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. If you have questions or observations about a shared AMI, use the [AWS forums](#).

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 User Guide](#) and [AWS Marketplace Seller Guide](#).

Contents

- [Finding Shared AMIs \(p. 89\)](#)
- [Making an AMI Public \(p. 91\)](#)
- [Sharing an AMI with Specific AWS Accounts \(p. 92\)](#)
- [Using Bookmarks \(p. 94\)](#)
- [Guidelines for Shared Linux AMIs \(p. 94\)](#)

Finding Shared AMIs

You can use the Amazon EC2 console or the command line to find shared AMIs.

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.config  
[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 \(p. 92\)](#)). You are not billed when your AMI is launched by other AWS accounts; only the accounts launching the AMI are billed.

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 \(p. 139\)](#).

To avoid exposing sensitive data when you share an AMI, read the security considerations in [Guidelines for Shared Linux AMIs \(p. 94\)](#) and follow the recommended actions.

Note

If an AMI has a product code, or contains a snapshot of an encrypted volume, you can't make it public. You must share the AMI with only specific AWS accounts.

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-12345678 --launch-permission "Add=[{Group=all}]"
```

2. To verify the launch permissions of the AMI, use the following `describe-image-attribute` command.

```
aws ec2 describe-image-attribute --image-id ami-12345678 --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-12345678 --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 are the AWS account IDs.

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 \(p. 139\)](#).

Note

If you are sharing an AMI containing a snapshot of an encrypted volume, see [Sharing an Amazon EBS Snapshot](#) for restrictions that apply.

Sharing 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.

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 \(p. 89\)](#).

Sharing 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-12345678 --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
```

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-12345678 --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-12345678 --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-12345678 AMI in the us-east-1 region:

```
https://console.aws.amazon.com/ec2/v2/home?region=us-  
east-1#LaunchInstanceWizard:ami=ami-12345678
```

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.

Note

No list of security guidelines can be exhaustive. Build your shared AMIs carefully and take time to consider where you might expose sensitive data.

Topics

- [Update the AMI Tools Before Using Them \(p. 95\)](#)
- [Disable Password-Based Remote Logins for Root \(p. 95\)](#)
- [Disable Local Root Access \(p. 95\)](#)
- [Remove SSH Host Key Pairs \(p. 96\)](#)
- [Install Public Key Credentials \(p. 96\)](#)
- [Disabling sshd DNS Checks \(Optional\) \(p. 97\)](#)
- [Identify Yourself \(p. 97\)](#)
- [Protect Yourself \(p. 97\)](#)

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 journalling 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.

```
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

The previous sections described how to make your shared AMIs safe, secure, and usable for the users who launch them. This section describes guidelines to protect yourself from the users of your AMI.

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 \(p. 122\)](#).

- 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 \(p. 99\)](#)
- [Finding a Paid AMI \(p. 99\)](#)
- [Purchasing a Paid AMI \(p. 99\)](#)
- [Getting the Product Code for Your Instance \(p. 100\)](#)
- [Using Paid Support \(p. 100\)](#)
- [Bills for Paid and Supported AMIs \(p. 101\)](#)
- [Managing Your AWS Marketplace Subscriptions \(p. 101\)](#)

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** from the first **Filter** list. In the Search bar choose **Product Code**, then **Marketplace**. In the Search bar again, choose **Platform** and then select the operating system from the list.

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"  
    }  
,
```

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 \(p. 372\)](#).

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 \(p. 372\)](#).

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 \(p. 454\)](#).

To retrieve a product code, use the following command:

```
[ec2-user ~]$ GET 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 \(p. 84\)](#). For more information about creating an instance store-backed Linux AMI, see [Creating an Instance Store-Backed Linux AMI \(p. 105\)](#).

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 \(p. 144\)](#). 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 \(p. 819\)](#).

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 an Amazon EBS Snapshot \(p. 795\)](#).

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.

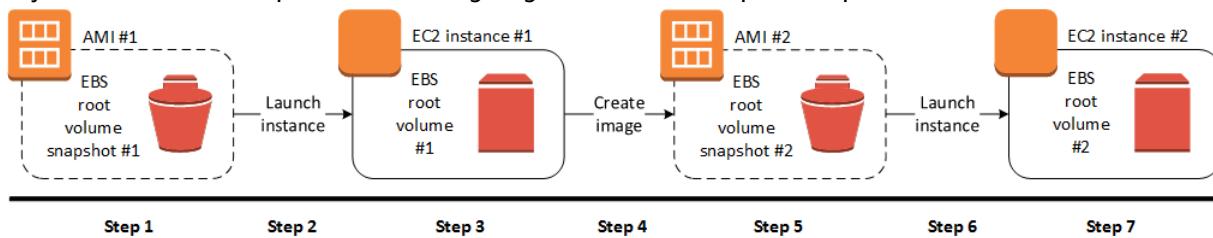
If you add instance-store volumes or EBS volumes to your instance in addition to the root device volume, the block device mapping for the new AMI contains information for these volumes, and the block device mappings for instances that you launch from the new AMI automatically contain information for 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 \(p. 866\)](#).

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 \(p. 355\)](#).
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 \(p. 355\)](#).
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 an Amazon EBS Snapshot \(p. 795\)](#).
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.

Note

Delete on Termination determines if the EBS volume is deleted or not; it does not affect the instance or the AMI.

- 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 \(p. 852\)](#). 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 \(p. 355\)](#).
 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.

Important

Some Linux distributions, such as Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES), use the Amazon EC2 `billingProduct` code associated with an AMI to verify subscription status for package updates. Creating an AMI from an EBS snapshot does not maintain this billing code, and instances launched from such an AMI are not able to connect to package update infrastructure. If you purchase a Reserved Instance offering for one of these Linux distributions and launch instances using an AMI that does not contain the required billing code, your Reserved Instance is not applied to these instances.

Similarly, although you can create a Windows AMI from a snapshot, you can't successfully launch an instance from the AMI.

In general, AWS advises against manually creating AMIs from snapshots.

For more information about creating Windows AMIs or AMIs for Linux operating systems that must retain AMI billing codes to work properly, see [Creating a Linux AMI from an Instance \(p. 102\)](#).

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 \(p. 864\)](#).
 - **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 \(p. 86\)](#).
 - (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 the Volume \(p. 787\)](#).

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)

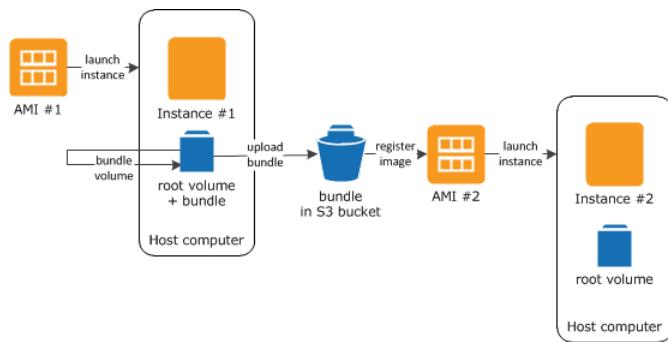
Creating an Instance Store-Backed Linux AMI

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.

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 \(p. 84\)](#). If you need to create an Amazon EBS-backed Linux AMI, see [Creating an Amazon EBS-Backed Linux AMI \(p. 101\)](#).

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 \(p. 144\)](#).

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 \(p. 866\)](#).

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 \(p. 106\)](#).
- 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 \(p. 108\)](#). The X.509 certificate and private key are used to encrypt and decrypt your AMI.
 - [China (Beijing)] Use the `$EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2-cn-north-1.pem` certificate.
 - [AWS GovCloud (US)] Use the `$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 \(p. 106\)](#)
- [Creating an AMI from an Instance Store-Backed Amazon Linux Instance \(p. 109\)](#)
- [Creating an AMI from an Instance Store-Backed Ubuntu Instance \(p. 112\)](#)
- [Converting your Instance Store-Backed AMI to an Amazon EBS-Backed AMI \(p. 116\)](#)

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 \(p. 119\)](#) 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 \(p. 119\)](#) 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 \(p. 119\)](#) 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 \(p. 106\)](#).

Topics

- [Creating an AMI from an Instance Store-Backed Amazon Linux Instance \(p. 109\)](#)
- [Creating an AMI from an Instance Store-Backed Ubuntu Instance \(p. 112\)](#)

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 \(p. 112\)](#).

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 \(p. 106\)](#).

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 \(p. 399\)](#). 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 \(p. 122\)](#) 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 \(p. 747\)](#), [attach \(p. 750\)](#), and [mount \(p. 751\)](#) 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 \(p. 122\)](#) 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) regions, use the `--ec2cert` parameter and specify the certificates as per the [prerequisites \(p. 106\)](#).

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 \(p. 866\)](#).

- 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.

```
<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 \(p. 133\)](#) 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 \(p. 110\)](#), 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 \(p. 133\)](#) 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. 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 \(p. 109\)](#).

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 GRUB Legacy must be installed. Proceed to [Step 1.b \(p. 112\)](#). If GRUB Legacy is already present, you can skip to [Step 2 \(p. 112\)](#).

- b. Install the `grub` package using the following command.

```
ubuntu:~$ sudo apt-get install -y grub
```

Verify that your instance is using GRUB Legacy.

```
ubuntu:~$ grub --version
grub (GNU GRUB 0.97)
```

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 \(p. 106\)](#).

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 \(p. 399\)](#). 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 \(p. 122\)](#) 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 \(p. 747\)](#), [attach \(p. 750\)](#), and [mount \(p. 751\)](#) 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. 122) 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. 866).

- 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/sdal</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. 133) 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. 114), 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. 133) 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 \(p. 355\)](#). 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. 399). 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. 128) 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. 132) command.

- Change directories to the bundle folder.

```
[ec2-user ~]$ cd /tmp/bundle/
```

- Run the [ec2-unbundle](#) (p. 132) 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 \(p. 106\)](#).

For information about your access keys, see [Best Practices for Managing AWS Access Keys](#).

Commands

- [ec2-ami-tools-version \(p. 119\)](#)
- [ec2-bundle-image \(p. 120\)](#)
- [ec2-bundle-vol \(p. 122\)](#)
- [ec2-delete-bundle \(p. 126\)](#)
- [ec2-download-bundle \(p. 128\)](#)
- [ec2-migrate-manifest \(p. 130\)](#)
- [ec2-unbundle \(p. 132\)](#)
- [ec2-upload-bundle \(p. 133\)](#)
- [Common Options for AMI Tools \(p. 135\)](#)

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 \(p. 106\)](#), 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`.

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 \(p. 105\)](#).

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 \(p. 106\)](#), 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

--sigvversion

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 myawsbucket -a your_access_key_id -s your_secret_access_key
Deleting files:
myawsbucket/image.manifest.xml
myawsbucket/image.part.00
myawsbucket/image.part.01
myawsbucket/image.part.02
myawsbucket/image.part.03
myawsbucket/image.part.04
myawsbucket/image.part.05
```

```
myawsbucket/image.part.06
Continue? [y/n]
y
Deleted myawsbucket/image.manifest.xml
Deleted myawsbucket/image.part.00
Deleted myawsbucket/image.part.01
Deleted myawsbucket/image.part.02
Deleted myawsbucket/image.part.03
Deleted myawsbucket/image.part.04
Deleted myawsbucket/image.part.05
Deleted myawsbucket/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 `myawsbucket` Amazon S3 bucket.

```
[ec2-user ~]$ mkdir bundled
[ec2-user ~]$ ec2-download-bundle -b myawsbucket/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 myawsbucket to mybundle/image.manifest.xml ...
Downloading part image.part.00 from myawsbucket/bundles/bundle_name to mybundle/
image.part.00 ...
Downloaded image.part.00 from myawsbucket
Downloading part image.part.01 from myawsbucket/bundles/bundle_name to mybundle/
image.part.01 ...
```

```
Downloaded image.part.01 from myawsbucket
Downloading part image.part.02 from myawsbucket/bundles/bundle_name to mybundle/
image.part.02 ...
Downloaded image.part.02 from myawsbucket
Downloading part image.part.03 from myawsbucket/bundles/bundle_name to mybundle/
image.part.03 ...
Downloaded image.part.03 from myawsbucket
Downloading part image.part.04 from myawsbucket/bundles/bundle_name to mybundle/
image.part.04 ...
Downloaded image.part.04 from myawsbucket
Downloading part image.part.05 from myawsbucket/bundles/bundle_name to mybundle/
image.part.05 ...
Downloaded image.part.05 from myawsbucket
Downloading part image.part.06 from myawsbucket/bundles/bundle_name to mybundle/
image.part.06 ...
Downloaded image.part.06 from myawsbucket
```

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 \(p. 106\)](#), 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 \(p. 157\)](#).

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 \(p. 157\)](#).

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-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem --privatekey pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.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 \(p. 105\)](#).

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 myawsbucket/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 myawsbucket ...
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.

AMIs with Encrypted Snapshots

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.

EC2 instances with encrypted volumes are launched from AMIs in the same way as other instances.

The `CopyImage` action can be used to create an AMI with encrypted snapshots from an AMI with unencrypted snapshots. By default, `CopyImage` preserves the encryption status of source snapshots when creating destination copies. However, you can configure the parameters of the copy process to also encrypt the destination snapshots.

Snapshots can be encrypted with either your default AWS Key Management Service customer master key (CMK), or with a custom key that you specify. You must in all cases have permission to use the selected key. If you have an AMI with encrypted snapshots, you can choose to re-encrypt them with a different encryption key as part of the `CopyImage` action. `CopyImage` accepts only one key at a time and encrypts all of an image's snapshots (whether root or data) to that key. However, it is possible to manually build an AMI with snapshots encrypted to multiple keys.

Support for creating AMIs with encrypted snapshots is accessible through the Amazon EC2 console, Amazon EC2 API, or the AWS CLI.

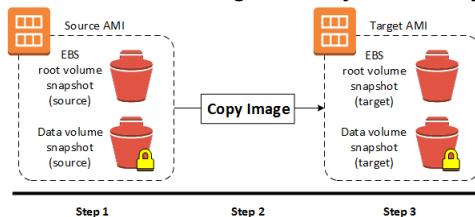
The encryption parameters of `CopyImage` are available in all regions where AWS KMS is available.

AMI Scenarios Involving Encrypted EBS Snapshots

You can copy an AMI and simultaneously encrypt its associated EBS snapshots using the AWS Management Console or the command line.

Copying an AMI with an Encrypted Data Snapshot

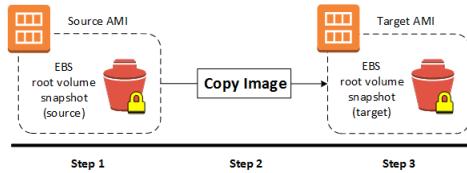
In this scenario, an EBS-backed AMI has an unencrypted root snapshot and an encrypted data snapshot, shown in step 1. The `CopyImage` action is invoked in step 2 without encryption parameters. As a result, the encryption status of each snapshot is preserved, so that the destination AMI, in step 3, is also backed by an unencrypted root snapshot and an encrypted data snapshot. Though the snapshots contain the same data, they are distinct from each other and you will incur storage costs for the snapshots in both AMIs, as well as charges for any instances you launch from either AMI.



You can perform a simple copy such as this using either the Amazon EC2 console or the command line. For more information, see [Copying an AMI \(p. 139\)](#).

Copying an AMI Backed by An Encrypted Root Snapshot

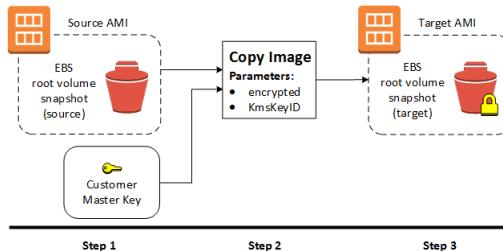
In this scenario, an Amazon EBS-backed AMI has an encrypted root snapshot, shown in step 1. The `CopyImage` action is invoked in step 2 without encryption parameters. As a result, the encryption status of the snapshot is preserved, so that the destination AMI, in step 3, is also backed by an encrypted root snapshot. Though the root snapshots contain identical system data, they are distinct from each other and you will incur storage costs for the snapshots in both AMIs, as well as charges for any instances you launch from either AMI.



You can perform a simple copy such as this using either the Amazon EC2 console or the command line. For more information, see [Copying an AMI \(p. 139\)](#).

Creating an AMI with Encrypted Root Snapshot from an Unencrypted AMI

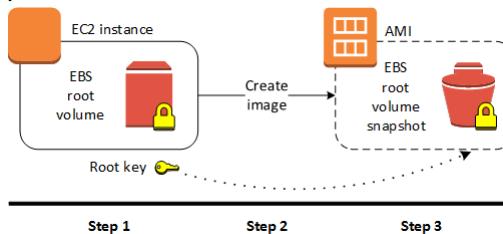
In this scenario, an Amazon EBS-backed AMI has an unencrypted root snapshot, shown in step 1, and an AMI is created with an encrypted root snapshot, shown in step 3. The `CopyImage` action in step 2 is invoked with two encryption parameters, including the choice of a CMK. As a result, the encryption 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 will incur storage costs for the snapshots in both AMIs, as well as charges for any instances you launch from either AMI.



You can perform a copy and encrypt operation such as this using either the Amazon EC2 console or the command line. For more information, see [Copying an AMI \(p. 139\)](#).

Creating an AMI with an Encrypted Root Snapshot from a Running Instance

In this scenario, an AMI is created from a running EC2 instance. The running instance in step 1 has an encrypted root volume, and the created AMI in step 3 has a root snapshot encrypted to the same key as the source volume. The `CreateImage` action has exactly the same behavior whether or not encryption is present.



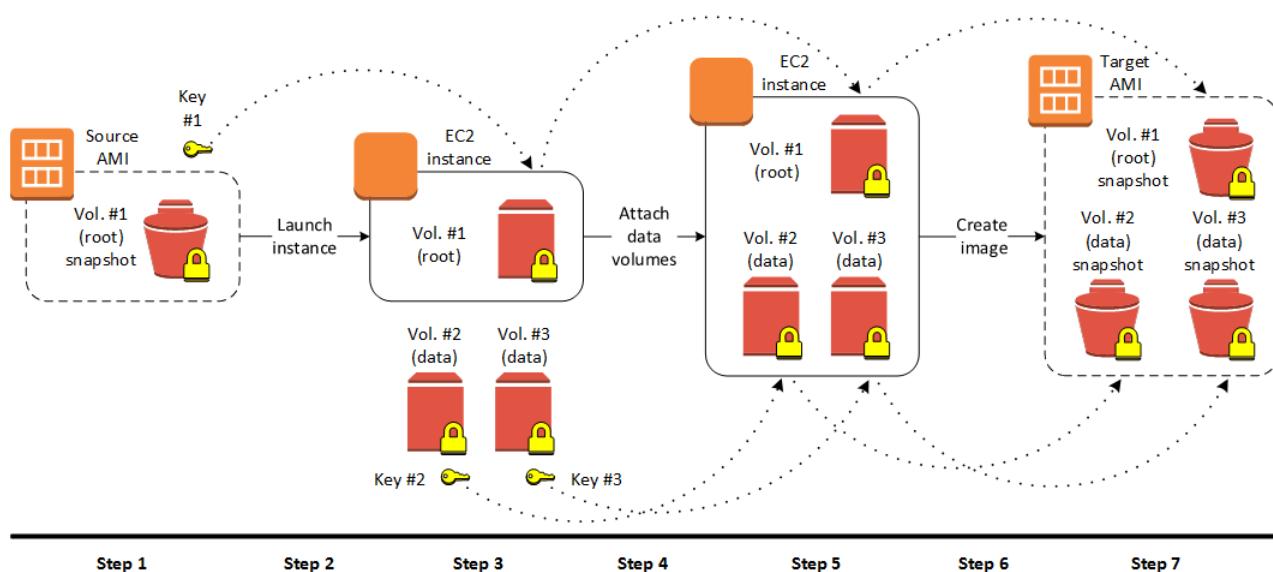
You can create an AMI from a running Amazon EC2 instance (with or without encrypted volumes) using either the Amazon EC2 console or the command line. For more information, see [Creating an Amazon EBS-Backed Linux AMI \(p. 101\)](#).

Creating an AMI with Unique CMKs for Each Encrypted Snapshot

This scenario starts with an AMI backed by a root-volume snapshot (encrypted to key #1), and finishes with an AMI that has two additional data-volume snapshots attached (encrypted to key #2 and key #3). The `CopyImage` action cannot apply more than one encryption key in a single operation. However, you can create an AMI from an instance that has multiple attached volumes encrypted to different keys. The resulting AMI has snapshots encrypted to those keys and any instance launched from this new AMI also has volumes encrypted to those keys.

The steps of this example procedure correspond to the following diagram.

1. Start with the source AMI backed by vol. #1 (root) snapshot, which is encrypted with key #1.
2. Launch an EC2 instance from the source AMI.
3. Create EBS volumes vol. #2 (data) and vol. #3 (data), encrypted to key #2 and key #3 respectively.
4. Attach the encrypted data volumes to the EC2 instance.
5. The EC2 instance now has an encrypted root volume as well as two encrypted data volumes, all using different keys.
6. Use the `CreateImage` action on the EC2 instance.
7. The resulting target AMI contains encrypted snapshots of the three EBS volumes, all using different keys.



You can carry out this procedure using either the Amazon EC2 console or the command line. For more information, see the following topics:

- [Launch Your Instance \(p. 354\)](#)
- [Creating an Amazon EBS-Backed Linux AMI \(p. 101\)](#).
- [Amazon EBS Volumes \(p. 733\)](#)
- [AWS Key Management in the AWS Key Management Service Developer Guide](#)

Copying an AMI

You can copy an Amazon Machine Image (AMI) within or across an AWS region using the AWS Management Console, the AWS 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 encrypted AMIs and AMIs with encrypted snapshots.

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 one exception is when you choose to 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.

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.

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*"  
            ]  
        }  
    ]  
}
```

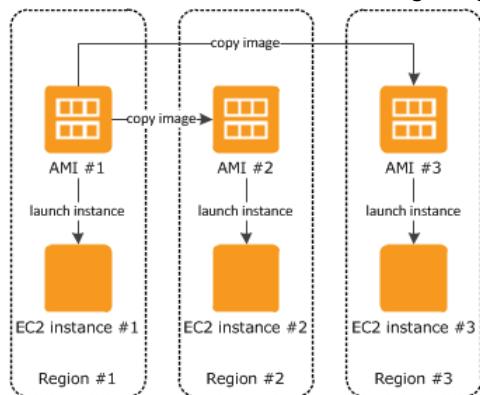
To find the Amazon Resource Name (ARN) of the AMI source bucket, open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>, in the navigation pane choose **AMIs**, and locate the bucket name in the **Source** column.

Cross-Region AMI Copy

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 at a time, with no more than 25 of those coming from a single source region.
- Not all regions support paravirtual (PV) AMIs; therefore, you cannot copy a PV AMI to those regions. For more information, see [Linux AMI Virtualization Types \(p. 86\)](#).

Cross-Account AMI Copy

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 \(p. 92\)](#).

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).

Limits

- You can't copy an encrypted AMI that was shared with you from another account. Instead, if the underlying snapshot and encryption key were shared with you, you can copy the snapshot while re-encrypting it with a key of your own. You own the copied snapshot, and can register it as a new AMI.
- You can't copy an AMI with an associated `billingProduct` code that was shared with you from another account. This includes Windows AMIs and AMIs from the AWS Marketplace. To copy a shared AMI with a `billingProduct` code, launch an EC2 instance in your account using the shared AMI and then create an AMI from the instance. For more information, see [Creating an Amazon EBS-Backed Linux AMI \(p. 101\)](#).

Encryption and AMI Copy

Encrypting during AMI copy applies only to Amazon EBS-backed AMIs. Because an instance store-backed AMI does not rely on snapshots, you cannot use AMI copy to change its encryption status.

You can use AMI copy to create a new AMI backed by encrypted Amazon EBS snapshots. If you invoke encryption while copying an AMI, each snapshot taken of its associated Amazon EBS volumes—including the root volume—is encrypted using a key that you specify. For more information about using AMIs with encrypted snapshots, see [AMIs with Encrypted Snapshots \(p. 136\)](#).

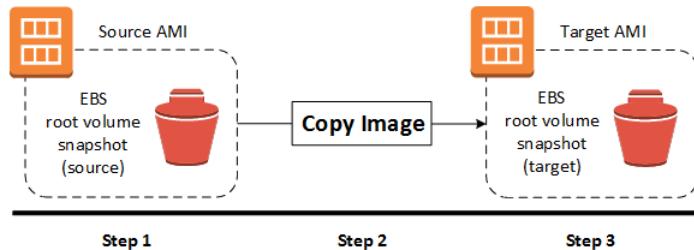
By default, the backing snapshot of an AMI is copied with its original encryption status. Copying an AMI backed by an unencrypted snapshot results in an identical target snapshot that is also unencrypted. If the source AMI is backed by an encrypted snapshot, copying it results in a target snapshot encrypted to the specified key. Copying an AMI backed by multiple snapshots preserves the source encryption status in each target snapshot. For more information about copying AMIs with multiple snapshots, see [AMIs with Encrypted Snapshots \(p. 136\)](#).

The following table shows encryption support for various scenarios. Note that 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

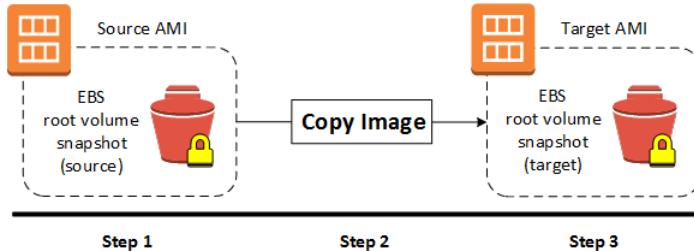
Copy an unencrypted source AMI to an unencrypted target AMI

In this scenario, a copy of an AMI with an unencrypted single backing snapshot is created in the specified geographical region (not shown). Although this diagram shows an AMI with a single backing snapshot, you can also copy an AMI with multiple snapshots. The encryption status of each snapshot is preserved. Therefore, an unencrypted snapshot in the source AMI results in an unencrypted snapshot in the target AMI, and an encrypted snapshot in the source AMI results in an encrypted snapshot in the target AMI.



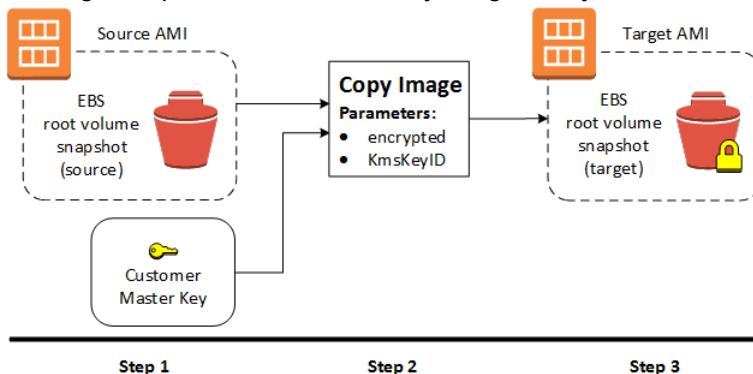
Copy an encrypted source AMI to an encrypted target AMI

Although this scenario involves encrypted snapshots, it is functionally equivalent to the previous scenario. If you apply encryption while copying a multi-snapshot AMI, all of the target snapshots are encrypted using the specified key or the default key if none is specified.



Copy an unencrypted source AMI to an encrypted target AMI

In this scenario, copying an AMI changes the encryption status of the destination image, for instance, by encrypting an unencrypted snapshot, or re-encrypting an encrypted snapshot with a different key. To apply encryption during the copy, you must provide an encryption flag and key. Volumes created from the target snapshot are accessible only using this key.



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 \(p. 101\)](#) and [Finding a Linux AMI \(p. 87\)](#).

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.
 - **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)

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 \(p. 85\)](#).

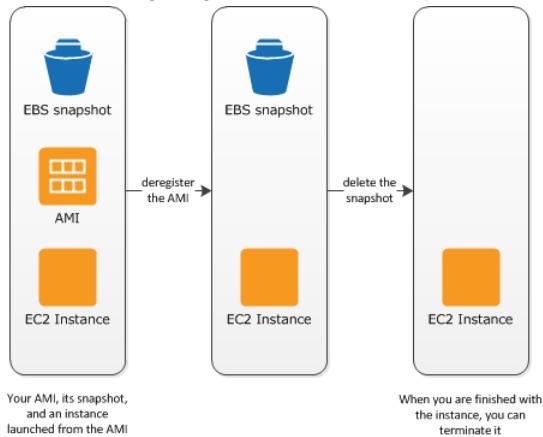
Contents

- [Cleaning Up Your Amazon EBS-Backed AMI \(p. 144\)](#)
- [Cleaning Up Your Instance Store-Backed AMI \(p. 145\)](#)

Cleaning Up Your Amazon EBS-Backed AMI

When you deregister an Amazon EBS-backed AMI, it doesn't affect the snapshot that was created for the root volume of the instance during the AMI creation process. You'll continue to incur storage costs for this snapshot. Therefore, if you are finished with the snapshot, you should delete it.

The following diagram illustrates the process for cleaning up your Amazon EBS-backed 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

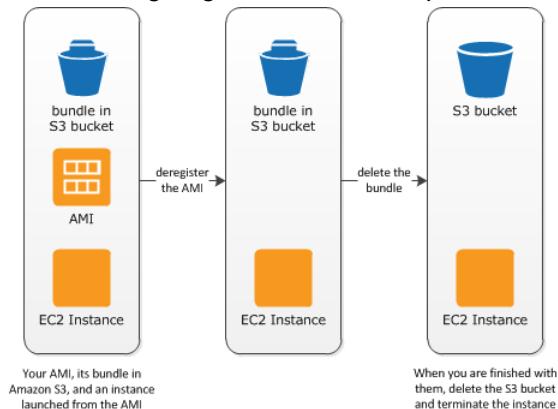
It may take a few minutes before the console removes the AMI from the list. Choose **Refresh** to refresh the status.

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. 126) (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 to all instances running Amazon Linux. Many applications developed on CentOS (and similar distributions) run on Amazon Linux.

AWS provides two versions of Amazon Linux: Amazon Linux 2 and the Amazon Linux AMI. For more information, including the complete list of AMIs, see [Amazon Linux 2](#) and [Amazon Linux AMI](#). For Amazon Linux Docker container images, see [amazonlinux](#) on Docker Hub.

If you are migrating from another Linux distribution to Amazon Linux, we recommend that you migrate to Amazon Linux 2. 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 image. Install your application on Amazon Linux 2, plus any packages required by your application. Test your application, and make any changes required for it to run on Amazon Linux 2. For more information about running Amazon Linux outside AWS, see [Running Amazon Linux 2 as a Virtual Machine On-Premises \(p. 153\)](#).

Contents

- [Connecting to an Amazon Linux Instance \(p. 146\)](#)
- [Identifying Amazon Linux Images \(p. 146\)](#)
- [AWS Command Line Tools \(p. 148\)](#)
- [Package Repository \(p. 148\)](#)
- [Extras Library \(Amazon Linux 2\) \(p. 150\)](#)
- [Accessing Source Packages for Reference \(p. 151\)](#)
- [cloud-init \(p. 151\)](#)
- [Subscribing to Amazon Linux Notifications \(p. 153\)](#)
- [Running Amazon Linux 2 as a Virtual Machine On-Premises \(p. 153\)](#)

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 log in, 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="2017.12"
image_arch="x86_64"
image_file="amzn2-ami-hvm-2017.12.0.20180509-x86_64.xfs.gpt"
image_stamp="c763-2a26"
image_date="20180509235735"
recipe_name="amzn2 ami"
recipe_id="0b8118bb-6f98-5619-0618-9570-26ae-ad18-178b79f1"
```

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 release 2.0 (2017.12) LTS Release Candidate
```

The following is an example of `/etc/os-release` for Amazon Linux 2:

```
[ec2-user ~]$ cat /etc/os-release
NAME="Amazon Linux"
VERSION="2 (2017.12)"
ID="amzn"
ID_LIKE="centos rhel fedora"
VERSION_ID="2"
PRETTY_NAME="Amazon Linux 2 (2017.12) LTS Release Candidate"
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-2018.03.0.20180508-x86_64.ext4.gpt"
image_stamp="c4fa-96f6"
image_date="20180508174933"
recipe_name="amzn ami"
recipe_id="1cb2680c-5325-ae3b-9562-95c2-ae09-5817-8b40e17d"
```

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-ec2
- 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 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 \(p. 151\)](#).

Security Updates

Security updates are provided using the package repositories as well as updated AMIs. 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 security updates at launch time. 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:

security

Apply outstanding 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
```

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 AMIs 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 \(p. 855\)](#). 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/v2/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**, type 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/v2/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**.

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, you need to do the following:

- [Step 1: Prepare the `seed.iso` Boot Image \(p. 154\)](#)
- [Step 2: Download the Amazon Linux 2 VM Image \(p. 156\)](#)
- [Step 3: Boot and Connect to Your New VM \(p. 156\)](#)

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 only includes 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` to store your `meta-data` and `user-data` configuration files.
2. Create the `meta-data` configuration file.
 - a. Add the VM's host name.

```
local-hostname: vm_hostname
```

- b. Specify any custom network settings, such as the network interface name.

```
#network-interfaces: |
#  iface interface_name inet static
```

For example, the following code block shows the contents of a `meta-data` configuration file that specifies the VM hostname (`amazonlinux.onprem`), configures the default network interface (`eth0`), and specifies static IP addresses for the necessary network devices.

```
local-hostname: amazonlinux.onprem
# eth0 is the default network interface enabled in the image. You can configure static
network settings with an entry like the following.
network-interfaces: |
  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
```

3. Create the `user-data` configuration file.

- a. Specify a custom password, in plaintext format, for the default ec2-user user account:

```
#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:'.
```

Note

Be sure to replace the *plain_text_password* placeholder with a plaintext password of your choice.

- b. (Optional) Create additional user accounts and specify their access mechanisms, passwords, and key pairs. For more information about the supported directives, see [Modules](#).
- c. (Optional) By default, cloud-init applies network settings each time the VM boots. Add the following code to the user-data configuration file 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
```

For example, the following code block shows the contents of a user-data configuration file that creates three additional users, specifies a custom password for the default ec2-user user account, and prevents cloud-init from applying network settings at each boot.

```
#cloud-config
# vim:syntax=yaml
users:
# A user by the name ec2-user is created in the image by default.
- default
# The following entry creates user1 and assigns a plain text password.
# Please note that the use of a plain text password is not recommended from security
best practices standpoint.
- name: user1
  groups: sudo
  sudo: ['ALL=(ALL) NOPASSWD:ALL']
  plain_text_passwd: myp@ssw0rd
  lock_passwd: false
# The following entry creates user2 and attaches a hashed password to the user.
# Hashed passwords can be generated with the following command on Amazon Linux 2:
# python -c 'import crypt,getpass; print(crypt.crypt(getpass.getpass()))'
- name: user2
  passwd: hashed-password
  lock_passwd: false
# The following entry creates user3, disables password-based login and enables an SSH
public key.
- name: user3
  ssh-authorized-keys:
```

```
- ssh-public-key-information
  lock_passwd: true
chpasswd:
  list: |
    ec2-user:myp@ssw0rd
# In the above line, do not add any spaces after 'ec2-user:'.

# NOTE: Cloud-init applies network settings on every boot by default. To retain network
# settings from first
boot, uncomment the 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
```

4. Place your `meta-data` and `user-data` configuration files in the `seedconfig` folder created in **Step 1**.
5. 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).

Note

You must connect the `seed.iso` boot image to the VM on first boot.

After the VM has booted, log in using one of the user accounts defined in the `user-data` configuration file. You can disconnect the boot image from the VM after you have logged in for the first time.

User Provided Kernels

If you have a need for 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 modify the `menu.1st` file 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 \(p. 86\)](#).

Contents

- [HVM AMIs \(GRUB\) \(p. 157\)](#)
- [Paravirtual AMIs \(PV-GRUB\) \(p. 158\)](#)

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 allows it to work with all currently supported Linux distributions. The most common bootloader is GRUB, and the following section describes configuring GRUB to use a custom kernel.

Configuring GRUB for HVM AMIs

The following is an example of a `menu.1st` configuration file for an HVM AMI. 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 that 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.

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 \(p. 941\)](#). If you are installing a custom kernel, you should consider enabling GRUB output by deleting the `hiddenmenu` line and adding `serial` and `terminal` lines to `/boot/grub/menu.1st` as shown in the example below.

Important

Avoid printing large amounts of debug information during the boot process; the serial console does not support high rate data transfer.

```
default=0
fallback=1
timeout=5
serial --unit=0 --speed=9600
terminal --dumb --timeout=5 serial console

title Vanilla Linux 4.16.4
root (hd0)
kernel /boot/vmlinuz-4.16.4 root=LABEL=/ console=tty1 console=ttyS0
initrd /boot/initrd.img-4.16.4

title Amazon Linux 2018.03 (4.14.26-46.32.amzn1.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-4.14.26-46.32.amzn1.x86_64 root=LABEL=/ console=tty1 console=ttyS0
initrd /boot/initramfs-4.14.26-46.32.amzn1.x86_64.img
```

You don't need to specify a fallback kernel in your `menu.1st` file, 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 allows the instance to boot even if the new kernel isn't found.

If your new Vanilla Linux kernel fails, the output will be similar to the example below.

```
^M Entry 0 will be booted automatically in 3 seconds. ^M Entry 0 will be booted
automatically in 2 seconds. ^M Entry 0 will be booted automatically in 1 seconds.

Error 13: Invalid or unsupported executable format
[ 0.000000] Initializing cgroup subsys cpuset
```

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 as shown in [Configuring GRUB \(p. 159\)](#) to boot with that kernel.

You can verify that the kernel image for an AMI is a PV-GRUB AKI by executing the following [describe-images](#) command with the Amazon EC2 command line tools (substituting the kernel image ID you want to check):

```
aws ec2 describe-images --filters Name=image-id,Values=aki-880531cd
```

Check whether the `Name` field starts with `pv-grub`.

Topics

- [Limitations of PV-GRUB \(p. 158\)](#)
- [Configuring GRUB for Paravirtual AMIs \(p. 159\)](#)
- [Amazon PV-GRUB Kernel Image IDs \(p. 159\)](#)
- [Updating PV-GRUB \(p. 161\)](#)

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.lst` file must exist in the image; the most common location for this file is `/boot/grub/menu.lst`.

The following is an example of a `menu.lst` 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 AMIs are available in all Amazon EC2 regions. There are AMIs for both 32-bit and 64-bit architecture types. Most modern AMIs use a PV-GRUB AMI 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
```

Note that 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, EU (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, EU (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-912fbcf8	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
aki-5c21674b	pv-grub-hd0_1.05-x86_64.gz

us-gov-west-1, AWS GovCloud (US)

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 AMI, as not all versions of the PV-GRUB AMI 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.

2. 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 \(p. 159\)](#)), 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 \(p. 159\)](#).
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
```

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 \(p. 19\)](#)
- [Getting Started with Amazon EC2 Linux Instances \(p. 26\)](#)
- [Instance Lifecycle \(p. 350\)](#)

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 \(p. 163\)](#).

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 \(p. 238\)](#).

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 \(p. 84\)](#).

Q. Would I benefit from using a virtual private cloud?

If you can launch instances in either EC2-Classic or EC2-VPC, you'll need to decide which platform meets your needs. For more information, see [Supported Platforms \(p. 637\)](#) and [Amazon EC2 and Amazon Virtual Private Cloud \(p. 630\)](#).

Q. Can I remotely manage a fleet of EC2 instances and machines in my hybrid environment?

Amazon Elastic Compute Cloud (Amazon EC2) Run Command lets you remotely and securely manage the configuration of your Amazon EC2 instances, virtual machines (VMs) and servers in hybrid environments, or VMs from other cloud providers. For more information, see [Systems Manager Remote Management \(Run Command\)](#).

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 \(p. 164\)](#)
- [Hardware Specifications \(p. 165\)](#)
- [Virtualization Types \(p. 166\)](#)
- [Networking and Storage Features \(p. 166\)](#)
- [Instance Limits \(p. 168\)](#)

Available Instance Types

Amazon EC2 provides the instance types listed in the following tables.

Current Generation Instances

For the best performance, we recommend that you use the current generation instance types when you launch new instances.

For more information about the current generation instance types, see [Amazon EC2 Instance Types](#).

Instance Family	Current Generation Instance Types
General purpose	t2.nano t2.micro t2.small t2.medium t2.large t2.xlarge t2.2xlarge m4.large m4.xlarge m4.2xlarge m4.4xlarge m4.10xlarge m4.16xlarge m5.large m5.xlarge m5.2xlarge m5.4xlarge m5.12xlarge m5.24xlarge m5d.large m5d.xlarge m5d.2xlarge m5d.4xlarge m5d.12xlarge m5d.24xlarge
Compute optimized	c4.large c4.xlarge c4.2xlarge c4.4xlarge c4.8xlarge c5.large c5.xlarge c5.2xlarge c5.4xlarge c5.9xlarge c5.18xlarge c5d.xlarge c5d.2xlarge c5d.4xlarge c5d.9xlarge c5d.18xlarge
Memory optimized	r4.large r4.xlarge r4.2xlarge r4.4xlarge r4.8xlarge r4.16xlarge r5.large r5.xlarge r5.2xlarge r5.4xlarge r5.12xlarge r5.24xlarge r5d.large r5d.xlarge r5d.2xlarge r5d.4xlarge r5d.12xlarge r5d.24xlarge x1.16xlarge x1.32xlarge x1e.xlarge x1e.2xlarge x1e.4xlarge x1e.8xlarge x1e.16xlarge x1e.32xlarge z1d.large z1d.xlarge z1d.2xlarge z1d.3xlarge z1d.6xlarge z1d.12xlarge
Storage optimized	d2.xlarge d2.2xlarge d2.4xlarge d2.8xlarge h1.2xlarge h1.4xlarge h1.8xlarge h1.16xlarge i3.large i3.xlarge i3.2xlarge i3.4xlarge i3.8xlarge i3.16xlarge i3.metal

Instance Family	Current Generation Instance Types
Accelerated computing	f1.2xlarge f1.16xlarge g3.4xlarge g3.8xlarge g3.16xlarge p2.xlarge p2.8xlarge p2.16xlarge p3.2xlarge p3.8xlarge p3.16xlarge

Previous Generation Instances

Amazon Web Services offers previous generation instances for users who have optimized their applications around these instances and have yet to upgrade. We encourage you to use the latest generation of instances to get the best performance, but we continue to support these previous generation instances. If you are currently using a previous generation instance, you can see which current generation instance would be a suitable upgrade. For more information, see [Previous Generation Instances](#).

Instance Family	Previous Generation Instance Types
General purpose	m1.small m1.medium m1.large m1.xlarge m3.medium m3.large m3.xlarge m3.2xlarge
Compute optimized	c1.medium c1.xlarge cc2.8xlarge c3.large c3.xlarge c3.2xlarge c3.4xlarge c3.8xlarge
Memory optimized	m2.xlarge m2.2xlarge m2.4xlarge cr1.8xlarge r3.large r3.xlarge r3.2xlarge r3.4xlarge r3.8xlarge
Storage optimized	hs1.8xlarge i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge
GPU optimized	g2.2xlarge g2.8xlarge
Micro	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 \(p. 228\)](#).

Note

Amazon EC2 instances 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.

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 \(p. 86\)](#).

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.

Networking and Storage Features

When you select an instance type, this determines the networking and storage features that are available.

Networking features

- Some instance types are not available in EC2-Classic, so you must launch them in a VPC. By launching an instance in a VPC, you can leverage features that are not available in EC2-Classic, such as enhanced networking, assigning multiple private IPv4 addresses to an instance, assigning IPv6 addresses to an instance, and changing the security groups assigned to an instance. For more information, see [Instance Types Available Only in a VPC \(p. 636\)](#).
- 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 \(p. 701\)](#).
 - 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 \(p. 708\)](#).
- 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 supported MTU 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 \(p. 706\)](#).

Storage features

- Some instance types support EBS volumes and instance store volumes, while other instance types support only EBS volumes. Some instances that support instance store volumes use solid state drives (SSD) to deliver very high random I/O performance. For more information, see [Storage \(p. 730\)](#).
- 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 \(p. 812\)](#).

The following table summarizes the networking and storage features supported by the current generation instance types.

	VPC only	EBS only	Instance store	Placement group	Enhanced networking
C4	Yes	Yes		Yes	Intel 82599 VF
C5	Yes	Yes		Yes	ENA
C5d	Yes		NVMe *	Yes	ENA
D2			HDD	Yes	Intel 82599 VF
F1	Yes		NVMe *	Yes	ENA
G3	Yes	Yes		Yes	ENA
H1	Yes		HDD	Yes	ENA
I3	Yes		NVMe *	Yes	ENA
M4	Yes	Yes		Yes	m4.16xlarge: ENA All other sizes: Intel 82599 VF
M5	Yes	Yes		Yes	ENA
M5d	Yes		NVMe *	Yes	ENA
P2	Yes	Yes		Yes	ENA
P3	Yes	Yes		Yes	ENA
R4	Yes	Yes		Yes	ENA
R5	Yes	Yes		Yes	ENA
R5d	Yes		NVMe *	Yes	ENA
T2	Yes	Yes			
X1	Yes		SSD	Yes	ENA
X1e	Yes		SSD	Yes	
z1d	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 earlier generation instance types.

	VPC only	EBS only	Instance store	Placement group	Enhanced networking
C3			SSD	Yes	Intel 82599 VF

	VPC only	EBS only	Instance store	Placement group	Enhanced networking
G2			SSD	Yes	
I2			SSD	Yes	Intel 82599 VF
M3			SSD		
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 Limits \(p. 896\)](#).

T2 Instances

T2 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. T2 instances are well suited for a wide range of general-purpose applications like microservices, low-latency interactive applications, small and medium databases, virtual desktops, development, build, and stage environments, code repositories, and product prototypes.

For more information about T2 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 within certain usage limits. For more information, see [AWS Free Tier](#).

Contents

- [Hardware Specifications \(p. 168\)](#)
- [T2 Instance Requirements \(p. 168\)](#)
- [Best Practices \(p. 169\)](#)
- [CPU Credits and Baseline Performance \(p. 169\)](#)
- [T2 Standard \(p. 171\)](#)
- [T2 Unlimited \(p. 180\)](#)
- [Working With T2 Instances \(p. 183\)](#)
- [Monitoring Your CPU Credits \(p. 186\)](#)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instance Types](#).

T2 Instance Requirements

The following are the requirements for T2 instances:

- You must launch your T2 instances into a virtual private cloud (VPC); they are not supported on the EC2-Classic platform. Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. You cannot change the instance type of an existing instance in EC2-Classic to a T2 instance type. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms \(p. 637\)](#). For more information about launching a VPC-only instance, see [Instance Types Available Only in a VPC \(p. 636\)](#).
- You must launch a T2 instance using an HVM AMI. For more information, see [Linux AMI Virtualization Types \(p. 86\)](#).
- You must launch your T2 instances using an Amazon EBS volume as the root device. For more information, see [Amazon EC2 Root Device Volume \(p. 13\)](#).
- T2 instances are available as On-Demand Instances, Reserved Instances, and Spot Instances, but they are not available as Scheduled Instances or Dedicated Instances. They are also not supported on a Dedicated Host. For more information about these options, see [Instance Purchasing Options \(p. 238\)](#).
- 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. By default, you can run up to 20 T2 instances simultaneously. If you need more T2 instances, you can request them using the [Amazon EC2 Instance Request Form](#).
- Ensure that the T2 instance size that 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) may require a `t2.micro`, or larger, instance size for many use cases. As the memory and CPU requirements of your workload grows over time, you can scale to larger T2 instance sizes, or other EC2 instance types.

Best Practices

Follow these best practices to get the maximum benefit from and satisfaction with T2 instances.

- Use a recommended AMI

For Linux T2 instances, we recommend the [latest Amazon Linux 2 AMI](#).

- Turn on auto-recover

You can create a CloudWatch alarm that monitors an EC2 instance and automatically recovers the instance if it becomes impaired for any reason. For more information, see [Adding Recover Actions to Amazon CloudWatch Alarms \(p. 505\)](#).

CPU Credits and Baseline Performance

Traditional Amazon EC2 instance types provide fixed performance, while T2 instances provide a baseline level of CPU performance with the ability to burst above that baseline level. The baseline performance and ability to burst are governed by CPU credits. A CPU credit provides the performance of a full CPU core for one minute.

Topics

- [CPU Credits \(p. 169\)](#)
- [Baseline Performance \(p. 171\)](#)

CPU Credits

One CPU credit is equal to one vCPU running at 100% utilization 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.

Earning CPU Credits

Each T2 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 T2 instance uses fewer CPU resources than is required for baseline performance (such as when it is idle), the unspent CPU credits are accrued in the CPU credit balance. If a T2 instance needs to burst above the baseline performance level, it spends the accrued credits. The more credits a T2 instance has accrued, the more time it can burst beyond its baseline when more performance is needed.

The following table lists 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 performance level as a percentage of a full core performance (using a single vCPU).

Instance type	CPU credits earned per hour	Maximum earned credits that can be accrued*	vCPUs	Baseline performance per vCPU
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	1944	8	17%**

* The number of credits that can be accrued is equivalent to the number of credits that can be earned in a 24-hour period. For T2 Standard, launch credits can also be accrued, and do not count towards the maximum earned credits that can be accrued.

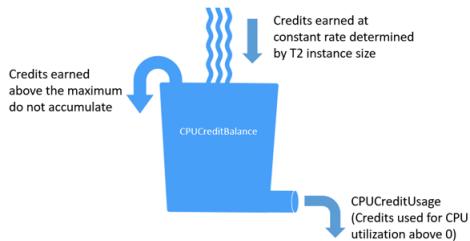
** t2.medium and larger instances have more than one vCPU. The baseline performance in the table is per vCPU. To calculate the baseline CPU utilization for the *instance*, multiply the vCPU percentage by the number of vCPUs. For example, a t2.large instance has two vCPUs, which provides a baseline CPU utilization for the instance of 60% (2 vCPUs x 30% baseline performance of one vCPU). In CloudWatch, CPU utilization is shown per vCPU; therefore the CPU utilization for a t2.large instance operating at the baseline performance is shown as 30% in CloudWatch CPU metrics.

CPU Credit Earn Rate

The number of CPU credits earned per hour is determined by the instance size. For example, a t2.nano earns three credits per hour, while a t2.small earns 12 credits per hour. The preceding table lists the credit earn rate for all T2 instances.

CPU Credit Accrual Limit

While earned credits never expire on a running instance, there is a limit to the number of earned credits an instance can accrue. The limit is determined by the CPU credit balance limit. Once the limit is reached, any new credits that are earned are discarded, indicated by the following image: the full bucket indicates the CPU credit balance limit, and the spillover indicates newly earned credits that exceed the limit.



The CPU credit balance limit differs for each T2 instance size. For example, a `t2.micro` instance can accrue a maximum of 144 earned CPU credits in the CPU credit balance. The preceding table lists the maximum number of earned credits that each T2 instance can accrue.

Launch credits do not count towards the CPU credit balance limit. If a T2 Standard 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 \(p. 172\)](#).

Accrued CPU Credits Are Lost When an Instance Is Stopped

CPU credits on a running instance do not expire. However, the CPU credit balance does not persist between instance stops and starts; if you stop an instance, the instance loses all its accrued credits. For more information, see `CPUCreditBalance` in the [CloudWatch metrics table \(p. 187\)](#).

Baseline Performance

The number of credits an instance earns per hour can be expressed as a percentage of CPU utilization, and is known as the *baseline performance*, and sometimes just as *the baseline*. For example, a `t2.nano` earns three credits per hour resulting in a baseline performance of 5% (3/60 minutes). A `t2.large`, with two vCPUs, earns 36 credits per hour, resulting in a baseline performance of 30% (18/60 minutes) per vCPU.

T2 Standard

A T2 Standard instance is suited to workloads with an average CPU utilization that is consistently below the baseline performance 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, performance is gradually lowered to the baseline performance 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 Performance \(p. 169\)](#).

Tip

To ensure that your workloads always get the performance they need, switch to [T2 Unlimited \(p. 180\)](#) or consider using a larger T2 instance size.

Topics

- [T2 Standard Concepts \(p. 171\)](#)
- [Example: Explaining Credit Use with T2 Standard \(p. 173\)](#)

T2 Standard Concepts

How T2 Standard Works

A T2 Standard instance receives 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.

When a T2 Standard 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.

Launch Credits

T2 Standard instances get 30 launch credits per vCPU at launch or start. For example, a `t2.micro` has one vCPU and gets 30 launch credits, while a `t2.xlarge` 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, once the instance spends the 30 launch credits, the credit balance cannot exceed 144. For more information about the CPU credit balance limit for each T2 instance size, see the [T2 credit table \(p. 170\)](#).

The following table lists the initial CPU credit allocation received at launch or start, and the number of vCPUs.

Instance type	Launch credits	vCPUs
<code>t2.nano</code>	30	1
<code>t2.micro</code>	30	1
<code>t2.small</code>	30	1
<code>t2.medium</code>	60	2
<code>t2.large</code>	60	2
<code>t2.xlarge</code>	120	4
<code>t2.2xlarge</code>	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.

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

	Launch credits	Earned credits
		per instance size, see the T2 credit table (p. 170) .
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 T2 credit table (p. 170) .
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 an instance is running, earned credits that have accrued do not expire. When the 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 \(p. 187\)](#).

Example: 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 [T2 credit table \(p. 170\)](#). For more information about launch credit limits, see [Launch Credit Limits \(p. 172\)](#).

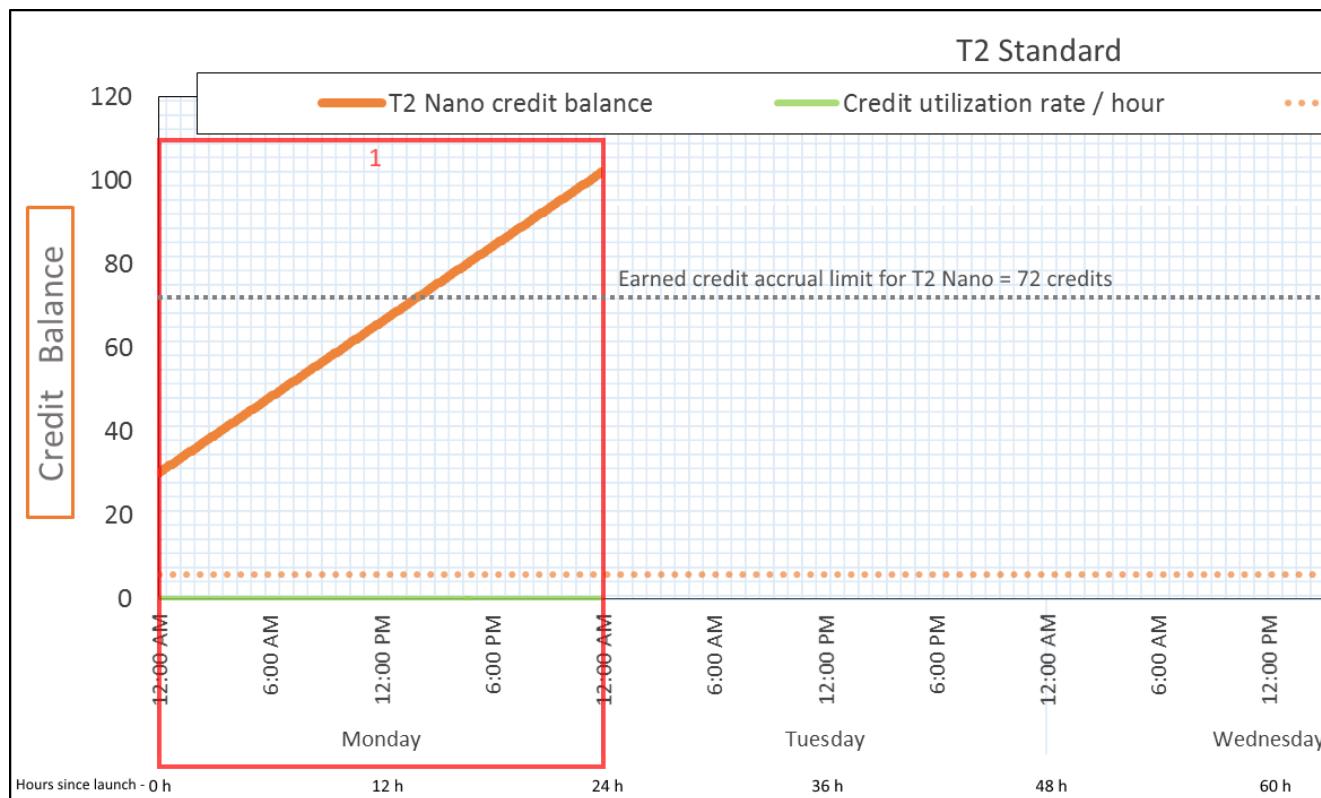
You might launch a T2 Standard instance and use it immediately, or you might launch a T2 Standard instance and leave it idle for a few days before running applications on it. Whether an instance is used or remains idle determines if credits are spent or accrued. If an instance remains idle for 24 hours 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)

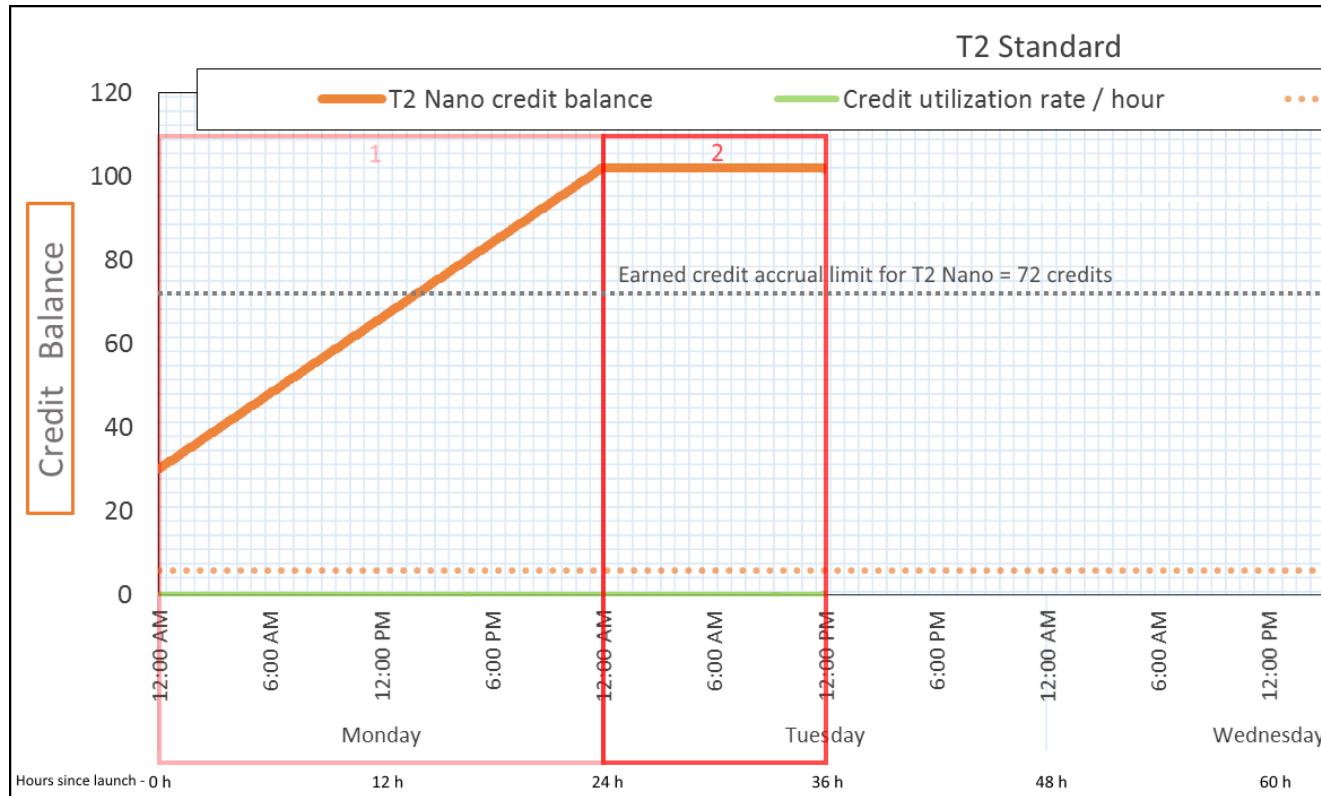
Conclusion

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.



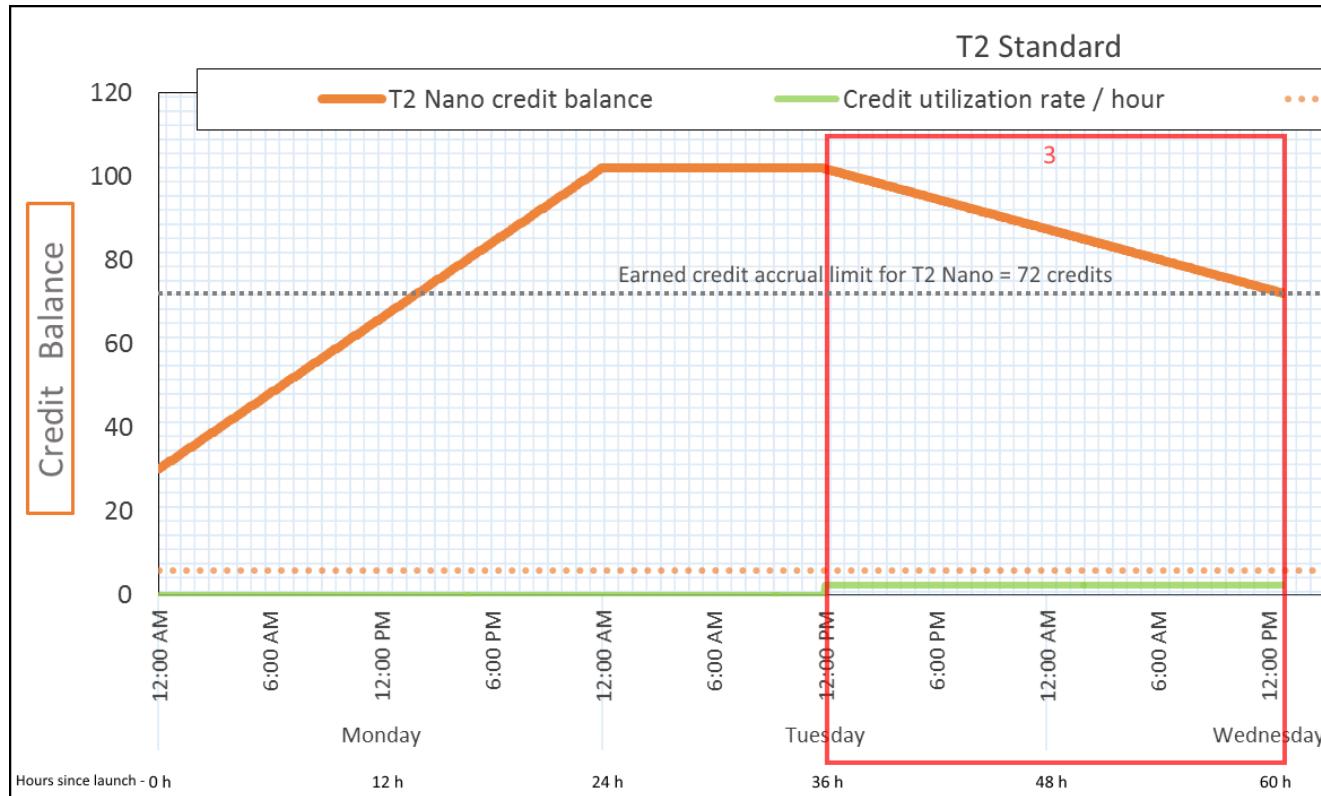
Credit Spend Rate	0 credits per 24 hours (0% CPU utilization)
Credit Earn Rate	72 credits per 24 hours (3 credits per hour)
Credit Discard Rate	72 credits per 24 hours (100% of credit earn rate)
Credit Balance	102 credits (30 launch credits + 72 earned credits) —balance is unchanged

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)

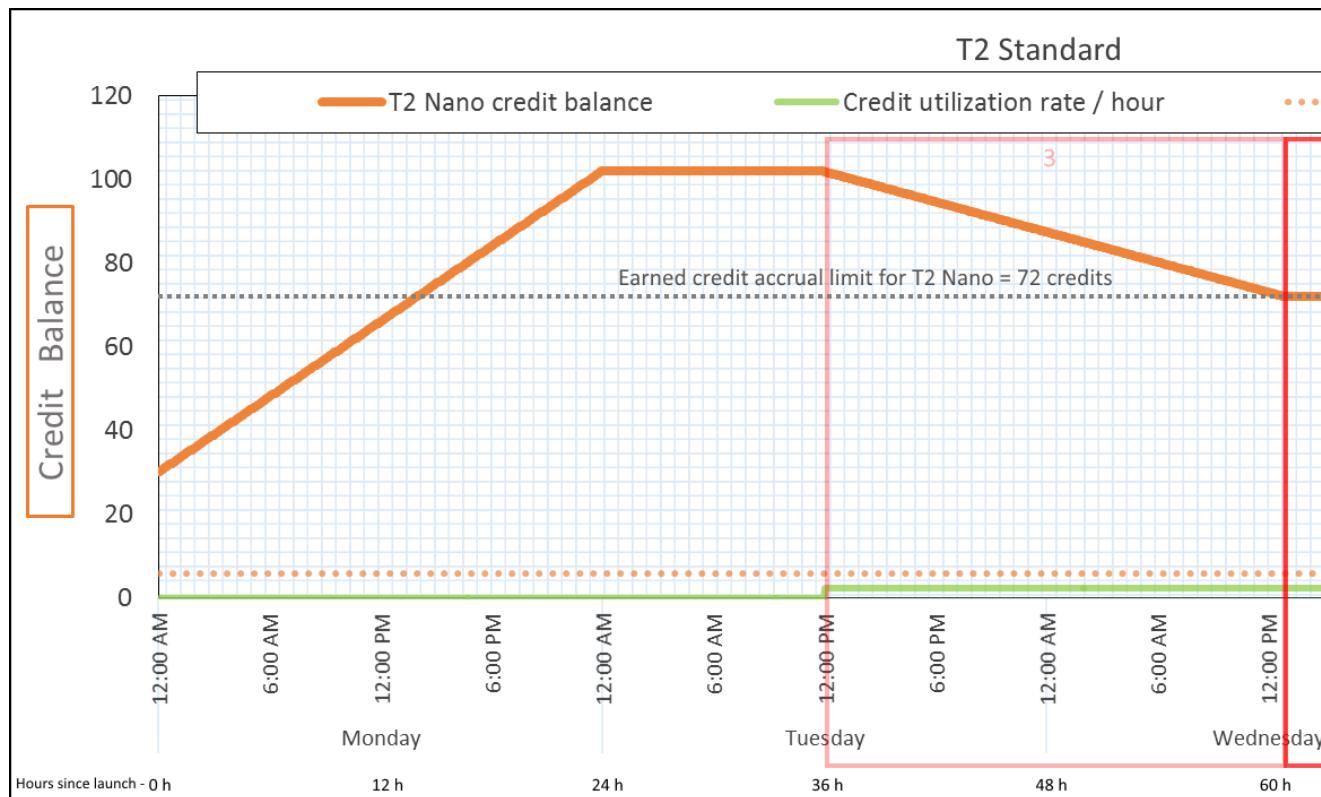
Conclusion

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 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.



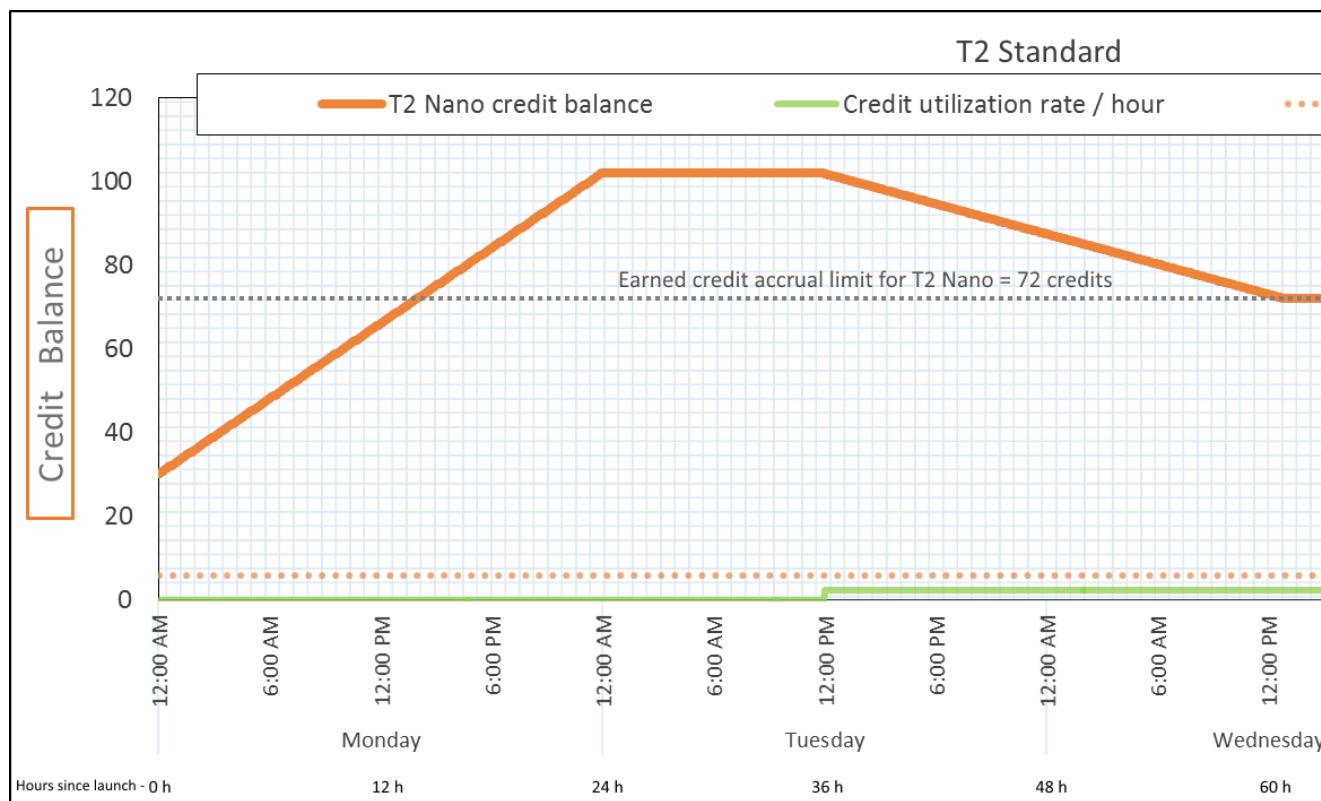
Credit Spend Rate	28.8 credits per 24 hours (1.2 credits per hour, 2% CPU utilization, 40% of credit earn rate)—13.2 credits over 11 hours
Credit Earn Rate	72 credits per 24 hours
Credit Discard Rate	43.2 credits per 24 hours (60% of credit earn rate)
Credit Balance	72 credits (0 launch credits, 72 earned credits)—balance is at its limit

Conclusion

After launch credits are spent, the credit balance limit is determined by the number of credits 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 288/24 + earn rate 72/24 = balance decrease rate 216/24)

Conclusion

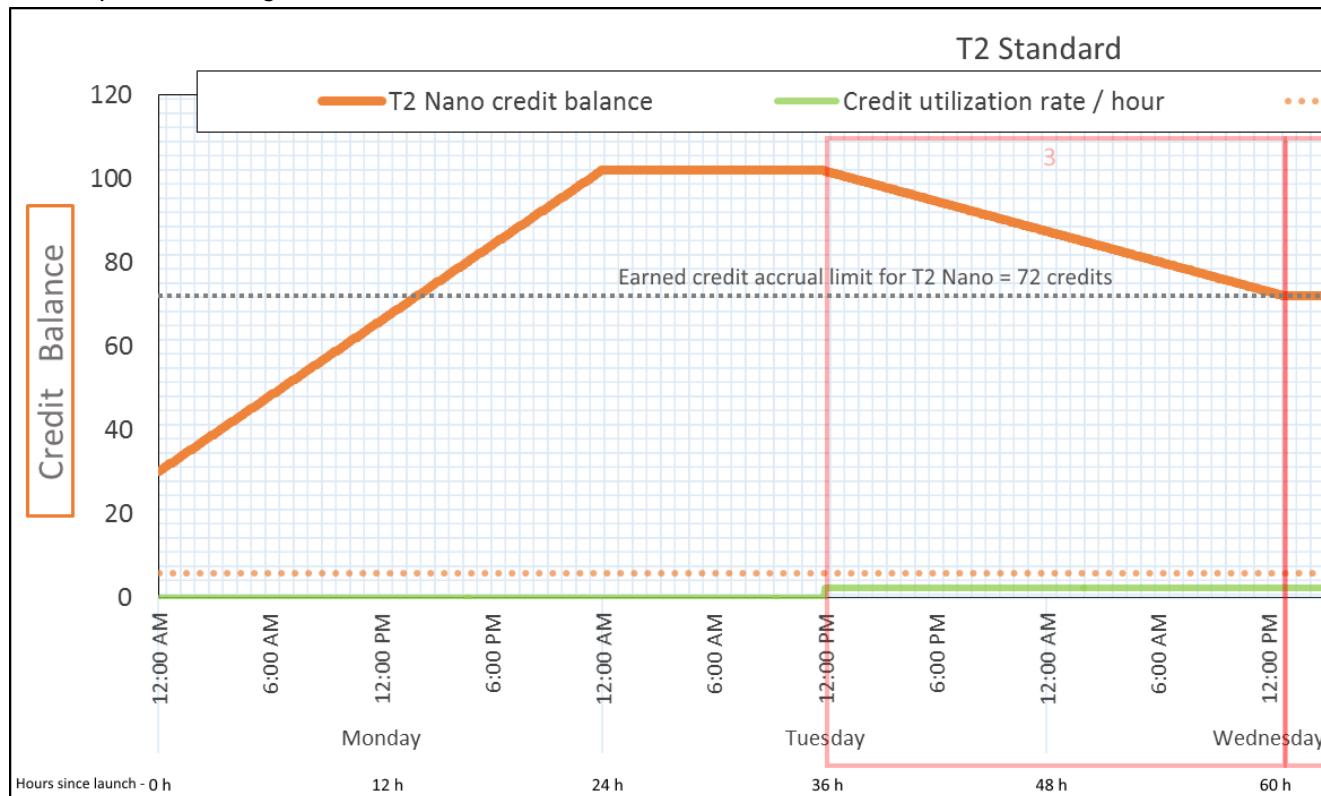
If an instance spends more credits than it earns, its credit balance decreases.

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, and 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)

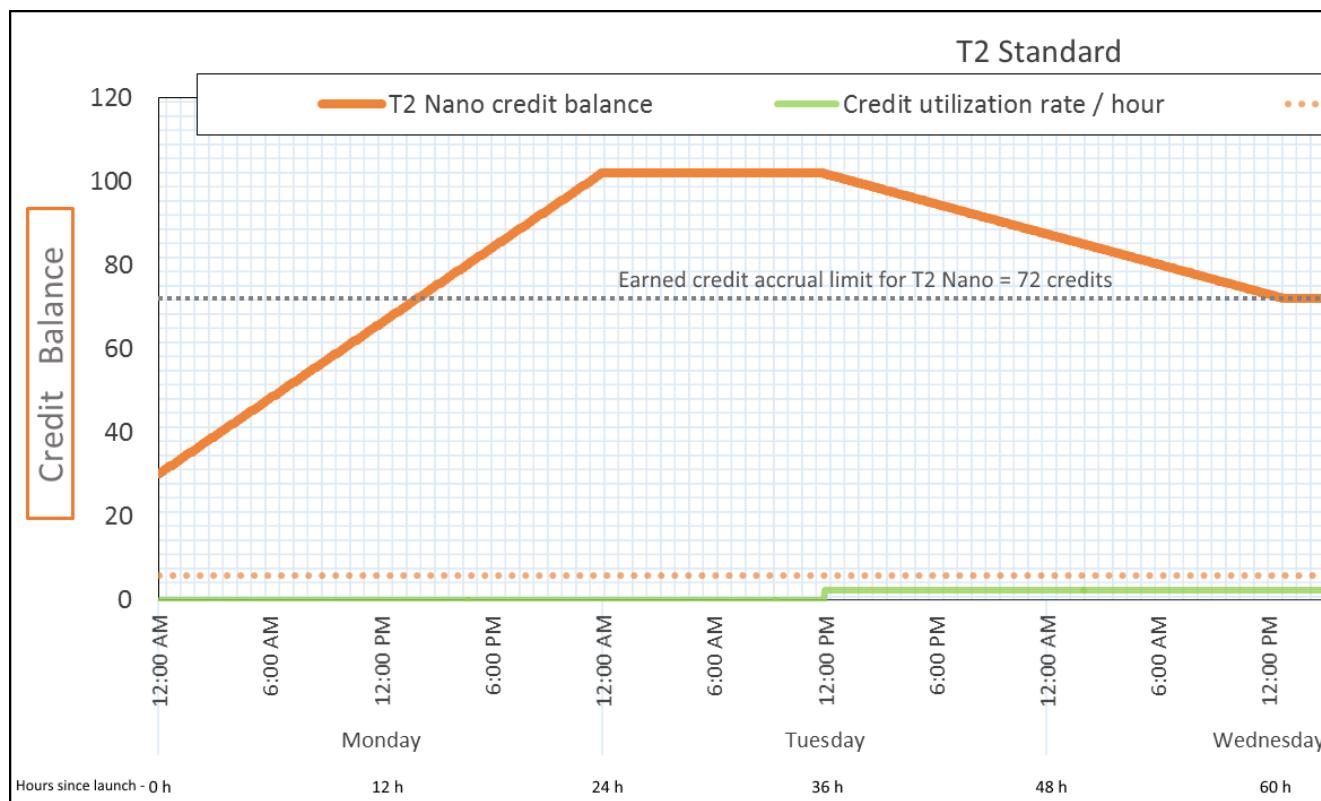
Conclusion

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 an instance can earn in 24 hours. For more information about credit balance limits, see the [T2 credit table \(p. 170\)](#).

T2 Unlimited

A T2 Unlimited instance can sustain high CPU performance for any period of time whenever required. The hourly T2 instance price automatically covers all interim spikes in usage 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. 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 T2 Unlimited Pricing section in [Amazon EC2 On-Demand Pricing](#).

Important

If you use a `t2.micro` instance under the [AWS Free Tier](#) offer and configure it as Unlimited, charges may apply if your average utilization over a rolling 24-hour period exceeds the baseline of the instance.

Topics

- [T2 Unlimited Concepts \(p. 181\)](#)
- [Example: Explaining Credit Use with T2 Unlimited \(p. 182\)](#)

T2 Unlimited Concepts

T2 Unlimited is a configuration option for T2 instances that can be set at launch, or enabled at any time for a running or stopped T2 instance.

T2 Unlimited instances can burst above the baseline for as long as required. This enables you to enjoy the low T2 instance hourly price for a wide variety of general-purpose applications, and ensures that your instances are never held to the baseline performance. The basic T2 hourly instance price automatically covers all CPU usage spikes if the average CPU utilization of a T2 Unlimited instance over a rolling 24-hour period is at or below the baseline. For a vast majority of general-purpose workloads, T2 Unlimited instances provide ample performance without any additional charges. If the average CPU utilization exceeds the baseline over a 24-hour period, there is a flat additional rate per vCPU-hour. For more information, see the T2 Unlimited Pricing section in [Amazon EC2 On-Demand Pricing](#).

How T2 Unlimited Works

If a T2 Unlimited instance depletes its CPU credit balance, it can spend *surplus* credits to burst beyond the baseline. When its CPU utilization falls below the baseline, it uses the CPU credits that it earns to pay down the surplus credits 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.

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 \(p. 170\)](#) in a 24-hour period (for example, a t2.micro can earn a maximum of 144 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 it has spent. The surplus credits that are not paid down are charged at a flat additional rate per vCPU-hour. For more information, see the T2 Unlimited Pricing section in [Amazon EC2 On-Demand Pricing](#).

Surplus credits that were spent earlier are charged when any of the following occurs:

- The spent surplus credits exceed the [maximum number of credits \(p. 170\)](#) 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 T2 Instances \(p. 186\)](#).

No Launch Credits for T2 Unlimited

T2 Unlimited instances do not receive launch credits. 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 an instance is switched from Standard to Unlimited, any accrued launch credits are removed from the `CPUCreditBalance` before the remaining `CPUCreditBalance` is carried over.

Enabling T2 Unlimited

T2 Standard is the default configuration; if you do not enable T2 Unlimited, your T2 instance launches as Standard. You can switch from Standard to Unlimited, and from Unlimited to Standard, at any time on a running or stopped instance. For more information, see [Launching a T2 Instance as Unlimited \(p. 184\)](#) and [Modifying the Credit Option for CPU Usage of a T2 Instance \(p. 186\)](#).

You can view if your T2 instance is configured as Standard or Unlimited using the Amazon EC2 console or the AWS CLI. For more information, see [Viewing the Credit Option for CPU Usage of a T2 Instance \(p. 185\)](#).

What Happens to Credits when Enabling or Disabling T2 Unlimited

`CPUCreditBalance` is a CloudWatch metric that tracks the number of credits a T2 Standard or T2 Unlimited instance has accrued. `CPUSurplusCreditBalance` is a CloudWatch metric that tracks the number of surplus credits a T2 Unlimited instance has spent.

When a T2 Standard instance is switched to Unlimited, the following occurs:

- Any launch credits are removed from the `CPUCreditBalance`, and the remaining `CPUCreditBalance` containing accrued earned credits is carried over.

When a T2 Unlimited instance is switched to Standard, the following occurs:

- The `CPUCreditBalance` remains unchanged and is carried over.
- The `CPUSurplusCreditBalance` is immediately charged.

Monitoring Credit Usage

To see if your T2 Unlimited instance is spending more credits than the baseline provides, you can use CloudWatch metrics to track and set up hourly alarms to be notified of credit usage. For more information, see [Monitoring Your CPU Credits \(p. 186\)](#).

Example: 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 performance.

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` 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:

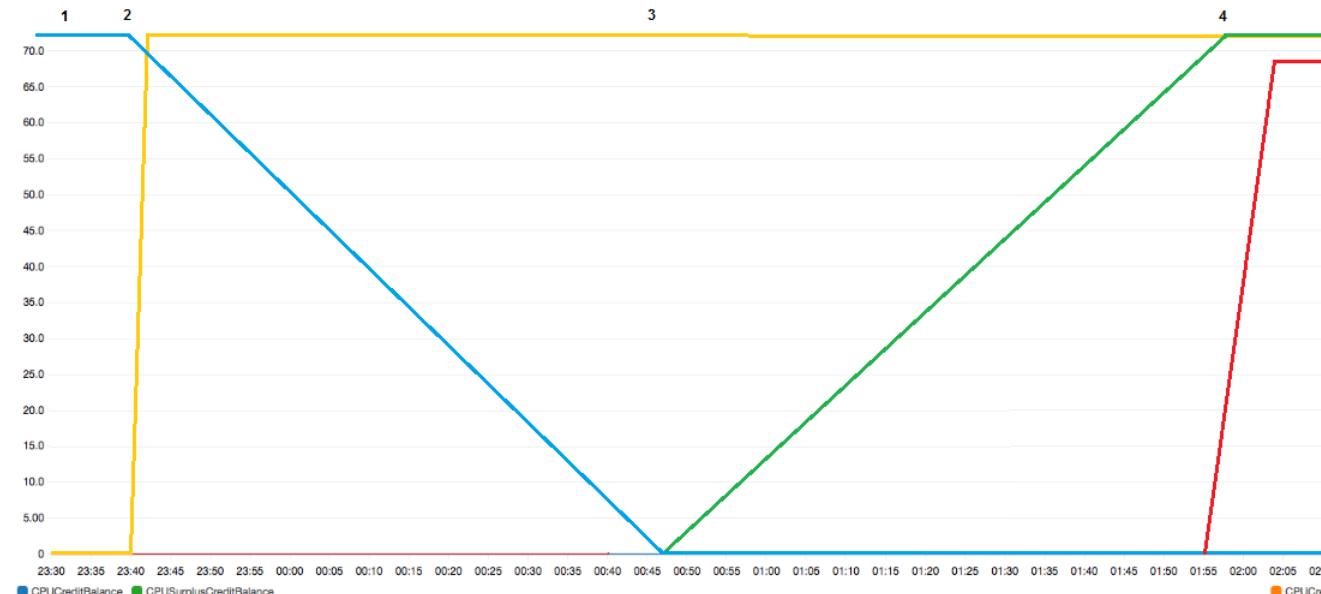
1 – In the first 10 minutes, `CPUCreditUsage` is at 0, and the `CPUCreditBalance` remains at its maximum of 72.

2 – At 23:40, as CPU utilization increases, the instance spends CPU credits and the `CPUCreditBalance` decreases.

3 – At around 00:47, the instance depletes its entire `CPUCreditBalance`, and starts to spend surplus credits to sustain high CPU performance.

4 – Surplus credits are spent until 01:55, when the CPUSurplusCreditBalance reaches 72 CPU credits. This is equal to the maximum a t2.nano 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 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	Statistic	Period
CPUCreditBalance	EC2 * InstanceId:i-0aa4b948d7eb37d6b * CPUCreditBalance	Maximum	5 Minutes
CPUCreditUsage	EC2 * InstanceId:i-0aa4b948d7eb37d6b * CPUCreditUsage	Maximum	5 Minutes
CPUSurplusCreditBalance	EC2 * InstanceId:i-0aa4b948d7eb37d6b * CPUSurplusCreditBalance	Maximum	5 Minutes
CPUSurplusCreditsCharged	EC2 * InstanceId:i-0aa4b948d7eb37d6b * CPUSurplusCreditsCharged	Maximum	5 Minutes

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.

Working With T2 Instances

Topics

- [Launching a T2 Instance as Unlimited \(p. 184\)](#)
- [Using an Auto Scaling Group to Launch a T2 Unlimited Instance \(p. 184\)](#)
- [Viewing the Credit Option for CPU Usage of a T2 Instance \(p. 185\)](#)
- [Modifying the Credit Option for CPU Usage of a T2 Instance \(p. 186\)](#)

Launching a T2 Instance as Unlimited

When you launch a T2 instance, the instance launches as standard by default. To launch a T2 instance as unlimited, you must specify the `unlimited` option.

You can launch a T2 Unlimited instance 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 T2 Unlimited Instance \(p. 184\)](#).

To launch a T2 instance as Unlimited 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 a T2 instance type, and choose **Next: Configure Instance Details**.

Note

T2 instance types are the only instance types that use CPU credits for CPU usage.

5. On the **Configure Instance Details** page, for **T2 Unlimited**, choose **Enable**, and then choose **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 T2 instance. For more information, see [Launching an Instance Using the Launch Instance Wizard \(p. 355\)](#).

To launch a T2 instance as Unlimited using the AWS CLI

- Launch a T2 instance using the `run-instances` command. Specify the credit option using the `--credit-specification CpuCredits=` parameter. Valid credit options are `standard` and `unlimited`. 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 t2.micro --key-name MyKeyPair --credit-specification "CpuCredits=unlimited"
```

Using an Auto Scaling Group to Launch a T2 Unlimited Instance

When T2 instances are launched or started, they require CPU credits for a good bootstrapping experience. If you use an Auto Scaling group to launch your T2 instances, we recommend that you configure the T2 instances as Unlimited so that they use surplus credits when they are automatically launched or restarted by the Auto Scaling group. Using surplus credits prevents performance restrictions.

You must use a launch template for launching a T2 instance as Unlimited in an Auto Scaling group; a launch configuration does not support launching a T2 instance as Unlimited.

To create a launch template that launches a T2 Unlimited instance using the AWS CLI

- To create a launch template that launches a T2 Unlimited instance, use the `create-launch-template` command and specify `unlimited` as the credit option for CPU usage.

Example

```
aws ec2 create-launch-template --launch-template-name MyLaunchTemplate  
--version-description FirstVersion --launch-template-data  
ImageId=ami-8c1be5f6,InstanceType=t2.medium,CreditSpecification={CpuCredits=unlimited}
```

To associate the launch template with an Auto Scaling group, create the group with the launch template, or add the launch template to an existing group.

To create an Auto Scaling group with a launch template using the AWS CLI

- Use the [create-auto-scaling-group](#) AWS CLI command and specify the --launch-template parameter.

To add a launch template to an Auto Scaling group using the AWS CLI

- Use the [update-auto-scaling-group](#) AWS CLI command and specify the --launch-template parameter.

For more information, see [Creating an Auto Scaling Group Using a Launch Template](#) in the *Amazon EC2 Auto Scaling User Guide*.

Viewing the Credit Option for CPU Usage of a T2 Instance

You can view the credit option (standard or unlimited) of a running or stopped T2 instance.

To view the credit option for CPU usage 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 T2 instance.
3. Choose **Description** and view the **T2 Unlimited** field.
 - If the value is **Enabled**, then your instance is configured as T2 Unlimited.
 - If the value is **Disabled**, then your instance is configured as T2 Standard.

To describe the credit option for CPU usage using the AWS CLI

- Describe the credit option for CPU usage using the [describe-instance-credit-specifications](#) command. If you do not specify one or more instance IDs, all T2 instances with the credit option of unlimited are returned.

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 Option for CPU Usage of a T2 Instance

You can switch the credit option for CPU usage of a running or stopped T2 instance at any time from standard to unlimited, and from unlimited to standard.

To modify the credit option for CPU usage of a T2 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 T2 instance. To modify the credit option for several T2 instances at once, select all applicable instances.
3. Choose **Actions, Instance Settings, Change T2 Unlimited**.

Note

The **Change T2 Unlimited** option is enabled only if you select a T2 instance.

4. To change the credit option to unlimited, choose **Enable**. To change the credit option to standard, choose **Disable**. The current credit option for the T2 instance appears in parentheses after the instance ID.

To modify the credit option for CPU usage of a T2 instance using the AWS CLI

- Modify the credit option for CPU usage for a T2 instance using the [modify-instance-credit-specification](#) command. Specify the instance and its credit option using the --instance-credit-specification parameter. Valid credit options are standard and unlimited.

```
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": []  
}
```

Monitoring Your CPU Credits

You can see the credit balance for each T2 instance in the Amazon EC2 per-instance metrics of the CloudWatch console.

Topics

- [Additional CloudWatch Metrics for T2 Instances \(p. 186\)](#)
- [Calculating CPU Credit Usage \(p. 188\)](#)

Additional CloudWatch Metrics for T2 Instances

T2 instances have four 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 a T2 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 performance when the CPUCreditBalance is zero.
- **CPUSurplusCreditsCharged** – The number of surplus CPU credits that exceed the [maximum number of CPU credits \(p. 170\)](#) that can be earned in a 24-hour period, and thus attract an additional charge.

The last two metrics apply only to T2 instances configured as **unlimited**.

The following table describes the CloudWatch metrics for T2 instances. For more information about using these metrics in CloudWatch, see [List the Available CloudWatch Metrics for Your Instances \(p. 485\)](#).

Metric	Description
CPUCreditUsage	<p>[T2 instances] 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>[T2 instances] 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. Once 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 the instance stops, the CPUCreditBalance 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>
CPUSurplusCreditBalance	[T2 Unlimited instances] The number of surplus credits that have been spent by a T2 Unlimited instance when its CPUCreditBalance is zero.

Metric	Description
	<p>The <code>CPUSurplusCreditBalance</code> is paid down by earned CPU credits. If the number of surplus credits exceeds the maximum number of credits 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>[T2 Unlimited instances] The number of spent surplus credits that are not paid down by earned CPU credits, and 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 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>Units: Credits (vCPU-minutes)</p>

Calculating CPU Credit Usage

The CPU credit usage of T2 Standard and T2 Unlimited instances is calculated using the T2 instance CloudWatch metrics described in the preceding table.

Amazon EC2 sends the metrics to CloudWatch every five minutes. A reference to *a 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 T2 Standard

- The CPU credit balance increases if CPU utilization is below the baseline, when credits spent are less than credits earned in the prior five-minute interval.
- The CPU credit balance decreases if CPU utilization is above the baseline, when credits spent are more than 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 it can accrue in the credit balance. For information about the number of credits earned per hour, and the credit balance limit for each T2 instance size, see [the T2 credit table \(p. 170\)](#).

Example

This example uses a `t2.micro` instance. To calculate the `CPUCreditBalance` 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 `t2.micro` 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 the instance earned in the past five minutes. A `t2.micro` 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 T2 Unlimited

When a T2 instance needs to burst above the baseline, it always spends its accrued credits before spending surplus credits. When it depletes its accrued CPU credit balance, it can spend surplus credits to burst 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 \(p. 170\)](#) 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 the `CPUCreditBalance` is set to 0. Again, the instance size determines the [maximum number of credits \(p. 170\)](#) it can accrue.

Example

```
CPUSurplusCreditBalance = min [max earned credit balance, -Adjusted balance]
CPUCreditBalance = 0
```

If the surplus credits spent exceed the maximum credits 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.

General Purpose Instances

General purpose instances provide a balance of compute, memory, and networking resources, and can be used for a variety of workloads.

M5 Instances

M5 instances are the latest generation in Amazon EC2's General purpose instance family. M5 instances give you 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. M5 instances are well-suited for the following applications:

- Web and application servers
- Small and medium databases
- Gaming servers
- Caching fleets
- Running backend servers for SAP, Microsoft SharePoint, cluster computing, and other enterprise applications

Note: M5 instances require EBS-backed AMIs with the NVMe and Elastic Network Adapter (ENA) drivers installed. For more information, see the [Release Notes \(p. 193\)](#).

T2 Instances

T2 instances provide a baseline level of CPU performance with the ability to burst to a higher level when required by your workload. A T2 Unlimited instance can sustain high CPU performance for any period of time whenever required. For more information, see [T2 Instances \(p. 168\)](#). T2 instances are well-suited for the following applications:

- Websites and web applications
- Code repositories
- Development, build, test, and staging environments
- Microservices

Contents

- [Hardware Specifications \(p. 191\)](#)
- [Instance Performance \(p. 192\)](#)
- [Network Performance \(p. 192\)](#)
- [Instance Features \(p. 192\)](#)

- [Release Notes \(p. 193\)](#)

Hardware Specifications

The following is a summary of the hardware specifications for General purpose instances.

Instance type	Default vCPUs	Memory (GiB)
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
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
m5.4xlarge	16	64
m5.12xlarge	48	192
m5.24xlarge	96	384
m5d.large	2	8
m5d.xlarge	4	16
m5d.2xlarge	8	32
m5d.4xlarge	16	64
m5d.12xlarge	48	192
m5d.24xlarge	96	384

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 \(p. 232\)](#).

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. M4, M5, and M5d instances are EBS-optimized by default at no additional cost. For more information, see [Amazon EBS-Optimized Instances \(p. 812\)](#).

The `m4.10xlarge` and `m4.16xlarge` instance types provide the ability to control processor C-states and P-states on Linux. The `m5.12xlarge`, `m5.24xlarge`, `m5d.12xlarge`, and `m5d.24xlarge` instance types provide the ability to control processor C-states. 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 \(p. 434\)](#).

Network Performance

You can enable enhanced networking capabilities on supported instance types. Enhanced networking provides significantly higher packet-per-second (PPS) performance, lower network jitter, and lower latencies. For more information, see [Enhanced Networking on Linux \(p. 708\)](#).

Instance types that use the Elastic Network Adapter (ENA) for enhanced networking deliver high packet per second performance with consistently low latencies. Most applications do not consistently need a high level of network performance, but can benefit from having access to increased bandwidth when they send or receive data. Instance types that use the ENA and support up to 10 Gbps of throughput use a network I/O credit mechanism to allocate network bandwidth to instances based on average bandwidth utilization. These instances accrue credits when their network throughput is below their baseline limits, and can use these credits when they perform network data transfers. For workloads that require access to 10 Gbps of bandwidth or more on a sustained basis, we recommend using instance types that support 10 Gbps or 25 Gbps network speeds.

The following is a summary of network performance for General purpose instances that support enhanced networking.

Instance type	Network performance	Enhanced networking
<code>m4.large</code>	Moderate	Intel 82599 VF (p. 709)
<code>m4.xlarge</code> , <code>m4.2xlarge</code> , <code>m4.4xlarge</code>	High	Intel 82599 VF (p. 709)
<code>m4.10xlarge</code>	10 Gbps	Intel 82599 VF (p. 709)
<code>m4.16xlarge</code>	25 Gbps	ENA (p. 716)
<code>m5.large</code> , <code>m5.xlarge</code> , <code>m5.2xlarge</code> , <code>m5.4xlarge</code> , <code>m5d.large</code> , <code>m5d.xlarge</code> , <code>m5d.2xlarge</code> , <code>m5d.4xlarge</code>	Up to 10 Gbps	ENA (p. 716)
<code>m5.12xlarge</code> , <code>m5d.12xlarge</code>	10 Gbps	ENA (p. 716)
<code>m5.24xlarge</code> , <code>m5d.24xlarge</code>	25 Gbps	ENA (p. 716)

Instance Features

The following is a summary of features for General purpose instances:

	VPC only	EBS only	Placement group
M4	Yes	Yes	Yes
M5	Yes	Yes	Yes
M5d	Yes		Yes
T2	Yes	Yes	

For more information, see the following:

- [Instance Types Available Only in a VPC \(p. 636\)](#)
- [Amazon EBS–Optimized Instances \(p. 812\)](#)
- [Amazon EC2 Instance Store \(p. 847\)](#)
- [Placement Groups \(p. 701\)](#)
- [Enhanced Networking on Linux \(p. 708\)](#)

Release Notes

- M4, M5, M5d, and t2.large and larger T2 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.
- M5 and M5d instances have the following requirements:
 - Must have the NVMe drivers installed. EBS volumes are exposed as [NVMe block devices \(p. 823\)](#).
 - Must have the Elastic Network Adapter ([ENA \(p. 716\)](#)) drivers installed.

The following AMIs meet these requirements:

- Amazon Linux 2
- Amazon Linux AMI 2014.03 or later
- Ubuntu 14.04 or later
- SUSE Linux Enterprise Server 12 or later
- Red Hat Enterprise Linux 7.4 or later
- CentOS 7 or later
- FreeBSD 11.1-RELEASE
- Windows Server 2008 R2 or later
- M5 and M5d instances support a maximum of 27 EBS volumes plus elastic network interface attachments. For example, m5.2xlarge instances support four network interfaces. Every instance has at least one network interface. If you have a m5.2xlarge instance with three additional elastic network interface attachments, you can attach 24 EBS volumes to that instance.
- M5 and M5d instances should have acpid installed to support clean shutdown through API requests.
- [ClassicLink \(p. 638\)](#) is not supported for M5 and M5d instances—you cannot use ClassicLink to link your EC2-Classic instances to these instances in your VPC.
- 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?](#). To request a limit increase, use the [Amazon EC2 Instance Request Form](#).

Compute Optimized Instances

Compute optimized instances are ideal for compute-bound applications that benefit from high-performance processors. They are well suited for the following applications:

- 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

Contents

- [Hardware Specifications \(p. 194\)](#)
- [Instance Performance \(p. 195\)](#)
- [Network Performance \(p. 195\)](#)
- [Instance Features \(p. 196\)](#)
- [Release Notes \(p. 196\)](#)

Hardware Specifications

The following is a summary of the hardware specifications for compute optimized instances.

Instance type	Default vCPUs	Memory (GiB)
c4.large	2	3.75
c4.xlarge	4	7.5
c4.2xlarge	8	15
c4.4xlarge	16	30
c4.8xlarge	36	60
c5.large	2	4
c5.xlarge	4	8
c5.2xlarge	8	16
c5.4xlarge	16	32
c5.9xlarge	36	72
c5.18xlarge	72	144
c5d.large	2	4
c5d.xlarge	4	8
c5d.2xlarge	8	16

Instance type	Default vCPUs	Memory (GiB)
c5d.4xlarge	16	32
c5d.9xlarge	36	72
c5d.18xlarge	72	144

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 \(p. 232\)](#).

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. C4, C5, and C5d instances are EBS-optimized by default at no additional cost. For more information, see [Amazon EBS-Optimized Instances \(p. 812\)](#).

The c4.8xlarge instance type provides the ability to control processor C-states and P-states on Linux. The c5.9xlarge, c5.18xlarge, c5d.9xlarge, and c5d.18xlarge instance types provide the ability to control processor C-states. 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 \(p. 434\)](#).

Network Performance

You can enable enhanced networking capabilities on supported instance types. Enhanced networking provides significantly higher packet-per-second (PPS) performance, lower network jitter, and lower latencies. For more information, see [Enhanced Networking on Linux \(p. 708\)](#).

Instance types that use the Elastic Network Adapter (ENA) for enhanced networking deliver high packet per second performance with consistently low latencies. Most applications do not consistently need a high level of network performance, but can benefit from having access to increased bandwidth when they send or receive data. Instance types that use the ENA and support up to 10 Gbps of throughput use a network I/O credit mechanism to allocate network bandwidth to instances based on average bandwidth utilization. These instances accrue credits when their network throughput is below their baseline limits, and can use these credits when they perform network data transfers. For workloads that require access to 10 Gbps of bandwidth or more on a sustained basis, we recommend using instance types that support 10 Gbps or 25 Gbps network speeds.

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	Up to 10 Gbps	ENA (p. 716)
c5.9xlarge c5d.9xlarge	10 Gbps	ENA (p. 716)
c5.18xlarge c5d.18xlarge	25 Gbps	ENA (p. 716)
c4.large	Moderate	Intel 82599 VF (p. 709)

Instance type	Network performance	Enhanced networking
c4.xlarge c4.2xlarge c4.4xlarge	High	Intel 82599 VF (p. 709)
c4.8xlarge	10 Gbps	Intel 82599 VF (p. 709)

Instance Features

The following is a summary of features for compute optimized instances:

	VPC only	EBS only	Placement group
C4	Yes	Yes	Yes
C5	Yes	Yes	Yes
C5d	Yes		Yes

For more information, see the following:

- [Instance Types Available Only in a VPC \(p. 636\)](#)
- [Amazon EBS–Optimized Instances \(p. 812\)](#)
- [Amazon EC2 Instance Store \(p. 847\)](#)
- [Placement Groups \(p. 701\)](#)
- [Enhanced Networking on Linux \(p. 708\)](#)

Release Notes

- C4, C5, and C5d instances require 64-bit EBS-backed HVM AMIs. They have high-memory (up to 144 GiB of RAM), 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.
- C5 and C5d instances have the following requirements:
 - Must have the NVMe drivers installed. EBS volumes are exposed as [NVMe block devices \(p. 823\)](#).
 - Must have the Elastic Network Adapter ([ENA \(p. 716\)](#)) drivers installed.

The following AMIs meet these requirements:

- Amazon Linux 2
- Amazon Linux AMI 2014.03 or later
- Ubuntu 14.04 or later
- SUSE Linux Enterprise Server 12 or later
- Red Hat Enterprise Linux 7.4 or later
- CentOS 7 or later
- FreeBSD 11.1-RELEASE
- Windows Server 2008 R2 or later
- C5 instances support a maximum of 27 EBS volumes plus elastic network interface attachments. For example, c5.2xlarge instances support four network interfaces. Every instance has at least one network interface. If you have a c5.2xlarge instance with three additional elastic network interface attachments, you can attach 24 EBS volumes to that instance.

- The `c5d.18xlarge` instance type supports a maximum of 25 EBS volumes plus elastic network interface attachments. Other sizes of C5d instances support a maximum of 26 EBS volumes plus elastic network interfaces.
- C5 and C5d instances should have acpid installed to support clean shutdown through API requests.
- [ClassicLink \(p. 638\)](#) is not supported for C5 and C5d instances—you cannot use ClassicLink to link your EC2-Classic instances to these instances in your VPC.
- 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?](#). To request a limit increase, use the [Amazon EC2 Instance Request Form](#).

Memory Optimized Instances

Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory.

R4, R5, and R5d Instances

These instances are well suited for the following applications:

- 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.

X1 Instances

These instances are well suited for the following applications:

- 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.

X1e Instances

These instances are well suited for the following applications:

- High-performance databases.
- In-memory databases such as SAP HANA. For more information, see [SAP HANA on the AWS Cloud](#).
- Memory-intensive enterprise applications.

z1d Instances

These instances deliver both high compute and high memory and are well-suited for the following applications:

- Electronic Design Automation (EDA)

- Relational database workloads

Contents

- [Hardware Specifications \(p. 198\)](#)
- [Memory Performance \(p. 199\)](#)
- [Instance Performance \(p. 199\)](#)
- [Network Performance \(p. 200\)](#)
- [Instance Features \(p. 200\)](#)
- [Support for vCPUs \(p. 201\)](#)
- [Release Notes \(p. 201\)](#)

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
r4.16xlarge	64	488
r5.large	2	16
r5.xlarge	4	32
r5.2xlarge	8	64
r5.4xlarge	16	128
r5.12xlarge	48	384
r5.24xlarge	96	768
r5d.large	2	16
r5d.xlarge	4	32
r5d.2xlarge	8	64
r5d.4xlarge	16	128
r5d.12xlarge	48	384
r5d.24xlarge	96	768
x1.16xlarge	64	976
x1.32xlarge	128	1,952

Instance type	Default vCPUs	Memory (GiB)
x1e.xlarge	4	122
x1e.2xlarge	8	244
x1e.4xlarge	16	488
x1e.8xlarge	32	976
x1e.16xlarge	64	1,952
x1e.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

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 \(p. 232\)](#).

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 \(p. 198\)](#).

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 high-memory instance types. For more information, see [Linux AMI Virtualization Types \(p. 86\)](#).

Instance Performance

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.

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.

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 \(p. 434\)](#).

Network Performance

You can enable enhanced networking capabilities on supported instance types. Enhanced networking provides significantly higher packet-per-second (PPS) performance, lower network jitter, and lower latencies. For more information, see [Enhanced Networking on Linux \(p. 708\)](#).

Instance types that use the Elastic Network Adapter (ENA) for enhanced networking deliver high packet per second performance with consistently low latencies. Most applications do not consistently need a high level of network performance, but can benefit from having access to increased bandwidth when they send or receive data. Instance types that use the ENA and support up to 10 Gbps of throughput use a network I/O credit mechanism to allocate network bandwidth to instances based on average bandwidth utilization. These instances accrue credits when their network throughput is below their baseline limits, and can use these credits when they perform network data transfers. For workloads that require access to 10 Gbps of bandwidth or more on a sustained basis, we recommend using instance types that support 10 Gbps or 25 Gbps network speeds.

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 r5d.4xlarge and smaller x1e.8large and smaller z1d.3xlarge and smaller	Up to 10 Gbps	ENA (p. 716)
r4.8xlarge r5.12xlarge r5d.12xlarge x1.16xlarge x1e.16xlarge z1d.6xlarge	10 Gbps	ENA (p. 716)
r4.16xlarge r5.24xlarge r5d.24xlarge x1.32xlarge x1e.32xlarge z1d.12xlarge	25 Gbps	ENA (p. 716)

Instance Features

The following is a summary of features for memory optimized instances.

	VPC only	EBS only	Instance store	Placement group
R4	Yes	Yes		Yes
R5	Yes	Yes		Yes
R5d	Yes		NVME *	Yes
X1	Yes		SSD	Yes
X1e	Yes		SSD	Yes
z1d	Yes		NVME *	Yes

* The root device volume must be an Amazon EBS volume.

For more information, see the following:

- [Instance Types Available Only in a VPC \(p. 636\)](#)
- [Amazon EBS-Optimized Instances \(p. 812\)](#)
- [Amazon EC2 Instance Store \(p. 847\)](#)
- [Placement Groups \(p. 701\)](#)
- [Enhanced Networking on Linux \(p. 708\)](#)

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 2016
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2 64-bit
- Windows Server 2008 SP2 64-bit

Release Notes

- The following are requirements for R5, R5d, and z1d instances:
 - NVMe drivers must be installed. EBS volumes are exposed as [NVMe block devices \(p. 823\)](#).
 - Elastic Network Adapter ([ENA \(p. 716\)](#)) drivers must be installed.

The following AMIs meet these requirements:

- Amazon Linux 2
- Amazon Linux 2014.03 or later
- Ubuntu 14.04 or later
- SUSE Linux Enterprise Server 12 or later
- Red Hat Enterprise Linux 7.4 or later
- CentOS 7 or later
- Windows Server 2008 R2 or later
- 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.
- [ClassicLink \(p. 638\)](#) is not supported for R5, R5d, and z1d instances—you cannot use ClassicLink to link your EC2-Classic instances to these instances in your VPC.

- 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?](#). To request a limit increase, use the [Amazon EC2 Instance Request Form](#).

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

D2 instances are well suited for the following applications:

- Massive parallel processing (MPP) data warehouse
- MapReduce and Hadoop distributed computing
- Log or data processing applications

H1 Instances

H1 instances are well suited for the following applications:

- 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 Instances

I3 instances are well suited for the following applications:

- High frequency online transaction processing (OLTP) systems
- Relational databases
- NoSQL databases
- Cache for in-memory databases (for example, Redis)
- Data warehousing applications
- Low latency Ad-Tech serving applications

i3.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

Contents

- [Hardware Specifications \(p. 203\)](#)
- [Instance Performance \(p. 203\)](#)
- [Network Performance \(p. 204\)](#)
- [SSD I/O Performance \(p. 204\)](#)

- [Instance Features \(p. 205\)](#)
- [Support for vCPUs \(p. 206\)](#)
- [Release Notes \(p. 207\)](#)

Hardware Specifications

The primary data storage for D2 instances is HDD instance store volumes. The primary data storage for I3 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 \(p. 847\)](#) and [SSD Instance Store Volumes \(p. 854\)](#).

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

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 \(p. 232\)](#).

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. D2 and H1 instances are EBS-optimized by default at no additional cost. For more information, see [Amazon EBS-Optimized Instances \(p. 812\)](#).

The h1.16xlarge, h1.8xlarge, d2.8xlarge, i3.16xlarge, and i3.metal 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 \(p. 434\)](#).

Network Performance

You can enable enhanced networking capabilities on supported instance types. Enhanced networking provides significantly higher packet-per-second (PPS) performance, lower network jitter, and lower latencies. For more information, see [Enhanced Networking on Linux \(p. 708\)](#).

Instance types that use the Elastic Network Adapter (ENA) for enhanced networking deliver high packet per second performance with consistently low latencies. Most applications do not consistently need a high level of network performance, but can benefit from having access to increased bandwidth when they send or receive data. Instance types that use the ENA and support up to 10 Gbps of throughput use a network I/O credit mechanism to allocate network bandwidth to instances based on average bandwidth utilization. These instances accrue credits when their network throughput is below their baseline limits, and can use these credits when they perform network data transfers. For workloads that require access to 10 Gbps of bandwidth or more on a sustained basis, we recommend using instance types that support 10 Gbps or 25 Gbps network speeds.

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, use network I/O credit mechanism	ENA (p. 716)
i3.8xlarge h1.8xlarge	10 Gbps	ENA (p. 716)
i3.16xlarge i3.metal h1.16xlarge	25 Gbps	ENA (p. 716)
d2.xlarge	Moderate	Intel 82599 VF (p. 709)
d2.2xlarge d2.4xlarge	High	Intel 82599 VF (p. 709)
d2.8xlarge	10 Gbps	Intel 82599 VF (p. 709)

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

* For i3.large and i3.xlarge 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 \(p. 855\)](#).

Instance Features

The following is a summary of features for storage optimized instances:

	VPC only	SSD volumes	Placement group	Enhanced networking
D2			Yes	Intel 82599 VF (p. 709)
H1	Yes		Yes	ENAv (p. 716)
I3	Yes	NVMe	Yes	ENAv (p. 716)

For more information, see the following:

- [Instance Types Available Only in a VPC \(p. 636\)](#)

- [Amazon EBS–Optimized Instances \(p. 812\)](#)
- [Amazon EC2 Instance Store \(p. 847\)](#)
- [Placement Groups \(p. 701\)](#)
- [Enhanced Networking on Linux \(p. 708\)](#)

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 \(p. 86\)](#).
- You must launch i3 instances using an Amazon EBS-backed AMI.
- The following are requirements for i3.metal instances:
 - NVMe drivers must be installed. EBS volumes are exposed as [NVMe block devices \(p. 823\)](#).
 - Elastic Network Adapter ([ENA \(p. 716\)](#)) drivers must be installed.

The following AMIs meet these requirements:

- Amazon Linux 2
- Amazon Linux 2014.03 or later
- Ubuntu 14.04 or later
- SUSE Linux Enterprise Server 12 or later
- Red Hat Enterprise Linux 7.4 or later
- CentOS 7 or later
- Windows Server 2008 R2 or later
- Launching an i3.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 an i3.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
```

- i3.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. i3.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, i3.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 \(p. 206\)](#).
- [ClassicLink \(p. 638\)](#) is not supported for `i3.metal` instances—you cannot use ClassicLink to link your EC2-Classic instances to these instances in your VPC.
- 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?](#). To request a limit increase, use the [Amazon EC2 Instance Request Form](#).

Linux Accelerated Computing Instances

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) or Field Programmable Gate Arrays (FPGAs). Accelerated computing instances enable more parallelism for higher throughput on compute-intensive workloads.

GPU-based instances provide access to NVIDIA GPUs with thousands of compute cores. You can use GPU-based accelerated computing 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.

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](#).

Important

FPGA-based instances do not support Microsoft Windows.

You can cluster accelerated computing instances into a cluster placement group. Cluster placement groups provide low latency and high-bandwidth connectivity between the instances within a single Availability Zone. For more information, see [Placement Groups \(p. 701\)](#).

Contents

- [Accelerated Computing Instance Families \(p. 209\)](#)
- [Hardware Specifications \(p. 210\)](#)
- [Instance Performance \(p. 210\)](#)
- [Network Performance \(p. 211\)](#)
- [Instance Features \(p. 211\)](#)
- [Release Notes \(p. 212\)](#)
- [AMIs for GPU-Based Accelerated Computing Instances \(p. 212\)](#)
- [Installing the NVIDIA Driver on Linux Instances \(p. 212\)](#)
- [Activate NVIDIA GRID Virtual Applications \(G3 Instances Only\) \(p. 216\)](#)
- [Optimizing GPU Settings \(P2, P3, and G3 Instances\) \(p. 216\)](#)
- [Getting Started with FPGA Development \(p. 217\)](#)

For information about Windows accelerated computing instances, see [Windows Accelerated Computing Instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Accelerated Computing Instance Families

Accelerated computing instance families 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. The following accelerated computing instance families are available for you to launch in Amazon EC2.

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](#).

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 16 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.

P3 instances support NVIDIA NVLink peer to peer transfers.

To view topology information about the system, run the following command:

```
nvidia-smi topo -m
```

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.

To view topology information about the system, run the following command:

```
nvidia-smi topo -m
```

For more information, see [NVIDIA GPUDirect](#).

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.

G3 instances support NVIDIA GRID Virtual Workstation and NVIDIA GRID Virtual Applications. To activate either of these features, see [Activate NVIDIA GRID Virtual Applications \(G3 Instances Only\) \(p. 216\)](#).

Hardware Specifications

The following is a summary of the hardware specifications for accelerated computing instances.

Instance type	Default vCPUs	Memory (GiB)
p2.xlarge	4	61
p2.8xlarge	32	488
p2.16xlarge	64	732
p3.2xlarge	8	61
p3.8xlarge	32	244
p3.16xlarge	64	488
g3.4xlarge	16	122
g3.8xlarge	32	244
g3.16xlarge	64	488
f1.2xlarge	8	122
f1.16xlarge	64	976

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 \(p. 232\)](#).

Instance Performance

There are several GPU setting optimizations that you can perform to achieve the best performance on P2, P3, and G3 instances. For more information, see [Optimizing GPU Settings \(P2, P3, and G3 Instances\) \(p. 216\)](#).

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. F1, P3, P2, and G3 instances are EBS-optimized by default at no additional cost. For more information, see [Amazon EBS-Optimized Instances \(p. 812\)](#).

The `g3.16xlarge`, `p2.16xlarge`, `p3.16xlarge`, and `f1.16xlarge` 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 \(p. 434\)](#).

Network Performance

You can enable enhanced networking capabilities on supported instance types. Enhanced networking provides significantly higher packet-per-second (PPS) performance, lower network jitter, and lower latencies. For more information, see [Enhanced Networking on Linux \(p. 708\)](#).

Instance types that use the Elastic Network Adapter (ENA) for enhanced networking deliver high packet per second performance with consistently low latencies. Most applications do not consistently need a high level of network performance, but can benefit from having access to increased bandwidth when they send or receive data. Instance types that use the ENA and support up to 10 Gbps of throughput use a network I/O credit mechanism to allocate network bandwidth to instances based on average bandwidth utilization. These instances accrue credits when their network throughput is below their baseline limits, and can use these credits when they perform network data transfers. For workloads that require access to 10 Gbps of bandwidth or more on a sustained basis, we recommend using instance types that support 10 Gbps or 25 Gbps network speeds.

The following is a summary of network performance for accelerated computing instances that support enhanced networking.

Instance type	Network performance	Enhanced networking
<code>f1.2xlarge</code> <code>g3.4xlarge</code> <code>p3.2xlarge</code>	Up to 10 Gbps	ENA (p. 716)
<code>g3.8xlarge</code> <code>p2.8xlarge</code> <code>p3.8xlarge</code>	10 Gbps	ENA (p. 716)
<code>f1.16xlarge</code> <code>g3.16.xlarge</code> <code>g3.16.xlarge</code> <code>p2.16xlarge</code> <code>p3.16xlarge</code>	25 Gbps	ENA (p. 716)

Instance Features

The following is a summary of features for accelerated computing instances.

	VPC only	EBS only	Instance store	Placement group
G3	Yes	Yes		Yes
P2	Yes	Yes		Yes
P3	Yes	Yes		Yes
F1	Yes		NVMe *	Yes

* The root device volume must be an Amazon EBS volume.

For more information, see the following:

- [Instance Types Available Only in a VPC \(p. 636\)](#)

- [Amazon EBS–Optimized Instances \(p. 812\)](#)
- [Amazon EC2 Instance Store \(p. 847\)](#)
- [Placement Groups \(p. 701\)](#)
- [Enhanced Networking on Linux \(p. 708\)](#)

Release Notes

- You must launch the instance using an HVM AMI.
- GPU-based instances can't access the GPU unless the NVIDIA drivers are installed.
- There is a limit of 100 AFIs per region.
- There is a limit on the number of instances that you can run. For more information, see [How many instances can I run in Amazon EC2?](#) in the Amazon EC2 FAQ. To request an increase in these limits, use the following form: [Request to Increase Amazon EC2 Instance Limit](#).

AMIs for GPU-Based Accelerated Computing Instances

To help you get started, NVIDIA and others provide AMIs for GPU-based accelerated computing instances. These reference AMIs include the NVIDIA driver, which enables full functionality and performance of the NVIDIA GPUs.

For a list of AMIs with the NVIDIA driver, search AWS Marketplace as follows:

- [NVIDIA P3 AMIs](#)
- [NVIDIA P2 AMIs](#)
- [NVIDIA GRID G3 AMIs](#)
- [NVIDIA GRID G2 AMIs](#)

You can launch accelerated computing instances using any HVM AMI.

Important

These AMIs include drivers, software, or toolkits that are developed, owned, or provided by NVIDIA Corporation. By using these AMIs, you agree to use these NVIDIA drivers, software, or toolkits only on Amazon EC2 instances that include NVIDIA hardware.

You can also install the NVIDIA driver manually. For more information, see [Installing the NVIDIA Driver on Linux Instances \(p. 212\)](#).

Installing the NVIDIA Driver on Linux Instances

A GPU-based accelerated computing instance must have the appropriate NVIDIA driver. The NVIDIA driver that you install must be compiled against the kernel that you plan to run on your instance.

Amazon provides AMIs with updated and compatible builds of the NVIDIA kernel drivers for each official kernel upgrade in the AWS Marketplace. If you decide to use a different NVIDIA driver version than the one that Amazon provides, or decide to use a kernel that's not an official Amazon build, you must uninstall the Amazon-provided NVIDIA packages from your system to avoid conflicts with the versions of the drivers that you are trying to install.

Use this command to uninstall Amazon-provided NVIDIA packages:

```
sudo yum erase nvidia cuda
```

The Amazon-provided 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.

Downloading the NVIDIA GRID Driver (G3)

For G3 instances, you can download the NVIDIA GRID driver from Amazon S3 using the AWS CLI or SDKs. To install the AWS CLI, see [Installing the AWS Command Line Interface](#) in the *AWS Command Line Interface User Guide*.

Important

This download is available to AWS customers only. By downloading, you agree to use the downloaded software only to develop AMIs for use with the NVIDIA Tesla M60 hardware. Upon installation of the software, you are bound by the terms of the [NVIDIA GRID Cloud End User License Agreement](#).

Use the following AWS CLI command to download the latest driver:

```
aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

Multiple versions of the NVIDIA GRID driver are stored in this bucket. You can see all of the available versions with the following command:

```
aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

If you receive an `Unable to locate credentials` error, the AWS CLI on the instance is not configured to use your AWS credentials. To configure the AWS CLI to use your AWS credentials, see [Quick Configuration](#) in the *AWS Command Line Interface User Guide*.

Downloading a Public NVIDIA Driver (G2, P2, P3)

For instance types other than G3, or if you are not using NVIDIA GRID functionality on a G3 instance, you can download the public NVIDIA drivers.

Download the 64-bit NVIDIA driver appropriate for your instance type from <http://www.nvidia.com/Download/Find.aspx>.

Instances	Product Type	Product Series	Product
G2	GRID	GRID Series	GRID K520
P2	Tesla	K-Series	K-80
P3	Tesla	V-Series	V100

For more information about installing and configuring the driver, choose the **ADDITIONAL INFORMATION** tab on the download page for the driver on the NVIDIA website and choose the README link.

Installing the NVIDIA Driver Manually

To install the driver on a Linux instance

1. Update your package cache and get necessary package updates for your instance.

- For Amazon Linux, CentOS, and Red Hat Enterprise Linux:

```
sudo yum update -y
```

- For Ubuntu and Debian:

```
sudo apt-get update -y
```

2. (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
```

3. Reboot your instance to load the latest kernel version.

```
sudo reboot
```

4. Reconnect to your instance after it has rebooted.
5. 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:

```
sudo yum install -y gcc kernel-devel-$(uname -r)
```

- For Ubuntu and Debian:

```
sudo apt-get install -y gcc make linux-headers-$(uname -r)
```

6. (Graphical desktop instances only) 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.

```
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.

- CentOS and Red Hat Enterprise Linux:

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

- For Ubuntu and Debian:

```
sudo update-grub
```

7. Download the driver package that you identified earlier as follows.
 - For P2 and P3 instances, the following command downloads the NVIDIA driver, where xxx.xxx represents the version of the NVIDIA driver.

```
wget http://us.download.nvidia.com/tesla/xxx.xxx/NVIDIA-Linux-x86_64-xxx.xxx.run
```
 - For G2 instances, the following command downloads the NVIDIA driver, where xxx.xxx represents the version of the NVIDIA driver.

```
wget http://us.download.nvidia.com/XFree86/Linux-x86_64/xxx.xxx/NVIDIA-Linux-x86_64-xxx.xxx.run
```
 - For G3 instances, you can download the driver from Amazon S3 using the AWS CLI or SDKs. To install the AWS CLI, see [Installing the AWS Command Line Interface](#) in the *AWS Command Line Interface User Guide*. Use the following AWS CLI command to download the latest driver:

```
aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

Important

This download is available to AWS customers only. By downloading, you agree to use the downloaded software only to develop AMIs for use with the NVIDIA Tesla M60 hardware. Upon installation of the software, you are bound by the terms of the [NVIDIA GRID Cloud End User License Agreement](#).

Multiple versions of the NVIDIA GRID driver are stored in this bucket. You can see all of the available versions with the following command:

```
aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

8. Run the self-install script to install the NVIDIA driver that you downloaded in the previous step. For example:

```
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).

9. Reboot the instance.

```
sudo reboot
```

10. Confirm that the driver is functional. The response for the following command lists the installed NVIDIA driver version and details about the GPUs.

Note

This command may take several minutes to run.

```
nvidia-smi -q | head
```

11. [G3 instances only] To enable NVIDIA GRID Virtual Applications on a G3 instance, complete the GRID activation steps in [Activate NVIDIA GRID Virtual Applications \(G3 Instances Only\) \(p. 216\)](#) (NVIDIA GRID Virtual Workstation is enabled by default).
12. [P2, P3, and G3 instances] Complete the optimization steps in [Optimizing GPU Settings \(P2, P3, and G3 Instances\) \(p. 216\)](#) to achieve the best performance from your GPU.

Activate NVIDIA GRID Virtual Applications (G3 Instances Only)

To activate the GRID Virtual Applications on G3 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 G3 Linux instances

1. Create the /etc/nvidia/gridd.conf file from the provided template file.

```
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.

```
sudo reboot
```

Optimizing GPU Settings (P2, P3, and G3 Instances)

There are several GPU setting optimizations that you can perform to achieve the best performance on P2, P3, and G3 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.

```
sudo nvidia-persistenced
```

2. Disable the autoboot feature for all GPUs on the instance.

```
sudo nvidia-smi --auto-boost-default=0
```

Note

GPUs on P3 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.

- P2 instances:

```
sudo nvidia-smi -ac 2505,875
```

- P3 instances:

```
sudo nvidia-smi -ac 877,1530
```

- G3 instances:

```
sudo nvidia-smi -ac 2505,1177
```

Getting Started with FPGA Development

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](#).

T1 Micro Instances

T1 Micro instances (`t1.micro`) provide a small amount of consistent CPU resources and allow you to increase CPU capacity in short bursts when additional cycles are available. They are well suited for lower throughput applications and websites that require additional compute cycles periodically.

Note

The `t1.micro` is a previous generation instance and it has been replaced by the `t2.micro`, which has a much better performance profile. We recommend using the `t2.micro` instance type instead of the `t1.micro`. For more information, see [T2 Instances \(p. 168\)](#).

The `t1.micro` instance is available as an Amazon EBS-backed instance only.

This documentation describes how `t1.micro` instances work so that you can understand how to apply them. It's not our intent to specify exact behavior, but to give you visibility into the instance's behavior so you can understand its performance.

Topics

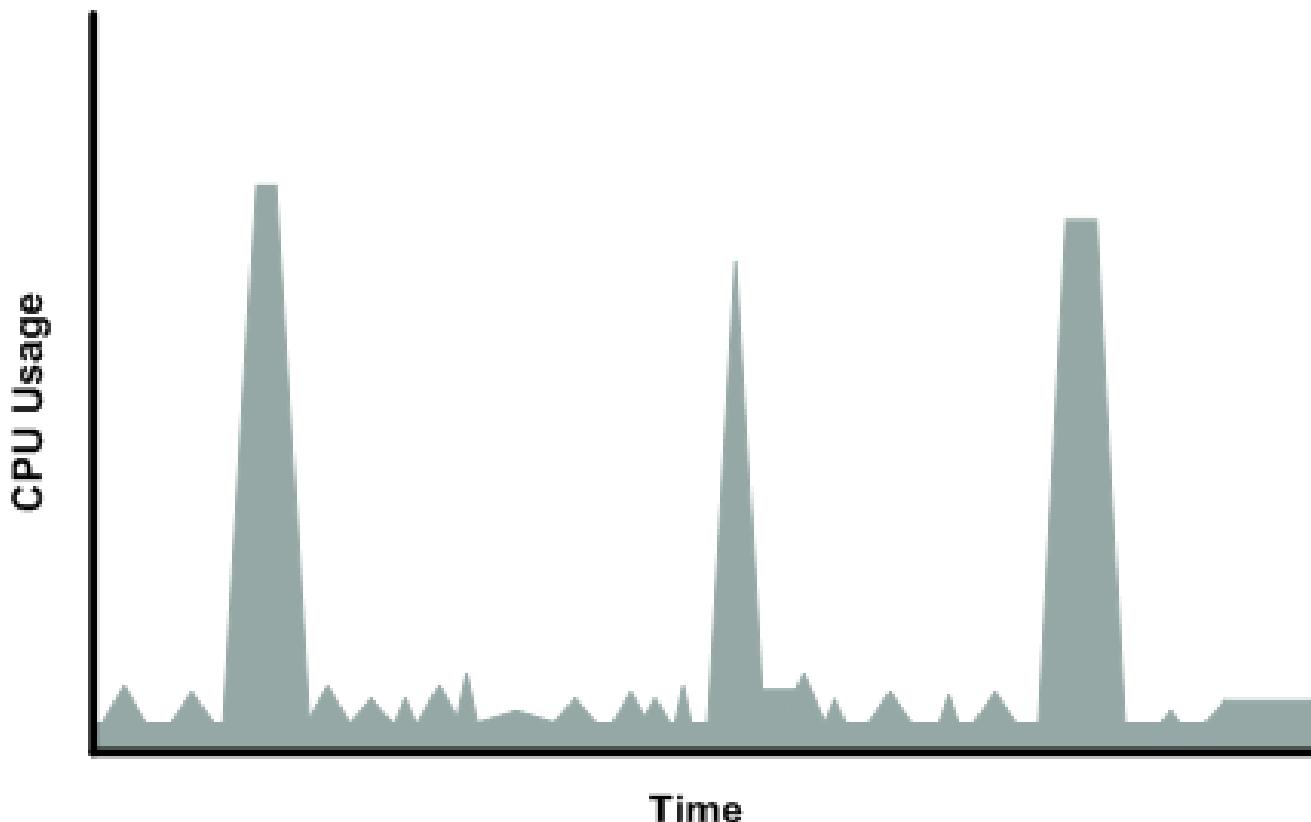
- [Hardware Specifications \(p. 217\)](#)
- [Optimal Application of T1 Micro Instances \(p. 217\)](#)
- [Available CPU Resources During Spikes \(p. 221\)](#)
- [When the Instance Uses Its Allotted Resources \(p. 221\)](#)
- [Comparison with the m1.small Instance Type \(p. 224\)](#)
- [AMI Optimization for Micro Instances \(p. 227\)](#)

Hardware Specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see [Amazon EC2 Instance Types](#).

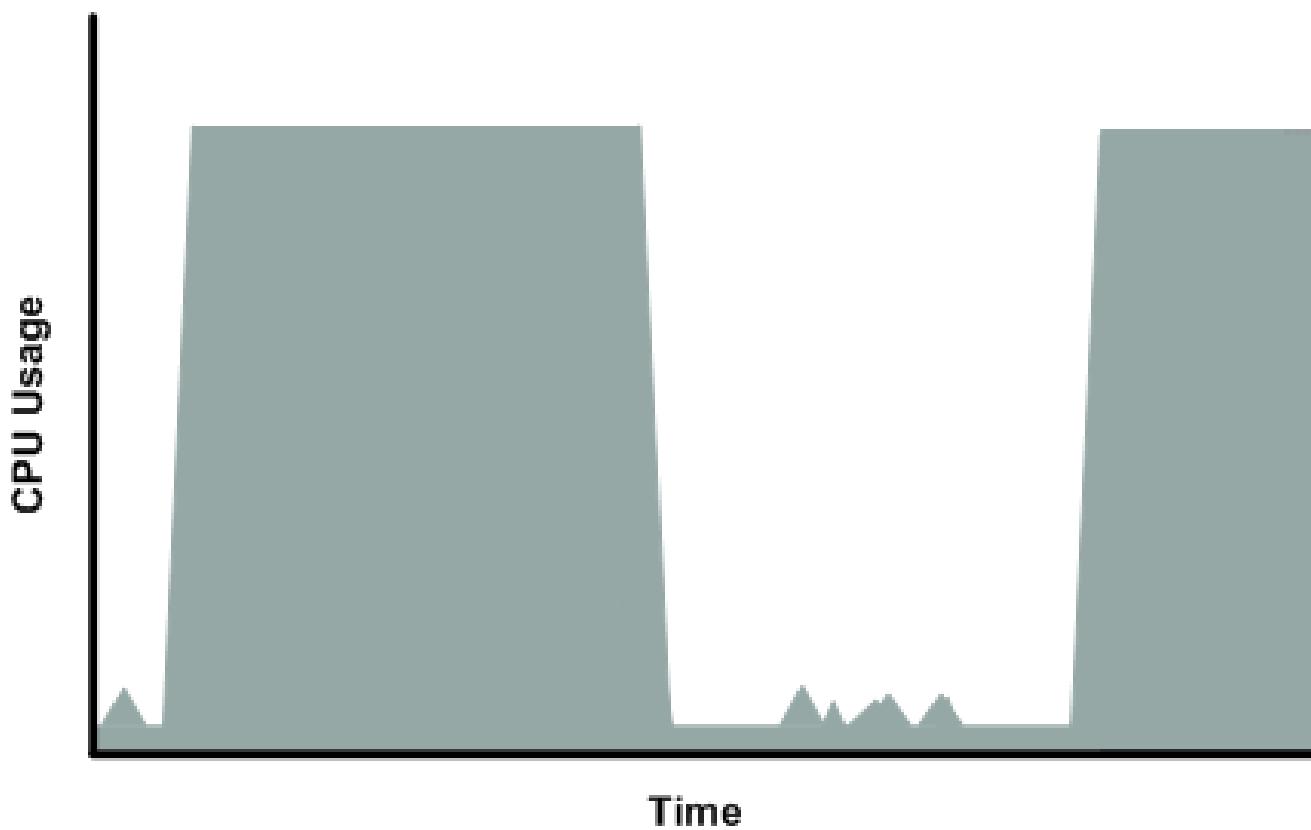
Optimal Application of T1 Micro Instances

A `t1.micro` instance provides spiky CPU resources for workloads that have a CPU usage profile similar to what is shown in the following figure.

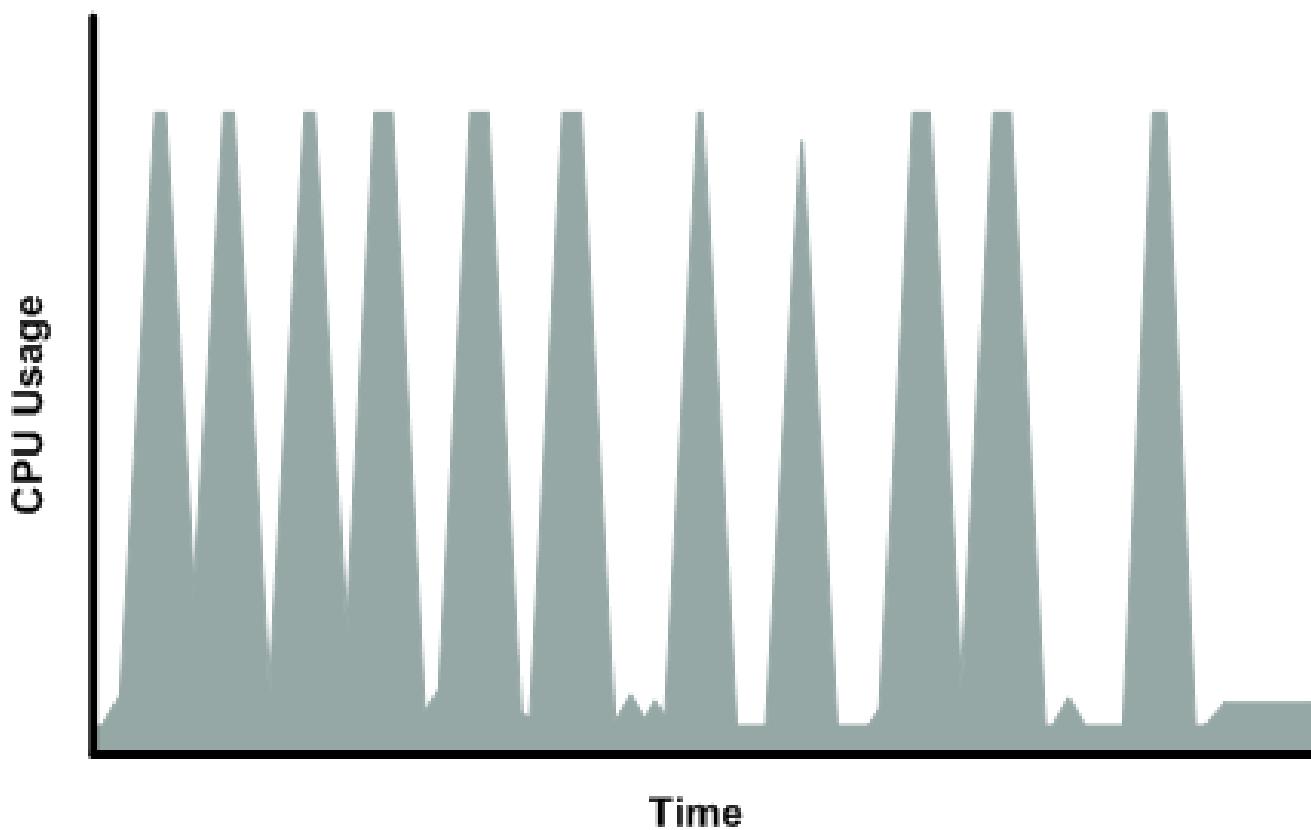


The instance is designed to operate with its CPU usage at essentially only two levels: the normal low background level, and then at brief spiked levels much higher than the background level. We allow the instance to operate at up to 2 EC2 compute units (ECUs) (one ECU provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor). The ratio between the maximum level and the background level is designed to be large. We designed `t1.micro` instances to support tens of requests per minute on your application. However, actual performance can vary significantly depending on the amount of CPU resources required for each request on your application.

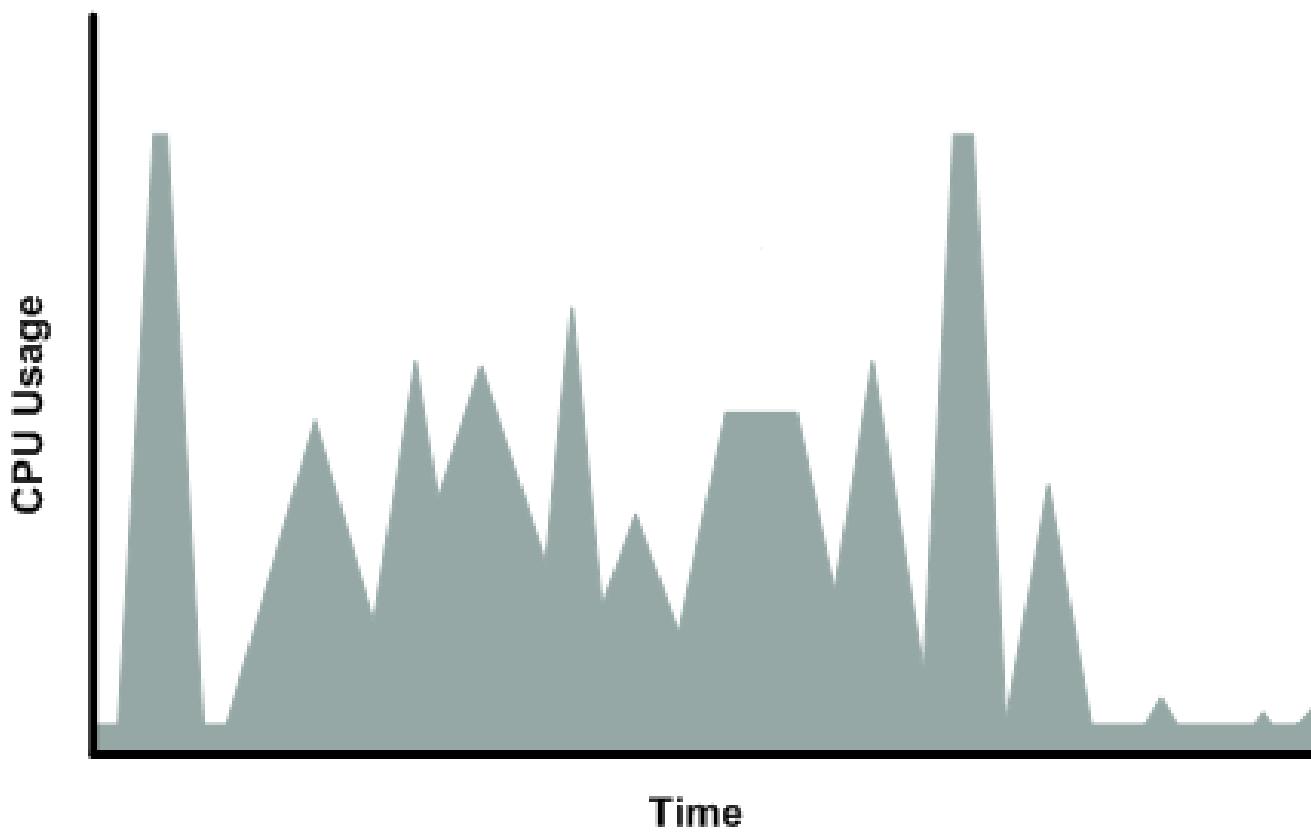
Your application might have a different CPU usage profile than that described in the preceding section. The following figure shows the profile for an application that isn't appropriate for a `t1.micro` instance. The application requires continuous data-crunching CPU resources for each request, resulting in plateaus of CPU usage that the `t1.micro` instance isn't designed to handle.



The following figure shows another profile that isn't appropriate for a `t1.micro` instance. Here the spikes in CPU use are brief, but they occur too frequently to be serviced by a micro instance.



The following figure shows another profile that isn't appropriate for a `t1.micro` instance. Here the spikes aren't too frequent, but the background level between spikes is too high to be serviced by a `t1.micro` instance.



In each of the preceding cases of workloads not appropriate for a `t1.micro` instance, we recommend that you consider using a different instance type. For more information about instance types, see [Instance Types \(p. 163\)](#).

Available CPU Resources During Spikes

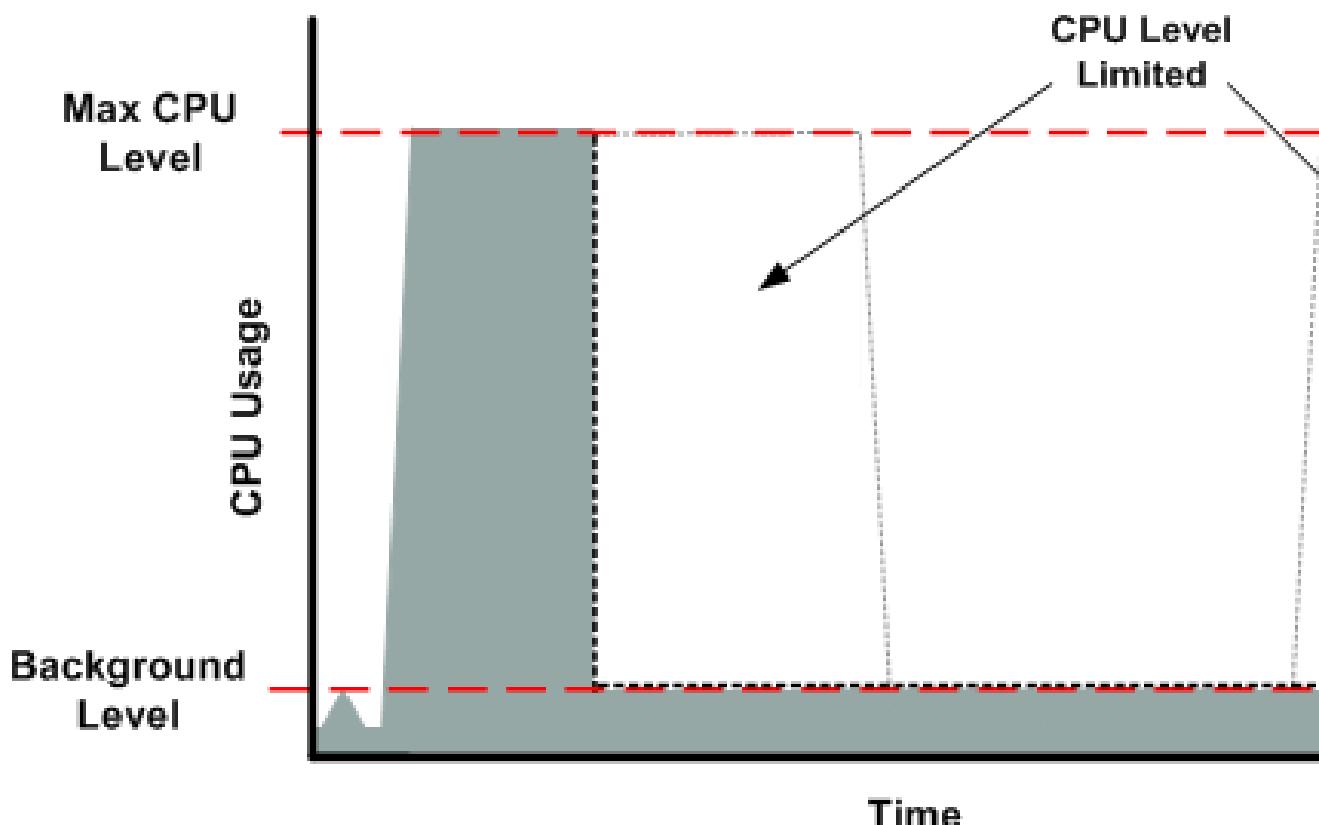
When your instance *bursts* to accommodate a spike in demand for compute resources, it uses unused resources on the host. The amount available depends on how much contention there is when the spike occurs. The instance is never left with zero CPU resources, whether other instances on the host are spiking or not.

When the Instance Uses Its Allotted Resources

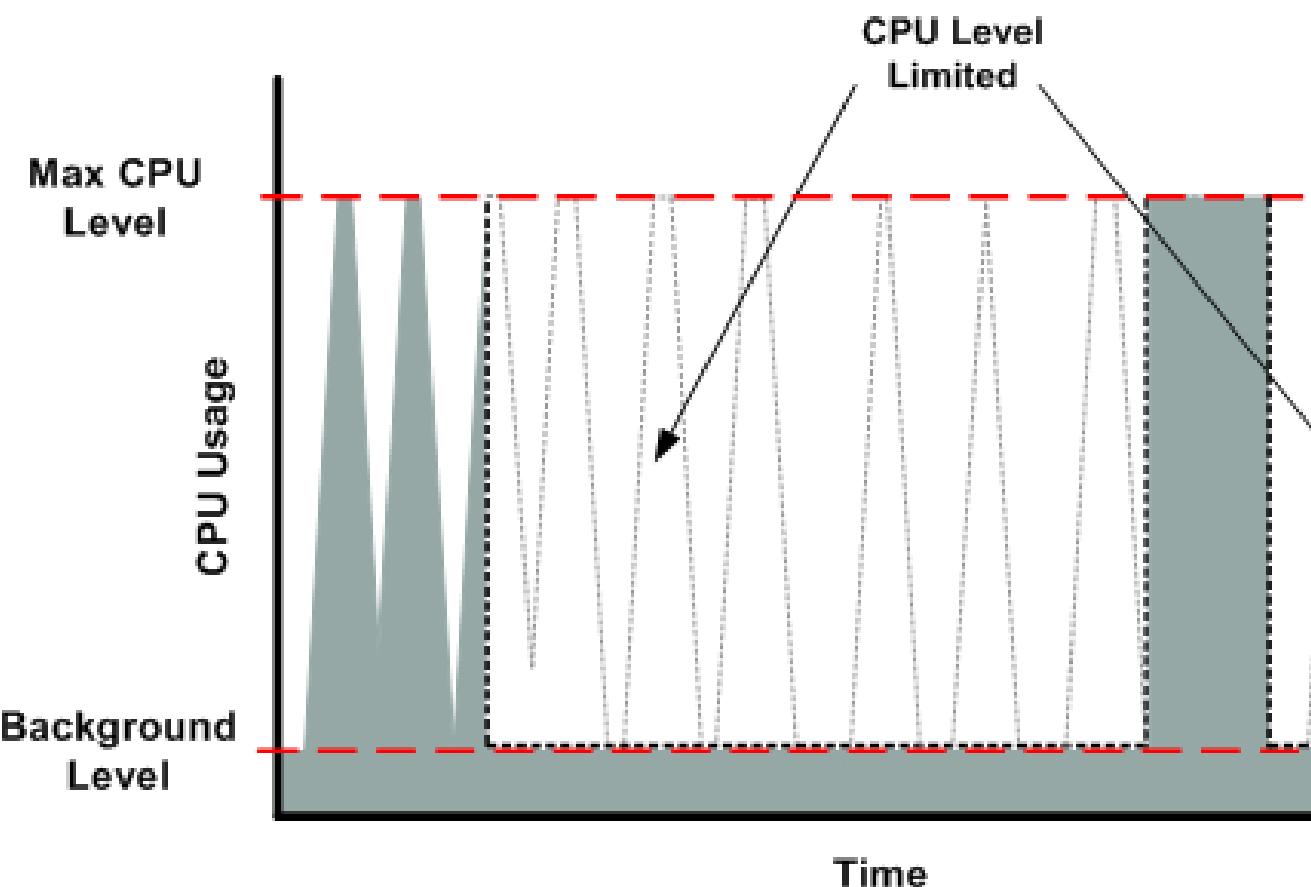
We expect your application to consume only a certain amount of CPU resources in a period of time. If the application consumes more than your instance's allotted CPU resources, we temporarily limit the instance so it operates at a low CPU level. If your instance continues to use all of its allotted resources, its performance will degrade. We will increase the time that we limit its CPU level, thus increasing the time before the instance is allowed to burst again.

If you enable CloudWatch monitoring for your `t1.micro` instance, you can use the "Avg CPU Utilization" graph in the AWS Management Console to determine whether your instance is regularly using all its allotted CPU resources. We recommend that you look at the maximum value reached during each given period. If the maximum value is 100%, we recommend that you use Amazon EC2 Auto Scaling to scale out (with additional `t1.micro` instances and a load balancer), or move to a larger instance type. For more information, see the [Amazon EC2 Auto Scaling User Guide](#).

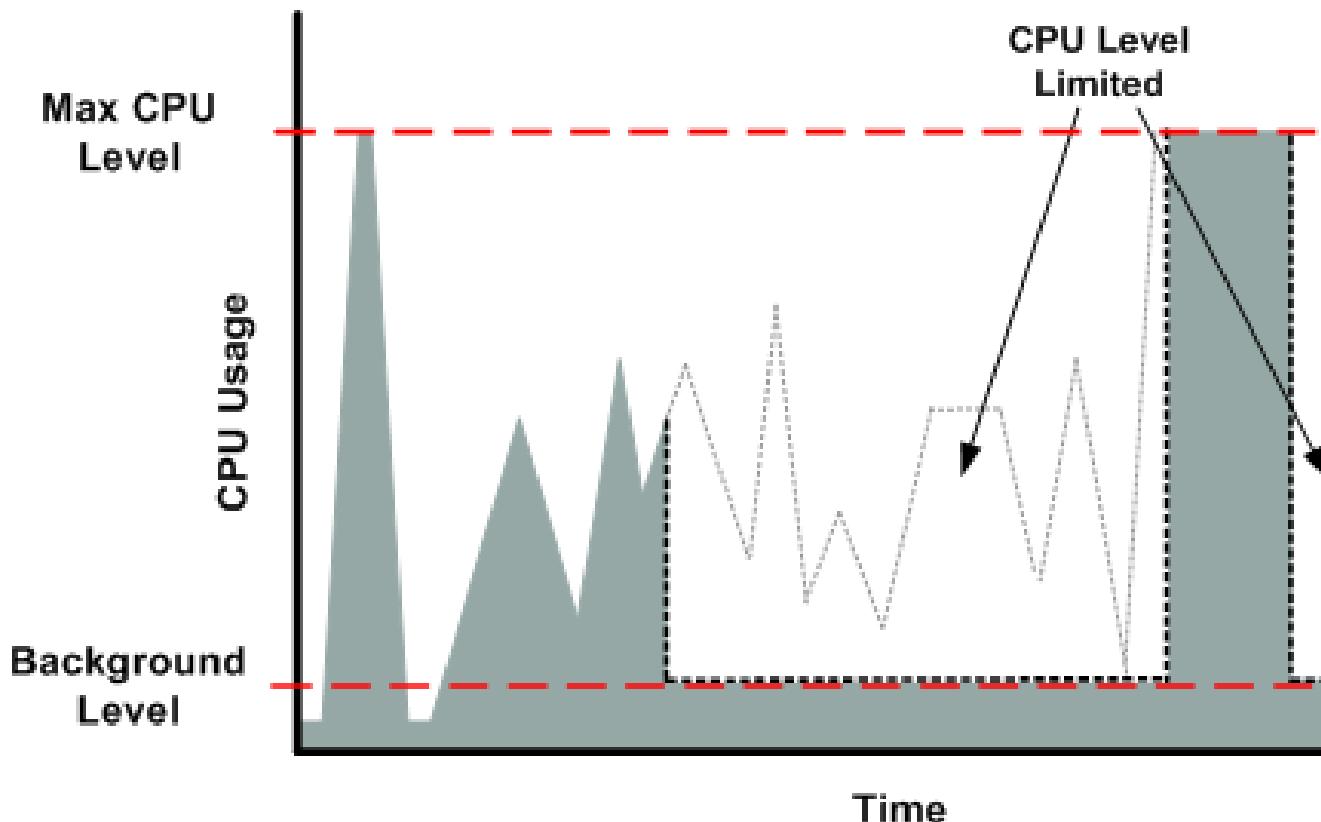
Consider the three suboptimal profiles from the preceding section and what it might look like when the instance consumes its allotted resources and we limit its CPU level. If an instance consumes its allotted resources, we restrict it to the low background level. The following figure shows long plateaus of data-crunching CPU usage. The CPU hits the maximum allowed level and stays there until the instance's allotted resources are consumed for the period. At that point, we limit the instance to operate at the low background level, and it operates there until we allow it to burst above that level again. The instance again stays there until the allotted resources are consumed and we limit it again (not seen on the graph).



The following figure shows requests that are too frequent. The instance uses its allotted resources after only a few requests and so we limit it. After we lift the restriction, the instance maxes out its CPU usage trying to keep up with the requests, and we limit it again.

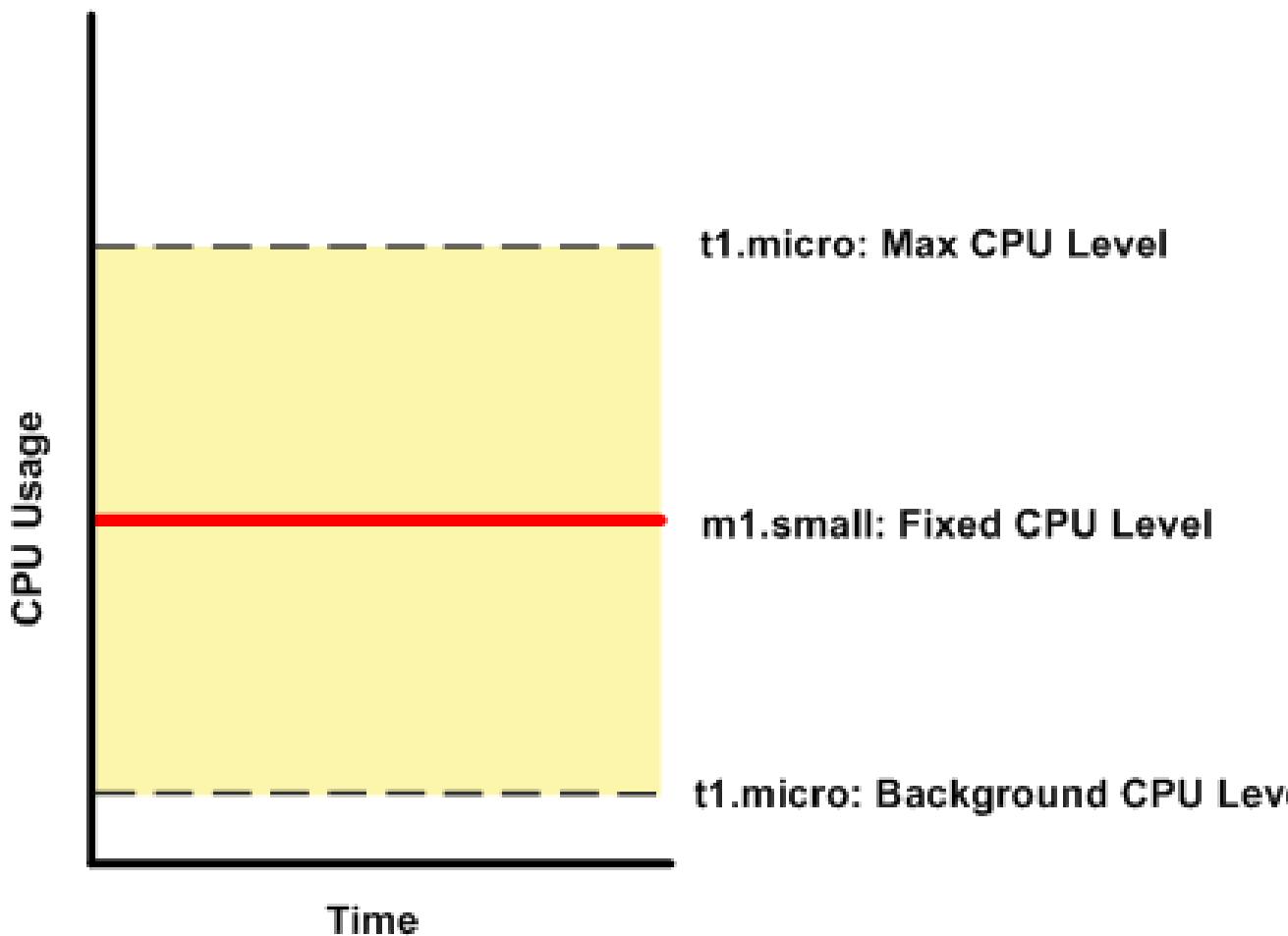


The following figure shows a background level that is too high. Notice that the instance doesn't have to be operating at the maximum CPU level for us to limit it. We limit the instance when it's operating above the normal background level and has consumed its allotted resources for the given period. In this case (as in the preceding one), the instance can't keep up with the work, and we limit it again.

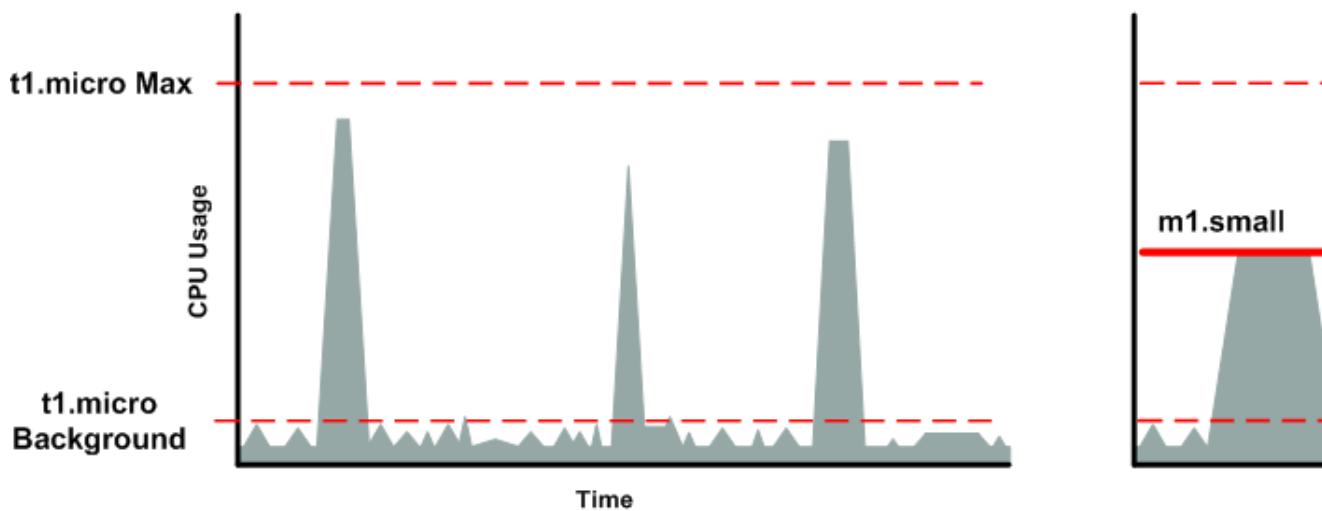


Comparison with the m1.small Instance Type

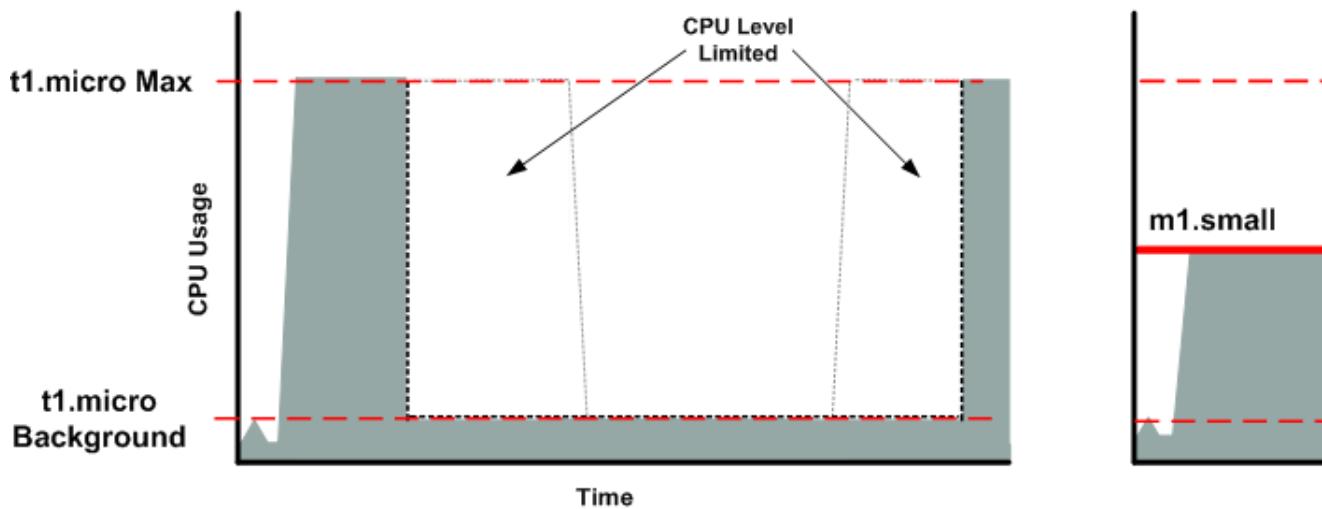
The `t1.micro` instance provides different levels of CPU resources at different times (up to 2 ECUs). By comparison, the `m1.small` instance type provides 1 ECU at all times. The following figure illustrates the difference.



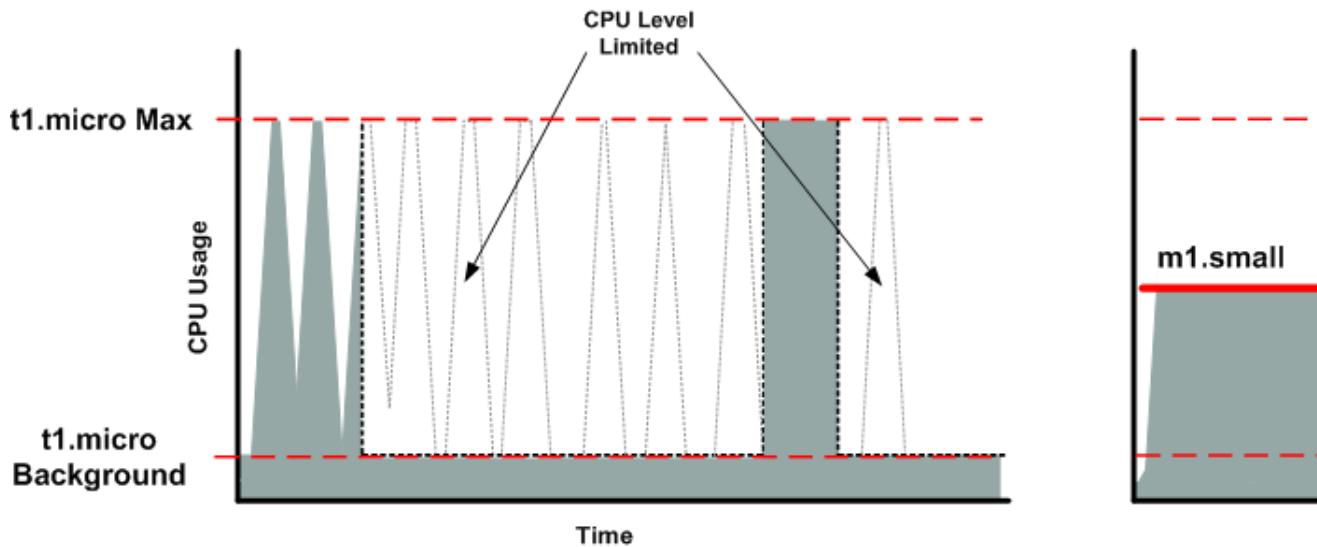
Let's compare the CPU usage of a `t1.micro` instance with an `m1.small` instance for the various scenarios we've discussed in the preceding sections. The following figure that follows shows an optimal scenario for a `t1.micro` instance (the left graph) and how it might look for an `m1.small` instance (the right graph). In this case, we don't need to limit the `t1.micro` instance. The processing time on the `m1.small` instance would be longer for each spike in CPU demand compared to the `t1.micro` instance.



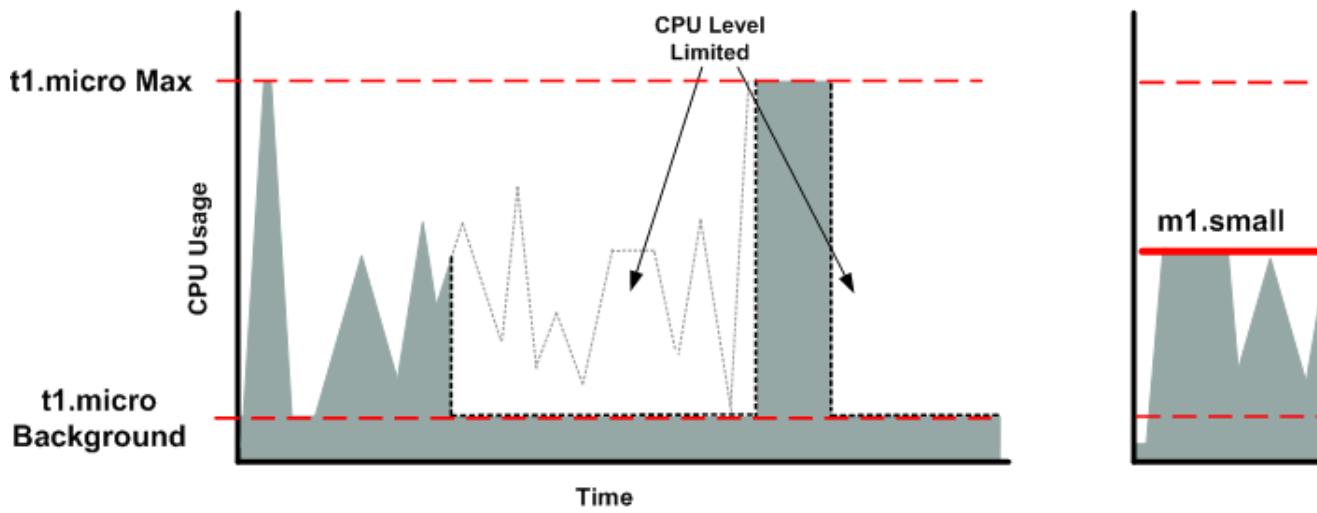
The following figure shows the scenario with the data-crunching requests that used up the allotted resources on the **t1.micro** instance, and how they might look with the **m1.small** instance.



The following figure shows the frequent requests that used up the allotted resources on the **t1.micro** instance, and how they might look on the **m1.small** instance.



The following figure shows the situation where the background level used up the allotted resources on the **t1.micro** instance, and how it might look on the **m1.small** instance.



AMI Optimization for Micro Instances

We recommend that you follow these best practices when optimizing an AMI for the **t1.micro** instance type:

- Design the AMI to run on 600 MB of RAM
- Limit the number of recurring processes that use CPU time (for example, cron jobs, daemons)

You can optimize performance using swap space and virtual memory (for example, by setting up swap space in a separate partition from the root file system).

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 an `m3.medium` instance.

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 \(p. 84\)](#).

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. If you want to add instance store volumes, you must migrate your application to a new instance with the instance type and instance store volumes that you need. An exception to this rule is when you resize to a storage-optimized instance type that by default contains a higher number of volumes. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 847\)](#).

Contents

- [Compatibility for Resizing Instances \(p. 228\)](#)
- [Resizing an Amazon EBS-backed Instance \(p. 229\)](#)
- [Migrating an Instance Store-backed Instance \(p. 230\)](#)
- [Migrating to a New Instance Configuration \(p. 231\)](#)

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:

Note

- **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 \(p. 86\)](#). 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.
- **Network:** Some instance types are not supported in EC2-Classic and must be launched in a VPC. Therefore, you can't resize an instance in EC2-Classic to a instance type that is available only in a VPC unless you have a nondefault VPC. For more information, see [Instance Types Available Only in a VPC \(p. 636\)](#). To check if your instance is in a VPC, check the **VPC ID** value on the details pane of the **Instances** screen in the Amazon EC2 console.
- **Platform:** All Amazon EC2 instance types support 64-bit AMIs, but only the following instance types 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. To check the platform of your instance, go to the **Instances** screen in the Amazon EC2 console and choose **Show/Hide Columns, Architecture**.

- **Enhanced networking:** Instance types that support [enhanced networking \(p. 708\)](#) require the necessary drivers installed. For example, the C5, C5d, M5, M5d, R5, R5d, and z1d instance types require EBS-backed AMIs with the Elastic Network Adapter (ENA) drivers installed. To resize an existing instance to an instance type that supports enhanced networking, you must first install the [ENA drivers \(p. 716\)](#) or [ixgbevf drivers \(p. 709\)](#) on your instance, as appropriate.
- **NVMe:** Instance types that support NVMe, such as C5, C5d, M5, M5d, R5, R5d, and z1d, expose EBS volumes as NVMe block devices. To resize an existing instance to an instance type that supports NVMe, you must first install the [NVMe drivers \(p. 823\)](#) on your instance. Also, the device names for NVMe devices are `/dev/nvme[0-26]n1`. For more information about the required AMIs for these instance types, see the Release Notes in [Compute Optimized Instances \(p. 194\)](#) and [General Purpose Instances \(p. 190\)](#).

For example, T2 instances are not supported in EC2-Classic and they are HVM only. On Linux, T1 instances do not support HVM and must be launched from PV AMIs. Therefore, you can't resize a T1 Linux instance to a T2 Linux instance.

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 is running in a VPC and 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 running in EC2-Classic, we give it new public and private IP addresses, and disassociate any Elastic IP address that's associated with the instance. Therefore, to ensure that your users can continue to use the applications that you're hosting on your instance uninterrupted, you must re-associate any Elastic IP address after you restart your instance.
- 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*.
- 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 \(p. 415\)](#).

Use the following procedure to resize an Amazon EBS-backed instance using the AWS Management 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 \(p. 228\)](#).
2. Open the Amazon EC2 console.
3. In the navigation pane, choose **Instances**, and select the instance.
4. [EC2-Classic] If the instance has an associated Elastic IP address, write down the Elastic IP address and the instance ID shown in the details pane.
5. Choose **Actions**, select **Instance State**, and then choose **Stop**.
6. In the confirmation dialog box, choose **Yes, Stop**. It can take a few minutes for the instance to stop.

[EC2-Classic] When the instance state becomes stopped, the **Elastic IP**, **Public DNS (IPv4)**, **Private DNS**, and **Private IPs** fields in the details pane are blank to indicate that the old values are no longer associated with the instance.

7. With the instance still selected, choose **Actions**, select **Instance Settings**, and then choose **Change Instance Type**. Note that this action is disabled if the instance state is not stopped.
8. 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).
 - 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. Note that 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.
9. To restart the stopped instance, select the instance, choose **Actions**, select **Instance State**, and then choose **Start**.
10. In the confirmation dialog box, choose **Yes, Start**. It can take a few minutes for the instance to enter the `running` state.
11. [EC2-Classic] When the instance state is `running`, the **Public DNS (IPv4)**, **Private DNS**, and **Private IPs** fields in the details pane contain the new values that we assigned to the instance. If your instance had an associated Elastic IP address, you must reassociate it as follows:
 - a. In the navigation pane, choose **Elastic IPs**.
 - b. Select the Elastic IP address that you wrote down before you stopped the instance.
 - c. Choose **Actions** and then choose **Associate address**.
 - d. From **Instance**, select the instance ID that you wrote down before you stopped the instance, and then choose **Associate**.

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.

To migrate an instance store-backed instance

1. [EC2-Classic] If the instance you are migrating has an associated Elastic IP address, record the Elastic IP address now so that you can associate it with the new instance later.
2. 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 an Amazon EBS Snapshot \(p. 795\)](#)) 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 an Instance \(p. 767\)](#)).
3. 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 \(p. 105\)](#). When you are finished creating an AMI from your instance, return to this procedure.
4. Open the Amazon EC2 console and in the navigation pane, select **AMIs**. From the filter lists, select **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.

5. 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.

Note that it can take a few minutes for the instance to enter the `running` state.
6. [EC2-Classic] If the instance that you started with had an associated Elastic IP address, you must associate it with the new instance as follows:
 - a. In the navigation pane, choose **Elastic IPs**.
 - b. Select the Elastic IP address that you recorded at the beginning of this procedure.
 - c. Choose **Actions** and then choose **Associate Address**.
 - d. From **Instance**, select the new instance, and then choose **Associate**.
7. (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**, select **Instance State**, and then choose **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:

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 an Amazon EBS Snapshot \(p. 795\)](#)) 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 an Instance \(p. 767\)](#)).
2. Launch a new instance, selecting the following:
 - An HVM AMI.
 - The HVM only instance type.
 - [EC2-VPC] 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, choose **Actions**, and then choose **Disassociate address**. When prompted for confirmation, choose **Disassociate address**.
 - c. With the Elastic IP address still selected, choose **Actions**, and then choose **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**, select **Instance State**, and then choose **Terminate**.

For information about migrating an application from an instance in EC2-Classic to an instance in a VPC, see [Migrating from a Linux Instance in EC2-Classic to a Linux Instance in a VPC \(p. 648\)](#).

Optimizing CPU Options

Amazon EC2 instances support Intel Hyper-Threading Technology, which enables multiple threads to run concurrently on a single Intel Xeon 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 hyperthread of an Intel Xeon CPU core, except for T2 instances.

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 Intel Hyper-Threading Technology 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 \(p. 232\)](#)
- [CPU Cores and Threads Per CPU Core Per Instance Type \(p. 233\)](#)
- [Specifying CPU Options for Your Instance \(p. 237\)](#)
- [Viewing the CPU Options for Your Instance \(p. 237\)](#)

Rules for Specifying CPU Options

To specify the CPU options for your instance, be aware of the following rules:

- CPU options are currently supported using the AWS CLI, an AWS SDK, or the Amazon EC2 API only.
- 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 examples, see [Specifying CPU Options for Your Instance \(p. 237\)](#).

- The total 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 \(p. 233\)](#).
- To disable Intel Hyper-Threading Technology, specify one thread per core.
- If you [change the instance type \(p. 228\)](#) of an existing instance, the CPU options automatically change to the default CPU options for the new instance type.
- The CPU options that you specify 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
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
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
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
f1.2xlarge	8	4	2	1, 2, 3, 4	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
f1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	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
c5.18xlarge	72	36	2	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
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
m5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5.24xlarge	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

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
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,	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
				12, 13, 14, 15, 16	
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

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
d2.xlarge	4	2	2	1, 2	1, 2
d2.2xlarge	8	4	2	1, 2, 3, 4	1, 2
d2.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
d2.8xlarge	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
h1.2xlarge	8	4	2	1, 2, 3, 4	1, 2
h1.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
h1.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
h1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
i3.large	2	1	2	1	1, 2
i3.xlarge	4	2	2	1, 2	1, 2
i3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
i3.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
i3.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
				12, 13, 14, 15, 16	
i3.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	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 \(p. 235\)](#):

- 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

To disable Intel Hyper-Threading Technology during instance launch

- Use the [run-instances](#) AWS CLI command and specify a value of 1 for `ThreadsPerCore` for the `--cpu-options` parameter. For `CoreCount`, specify the default CPU core count for the instance type (in this example, 8 for an `r4.4xlarge` instance).

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options "CoreCount=8,ThreadsPerCore=1" --key-name MyKeyValuePair
```

To specify a custom number of vCPUs during instance launch

The following example launches an `r4.4xlarge` instance with six vCPUs.

1. 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 MyKeyValuePair
```

2. Alternatively, specify six CPU cores and one thread per core (disable hyperthreading) to get six vCPUs:

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options "CoreCount=6,ThreadsPerCore=1" --key-name MyKeyValuePair
```

Viewing the CPU Options for Your Instance

You can view the CPU options for an existing instance by describing the instance.

To view the CPU options for an instance using the command line

- Use the [describe-instances](#) AWS CLI 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*.

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.
- **Reserved Instances** – Purchase, at a significant discount, instances that are always available, for a term from one to three 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 lower 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.

If you require a capacity reservation, purchase Reserved Instances 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 can help you address compliance requirements and reduce costs by using your existing server-bound software licenses. For more information, see [Amazon EC2 Instance Purchasing Options](#).

Contents

- [Determining the Instance Lifecycle \(p. 239\)](#)
- [Reserved Instances \(p. 240\)](#)
- [Scheduled Reserved Instances \(p. 274\)](#)
- [Spot Instances \(p. 278\)](#)
- [Dedicated Hosts \(p. 332\)](#)
- [Dedicated Instances \(p. 345\)](#)

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.

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 \(p. 898\)](#).

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:

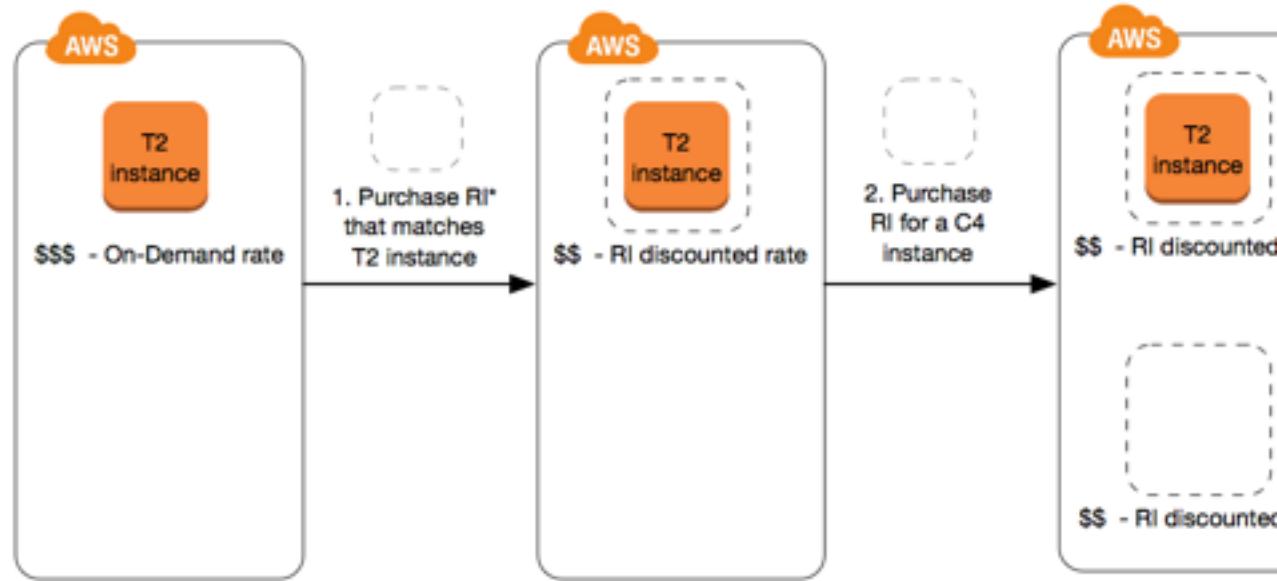
```
"InstanceLifecycle": "scheduled"
```

Otherwise, the output does not contain `InstanceLifecycle`.

Reserved Instances

Reserved Instances provide you with a significant discount 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 in order to benefit from the billing discount.

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.

When you purchase a Reserved Instance, choose a combination of the following that suits your needs:

- **Payment option:** No Upfront, Partial Upfront, or All Upfront.
- **Term:** One-year or three-year. A year is defined as 31536000 seconds (365 days). Three years is defined as 94608000 seconds (1095 days).
- **Offering class:** Convertible or Standard.

In addition, a Reserved Instance has a number of attributes that determine how it is applied to a running instance in your account:

- **Instance type:** For example, m4.large. This is composed of the instance family (m4) and the instance size (large).
- **Scope:** Whether the Reserved Instance applies to a region or specific Availability Zone.
- **Tenancy:** Whether your instance runs on shared (default) or single-tenant (dedicated) hardware. For more information, see [Dedicated Instances \(p. 345\)](#).
- **Platform:** The operating system; for example, Windows or Linux/Unix. For more information, see [Choosing a Platform \(p. 251\)](#).

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.

After you purchase a Reserved Instance, you cannot cancel your purchase. However, you may be able to [modify \(p. 264\)](#), [exchange \(p. 270\)](#), or [sell \(p. 257\)](#) your Reserved Instance if your needs change.

Payment Options

The following payment options are available for Reserved Instances.

- **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.

- **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.
- **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.

Generally speaking, you can save more money choosing Reserved Instances with a higher upfront payment. 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 [Selling on the Reserved Instance Marketplace \(p. 257\)](#).

For more information about pricing, see [Amazon EC2 Reserved Instances Pricing](#).

Using Reserved Instances in a VPC

If your account supports EC2-Classic, you can purchase Reserved Instances for use in either EC2-Classic or a VPC. You can purchase Reserved Instances to apply to instances launched into a VPC by selecting a platform that includes *Amazon VPC* in its name.

If you have an EC2-VPC-only account, the listed platforms available do not include *Amazon VPC* in its name because all instances must be launched into a VPC.

For more information, see [Detecting Your Supported Platforms and Whether You Have a Default VPC](#).

Reserved Instance Limits

You are limited to purchasing 20 Reserved Instances per Availability Zone, per month, plus 20 regional Reserved Instances. Therefore, in a region that has three Availability Zones, you can purchase 80 Reserved Instances in total: 20 per Availability Zone (60) plus 20 regional Reserved Instances.

Reserved Instances that are purchased for a specific Availability Zone (zonal Reserved Instances) allow you to launch as many instances that are covered by the zonal Reserved Instances, even if it means that you exceed your On-Demand Instance limit. For example, your running On-Demand Instance limit is 20, and you are currently running 18 On-Demand Instances. You have five unused zonal Reserved Instances. You can launch two more On-Demand Instances with any specifications, and you can launch five instances that exactly match the specifications of your zonal Reserved Instances; giving you a total of 25 instances.

Regional Reserved Instances do not increase your On-Demand Instance limit.

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 family, 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 type cannot be modified. You cannot exchange a Standard Reserved Instance, only modify it. For more information, see Modifying Reserved Instances (p. 264) .	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. 270) . You can also modify some attributes of a Convertible Reserved Instance. For more information, see Modifying Reserved Instances (p. 264) .
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, or to instances in a region. When you purchase a Reserved Instance in a specific Availability Zone, it provides a capacity reservation. When you purchase a Reserved Instance for a region, it's referred to as a *regional Reserved Instance*. Regional Reserved Instances do not provide a capacity reservation.

Regional Reserved Instances have the following attributes:

- **Availability Zone flexibility:** the Reserved Instance discount applies to instance usage in any Availability Zone in a region.

- **Instance size flexibility:** the Reserved Instance discount applies to instance usage regardless of size, within that instance family. Only supported on Linux/Unix Reserved Instances with default tenancy.

For more information and examples, see [How Reserved Instances Are Applied \(p. 243\)](#).

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 \(p. 274\)](#).

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 \(p. 256\)](#).

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

Reserved Instances purchased for a region (regional Reserved Instances) provide Availability Zone flexibility—the Reserved Instance discount applies to instance usage in any Availability Zone in that region.

Regional Reserved Instances on the Linux/Unix platform with default tenancy also provide instance size flexibility, where the Reserved Instance discount applies to instance usage within that instance type, regardless of size.

Note

Instance size flexibility does not apply to Reserved Instances that are purchased for a specific Availability Zone, bare metal instances, Reserved Instances with dedicated tenancy, and Reserved Instances for Windows, Windows with SQL Standard, Windows with SQL Server Enterprise, Windows with SQL Server Web, RHEL, and SLES.

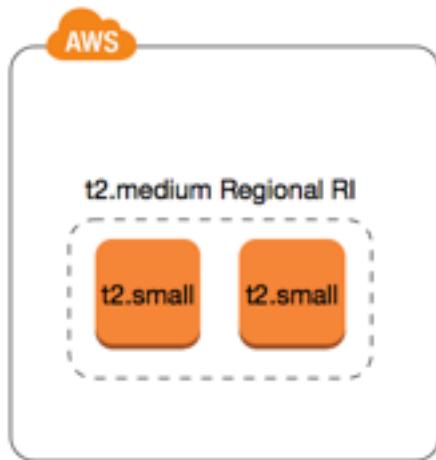
Instance size flexibility is determined by the normalization factor of the instance size. The discount applies either fully or partially to running instances of the same instance type, depending on the instance size of the reservation, in any Availability Zone in the region. The only attributes that must be matched are the instance type, 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.

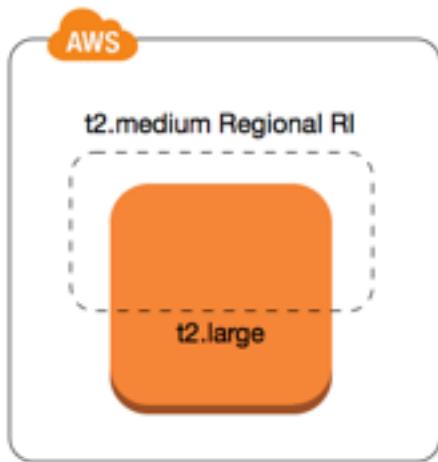
The table below describes the different sizes within an instance type, and corresponding normalization factor per hour. This scale is used to apply the discounted rate of Reserved Instances to the normalized usage of the instance type.

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

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.



Note

The normalization factor is also applied when modifying Reserved Instances. For more information, see [Modifying Reserved Instances \(p. 264\)](#).

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 . xlarge 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 \(p. 249\)](#). 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 in account A and the `m4.2xlarge` instance in account A. All three instances match the attributes (instance family, region, platform, tenancy). There is no capacity reservation.
- 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.

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 \(p. 241\)](#) specified for the Reserved Instance.

When Reserved Instances expire, you are charged On-Demand rates for EC2 instance usage. You can set up a billing alert to warn you when your bill exceeds a threshold you define. For more information, see [Monitoring Charges with Alerts and Notifications](#) in the *AWS Billing and Cost Management User Guide*.

Note

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](#).

Topics

- [Usage Billing \(p. 247\)](#)
- [Viewing Your Bill \(p. 248\)](#)
- [Reserved Instances and Consolidated Billing \(p. 249\)](#)
- [Reserved Instance Discount Pricing Tiers \(p. 249\)](#)

Usage Billing

Reserved Instances are billed for every clock-hour during the term that you select, regardless of whether an instance is running or not. A clock-hour is defined as the standard 24-hour clock that runs from

midnight to midnight, and is divided into 24 hours (for example, 1:00:00 to 1:59:59 is one clock-hour). For more information about instance states, see [Instance Lifecycle \(p. 350\)](#).

A Reserved Instance billing benefit is applied to a running instance on a per-second basis. 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 payer account, any member accounts that benefit from Reserved Instances billing discounts continue to benefit from the discount until the Reserved Instances expire, or until the member account is removed.

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 or Windows with SQL Server Web.
- 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 \(p. 249\)](#)
- [Buying with a Discount Tier \(p. 250\)](#)
- [Crossing Pricing Tiers \(p. 250\)](#)
- [Consolidated Billing for Pricing Tiers \(p. 251\)](#)

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) listed on the [Reserved Instances pricing page](#) 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 \(p. 251\)](#).

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 \(p. 250\)](#).

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 \(p. 249\)](#).

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](#).

Topics

- [Choosing a Platform \(p. 251\)](#)
- [Buying Standard Reserved Instances \(p. 251\)](#)
- [Buying Convertible Reserved Instances \(p. 254\)](#)
- [Viewing Your Reserved Instances \(p. 256\)](#)
- [Using Your Reserved Instances \(p. 256\)](#)

Choosing a Platform

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. For all other Linux distributions (including Ubuntu), choose an offering for the **Linux/UNIX** platform.

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 Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances, 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**.

Note

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. To complete the order, choose **Purchase**.

Note

If, at the time of purchase, there are offerings similar to your choice but with a lower price, AWS sells you the offerings at the lower price.

9. The status of your purchase 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
```

```
{
    "ReservedInstancesOfferings": [
        {
            "OfferingClass": "standard",
            "OfferingType": "No Upfront",
            "ProductDescription": "Linux/UNIX",
            "InstanceTenancy": "default",
            "PricingDetails": [],
            "UsagePrice": 0.0,
            "RecurringCharges": [
                {
                    "Amount": 0.0672,
                    "Frequency": "Hourly"
                }
            ],
            "Marketplace": false,
            "CurrencyCode": "USD",
            "FixedPrice": 0.0,
            "Duration": 31536000,
            "Scope": "Region",
            "ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2",
            "InstanceType": "t2.large"
        },
    ]
}
```

```
"OfferingClass": "standard",
"OfferingType": "Partial Upfront",
"ProductDescription": "Linux/UNIX",
"InstanceTenancy": "default",
"PricingDetails": [],
"UsagePrice": 0.0,
"RecurringCharges": [
    {
        "Amount": 0.032,
        "Frequency": "Hourly"
    }
],
"Marketplace": false,
"CurrencyCode": "USD",
"FixedPrice": 280.0,
"Duration": 31536000,
"Scope": "Region",
"ReservedInstancesOfferingId": "6b15a842-3acb-4320-bd55-fa43a79f3fe3",
"InstanceType": "t2.large"
},
{
    "OfferingClass": "standard",
    "OfferingType": "All Upfront",
    "ProductDescription": "Linux/UNIX",
    "InstanceTenancy": "default",
    "PricingDetails": [],
    "UsagePrice": 0.0,
    "RecurringCharges": [],
    "Marketplace": false,
    "CurrencyCode": "USD",
    "FixedPrice": 549.0,
    "Duration": 31536000,
    "Scope": "Region",
    "ReservedInstancesOfferingId": "5062dc97-d284-417b-b09e-8abed1e5a183",
    "InstanceType": "t2.large"
}
]
```

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 `ReservedInstancesOfferingId`.

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 ec06327e-dd07-46ee-9398-75b5fexample --instance-count 1
```

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 \(p. 256\)](#).

For examples of how Reserved Instances are applied to your running instances, see [How Reserved Instances Are Applied \(p. 243\)](#).

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 Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reserved Instances, 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. To complete the order, choose **Purchase**.

Note

If, at the time of purchase, there are offerings similar to your choice but with a lower price, AWS sells you the offerings at the lower price.

9. The status of your purchase 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
```

```
{
```

```
    "ReservedInstancesOfferings": [
```

```
{  
    "OfferingClass": "convertible",  
    "OfferingType": "No Upfront",  
    "ProductDescription": "Linux/UNIX",  
    "InstanceTenancy": "default",  
    "PricingDetails": [],  
    "UsagePrice": 0.0,  
    "RecurringCharges": [  
        {  
            "Amount": 0.0556,  
            "Frequency": "Hourly"  
        }  
    ],  
    "Marketplace": false,  
    "CurrencyCode": "USD",  
    "FixedPrice": 0.0,  
    "Duration": 94608000,  
    "Scope": "Region",  
    "ReservedInstancesOfferingId": "e242e87b-b75c-4079-8e87-02d53f145204",  
    "InstanceType": "t2.large"  
},  
{  
    "OfferingClass": "convertible",  
    "OfferingType": "Partial Upfront",  
    "ProductDescription": "Linux/UNIX",  

```

When you find a Reserved Instance that meets your needs, take note of the `ReservedInstancesOfferingId`.

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 ec06327e-  
dd07-46ee-9398-75b5fexample --instance-count 1
```

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 \(p. 256\)](#).

For examples of how Reserved Instances are applied to your running instances, see [How Reserved Instances Are Applied \(p. 243\)](#).

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 \(p. 257\)](#). For more information, see [Reserved Instance Listing States \(p. 262\)](#).

To view your Reserved Instances using the command line

- [describe-reserved-instances](#) (AWS CLI)
- [Get-EC2ReservedInstance](#) (Tools for Windows PowerShell)

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 \(p. 345\)](#).

For more information, see [Launching an Instance Using the Launch Instance Wizard \(p. 355\)](#). For examples of how Reserved Instances are applied to your running instances, see [How Reserved Instances Are Applied \(p. 243\)](#).

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](#).

Selling on the 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.

Topics

- [Selling in the Reserved Instance Marketplace \(p. 257\)](#)
- [Buying in the Reserved Instance Marketplace \(p. 263\)](#)

Selling in 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.

Topics

- [Restrictions and Limitations \(p. 258\)](#)
- [Registering as a Seller \(p. 258\)](#)
- [Pricing Your Reserved Instances \(p. 260\)](#)
- [Listing Your Reserved Instances \(p. 261\)](#)
- [Lifecycle of a Listing \(p. 262\)](#)

- [After Your Reserved Instance Is Sold \(p. 263\)](#)

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 \(p. 258\)](#).

The following limitations and restrictions apply when selling Reserved Instances:

- Only Amazon EC2 Standard Reserved Instances can be sold in the Reserved Instance Marketplace. Convertible Reserved Instances cannot be sold. There must be at least one month remaining in the term of the Standard Reserved Instance.
- 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 \(p. 260\)](#). You can also modify your Reserved Instances before listing them. For information, see [Modifying Reserved Instances \(p. 264\)](#).
- 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.
- Only Amazon EC2 Standard Reserved Instances can be sold in the Reserved Instance Marketplace. Other AWS Reserved Instances, such as Amazon RDS and Amazon ElastiCache Reserved Instances cannot be sold in the Reserved Instance Marketplace.

Registering 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 Accounts \(p. 258\)](#).
- **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 \(p. 259\)](#).

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.

Topics

- [Bank Accounts \(p. 258\)](#)
- [Tax Information \(p. 259\)](#)
- [Sharing Information with the Buyer \(p. 260\)](#)
- [Getting Paid \(p. 260\)](#)

Bank Accounts

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.

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 may 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](#).

Sharing Information 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](#).

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. You can view the state of this disbursement by viewing your Reserved Instance disbursement report. Disbursements take place once a day. Keep in mind that you can't receive disbursements until AWS has received verification from your bank. This period 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.

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. You cannot specify the usage fee or the recurring fee; The buyer pays the same usage or recurring fees that were set when the reservations were originally purchased.

The following are important limits to note:

- **You can sell up to \$50,000 in Reserved Instances per year.** To sell more, complete the [Request to Raise Sales Limit on Amazon EC2 Reserved Instances](#) 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.

Setting a Pricing Schedule

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.

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.

Topics

- [Listing Your Reserved Instance Using the AWS Management Console \(p. 261\)](#)
- [Listing Your Reserved Instances Using the AWS CLI or Amazon EC2 API \(p. 261\)](#)
- [Reserved Instance Listing States \(p. 262\)](#)

[Listing Your Reserved Instance Using the AWS Management Console](#)

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**.

[Listing Your Reserved Instances Using the AWS CLI or Amazon EC2 API](#)

To list a Reserved Instance 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.

To cancel your listing, use the [cancel-reserved-instances-listings](#) command.

To list a Reserved Instance in the Reserved Instance Marketplace using the Amazon EC2 API

- [DescribeReservedInstances](#)
- [CreateReservedInstancesListing](#)
- [DescribeReservedInstancesListings](#)
- [CancelReservedInstancesListing](#)

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 **State** 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 may create or sell a listing, or AWS may send 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.

Buying in 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 \(p. 251\)](#).

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 computing needs change, you can modify your Standard or Convertible Reserved Instances and continue to benefit from the billing benefit. You can modify the Availability Zone, scope, network platform, or instance size (within the same instance type) of your Reserved Instance. To modify a Reserved Instance, you specify the Reserved Instances that you want to modify, and you specify one or more target configurations.

Note

You can also exchange a Convertible Reserved Instance for another Convertible Reserved Instance with a different configuration, including instance family. For more information, see [Exchanging Convertible Reserved Instances \(p. 270\)](#).

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 [Modifying the Instance Size of Your Reservations \(p. 266\)](#).

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 [DescribeReservedInstances](#) API action or the [describe-reserved-instances](#) command (AWS CLI).)
- 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.

Note

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.

Topics

- Requirements and Restrictions for Modification (p. 265)
- Modifying the Instance Size of Your Reservations (p. 266)
- Submitting Modification Requests (p. 268)
- Troubleshooting Modification Requests (p. 269)

Requirements and Restrictions for Modification

Not all attributes of a Reserved Instance can be modified, and restrictions may apply.

Modifiable attribute	Supported platforms	Limitations
Change Availability Zones within the same region	All Windows and Linux	-
Change the scope from Availability Zone to Region and vice versa	All Windows and Linux	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 information, see How Reserved Instances Are Applied (p. 243) .
Change the network platform between EC2-VPC and EC2-Classic	All Windows and Linux	Only applicable if your account supports EC2-Classic.
Change the instance size within the same instance type	Supported on Linux, except for RedHat and SUSE Linux due to licensing differences. For more information about RedHat and SUSE pricing, see Amazon EC2 Reserved Instance Pricing . Not supported on Windows.	Some instance types are not supported, because there are no other sizes available. For more information, see Modifying the Instance Size of Your Reservations (p. 266) .

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 Instances that you want to modify must be:

- Active
- Not pending another modification request
- Not listed in the Reserved Instance Marketplace

Note

To modify your Reserved Instances that are listed in the Reserved Instance Marketplace, cancel the listing, request modification, and then list them again.

- Terminating in the same hour (but not minutes or seconds)
- Already purchased by you (you cannot modify an offering before or at the same time that you purchase it)

Your modification request must meet the following conditions:

- There must be a match between the instance size footprint of the active reservation and the target configuration. For more information, see [Modifying the Instance Size of Your Reservations \(p. 266\)](#).
- The input Reserved Instances must be either Standard Reserved Instances or Convertible Reserved Instances, but not a combination of both.

Modifying the Instance Size of Your Reservations

If you have Amazon Linux reservations in an instance type with multiple sizes, you can modify the instance size of your Reserved Instances.

Note

Instances are grouped by family (based on storage, or CPU capacity); type (designed for specific use cases); and size. For example, the `c4` instance type is in the Compute optimized instance family and is available in multiple sizes. While `c3` instances are in the same family, you can't modify `c4` instances into `c3` instances because they have different hardware specifications. For more information, see [Amazon EC2 Instance Types](#).

You cannot modify the instance size of the Reserved Instances for the following instance types, because only one size is available for each of these instance types.

- `cc2.8xlarge`
- `cr1.8xlarge`
- `hs1.8xlarge`
- `i3.metal`
- `t1.micro`

Each Reserved Instance has an *instance size footprint*, which is determined by the normalization factor of the instance type and the number of instances in the reservation. When you modify a Reserved Instance, the footprint of the target configuration must match that of the original configuration, otherwise the modification request is not processed.

The normalization factor is based on instance size within the instance type (for example, `m1.xlarge` instances within the `m1` instance type). This is only meaningful within the same instance type. Instance types cannot be modified from one type to another. In the Amazon EC2 console, this is measured in units. The following table illustrates the normalization factor that applies within an instance type.

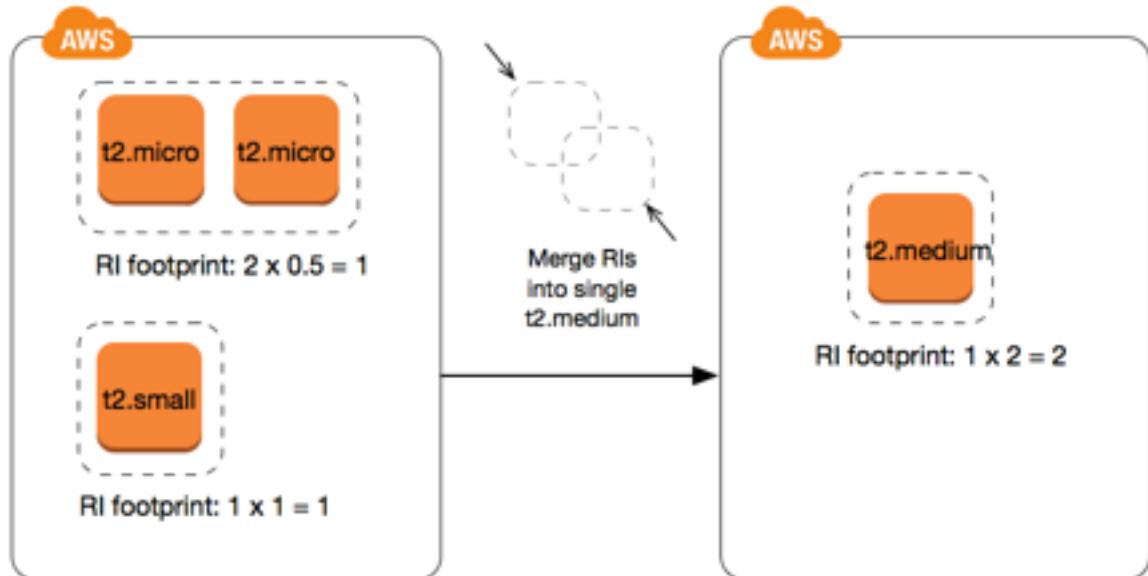
Instance size	Normalization factor
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
4xlarge	32

Instance size	Normalization factor
8xlarge	64
9xlarge	72
10xlarge	80
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256

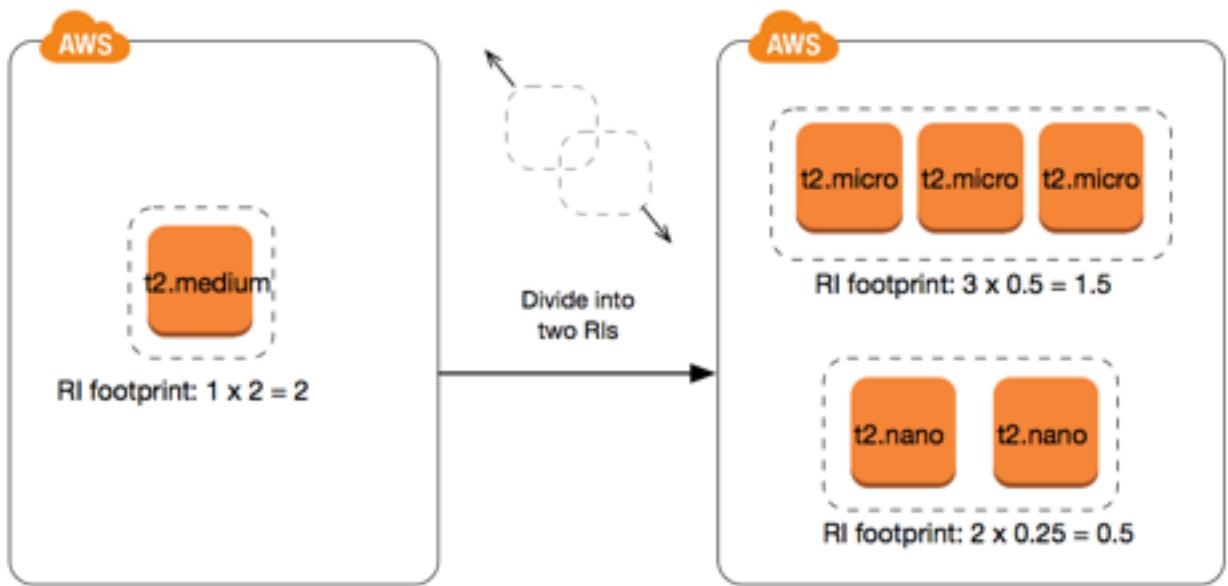
To calculate the instance size footprint of a Reserved Instance, multiply the number of instances by the normalization factor. For example, an `t2.medium` has a normalization factor of 2 so a reservation for four `t2.medium` instances has a footprint of 8 units.

You can allocate your reservations into different instance sizes across the same instance type 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) instance into four `t2.small` (4×1) instances, or 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` (2×1) instances into one `t2.large` (1×4) instance. This is because the existing instance size footprint of your current reservation is smaller than the proposed reservation.

In the following example, you have a reservation with two `t2.micro` instances (giving you a footprint of 1) and a reservation with one `t2.small` instance (giving you a footprint of 1). You merge both reservations to a single reservation with one `t2.medium` instance—the combined instance size footprint of the two original reservations equals the footprint of the modified reservation.



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. You divide the reservation into a reservation with two `t2.nano` instances and a reservation with three `t2.micro` instances.



Submitting Modification Requests

You can modify your Reserved Instances using the Amazon EC2 console, the Amazon EC2 API, or a command line tool.

Amazon EC2 Console

Before you modify your Reserved Instances, ensure that you have read the applicable [restrictions \(p. 265\)](#). If you are modifying instance size, ensure that you've calculated the total [instance size footprint \(p. 266\)](#) 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 **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 choose **Continue** when you're done:
 - **Network:** Choose whether the Reserved Instance applies to EC2-Classic or EC2-VPC. This option is only available if your account supports EC2-Classic.
 - **Scope:** Choose whether the Reserved Instance 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. Only available for supported platforms. For more information, see [Requirements and Restrictions for Modification \(p. 265\)](#).
 - **Count:** Specify the number of instances to be covered by the reservation.

Note

If your combined target configurations are larger or smaller than the instance size footprint of your original Reserved Instances, the allocated total in the **Units** column displays in red.

4. To confirm your modification choices when you finish specifying your target configurations, choose **Submit Modifications**. If you change your mind at any point, choose **Cancel** to exit the wizard.

You can determine the status of your modification request by looking at the **State** column in the Reserved Instances screen. The following table illustrates the possible **State** values.

State	Description
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	New Reserved Instances created from a successful modification request. -Or- Original Reserved Instances after a failed modification request.

Amazon EC2 API or Command Line Tool

To modify your Reserved Instances, you can use one of the following:

- [modify-reserved-instances](#) (AWS CLI)
- [Edit-EC2ReservedInstance](#) (AWS Tools for Windows PowerShell)
- [ModifyReservedInstances](#) (Amazon EC2 API)

To get the status of your modification, use one of the following:

- [describe-reserved-instances-modifications](#) (AWS CLI)
- [Get-EC2ReservedInstancesModifications](#) (AWS Tools for Windows PowerShell)
- [DescribeReservedInstancesModifications](#) (Amazon EC2 API)

The state returned shows your request as `processing`, `fulfilled`, or `failed`.

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 \(p. 265\)](#) 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 \(p. 270\)](#)
- [Calculating Convertible Reserved Instances Exchanges \(p. 271\)](#)
- [Merging Convertible Reserved Instances \(p. 272\)](#)
- [Exchanging a Portion of a Convertible Reserved Instance \(p. 272\)](#)
- [Submitting Exchange Requests \(p. 273\)](#)

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 \(p. 272\)](#). For more information about modifying your Reserved Instances, see [Modifying Reserved Instances \(p. 264\)](#).
- 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 \(p. 272\)](#).

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.

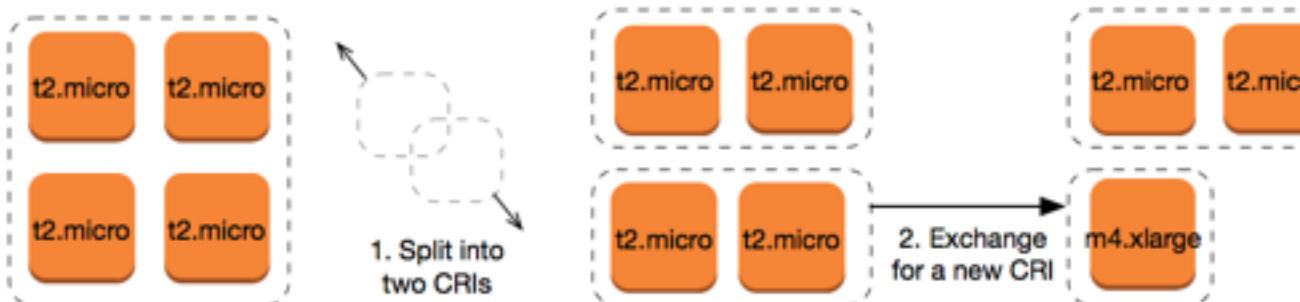
Exchanging a Portion of a Convertible Reserved Instance

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:

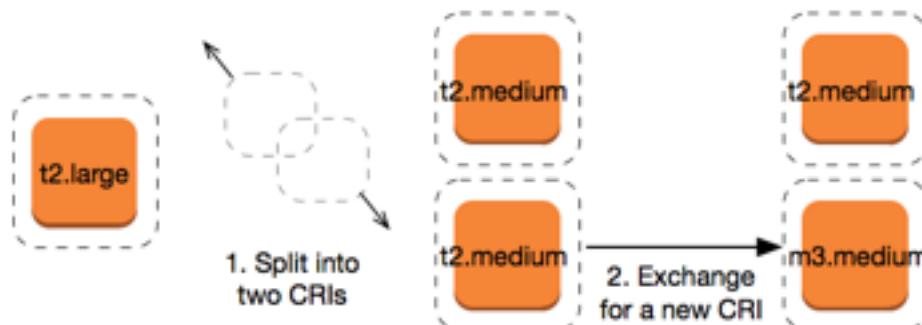
1. Modify the `t2.micro` Convertible Reserved Instance by splitting it into two `t2.micro` Convertible Reserved Instances with two instances each.
2. Exchange one of the new `t2.micro` Convertible Reserved Instances for an `m4.xlarge` Convertible Reserved Instance.



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. For more information, see [Modifying the Instance Size of Your Reservations \(p. 266\)](#).
2. Exchange one of the new `t2.medium` Convertible Reserved Instances for an `m3.medium` Convertible Reserved Instance.



For more information, see [Modifying the Instance Size of Your Reservations \(p. 266\)](#) and [Submitting Exchange Requests \(p. 273\)](#).

Not all Reserved Instances can be modified. Ensure that you read the applicable [restrictions \(p. 265\)](#).

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)
- [GetEC2-ReservedInstancesExchangeQuote](#) (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 \(p. 240\)](#). If you are flexible about when your instances run, Spot Instances might meet your needs and decrease costs. For more information, see [Spot Instances \(p. 278\)](#).

Contents

- [How Scheduled Instances Work \(p. 275\)](#)
- [Service-Linked Roles for Scheduled Instances \(p. 275\)](#)
- [Purchasing a Scheduled Instance \(p. 276\)](#)

- [Launching a Scheduled Instance \(p. 276\)](#)
- [Scheduled Instance Limits \(p. 277\)](#)

How Scheduled Instances Work

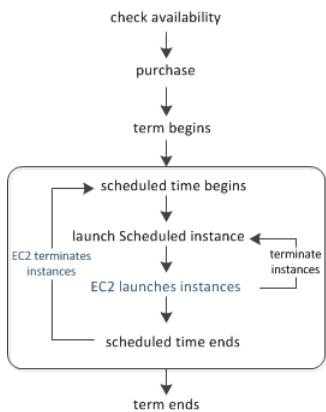
Amazon EC2 sets aside pools of EC2 instances in each Availability Zone for use as Scheduled Instances. Each pool supports a specific combination of instance type, operating system, and network (EC2-Classic or EC2-VPC).

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 using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **INSTANCES**, choose **Scheduled Instances**.
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 [date] for duration 4 hours

+/- 2 hours

Recurring Daily

5. 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 Linux/UNIX (Amazon VPC)

Instance type Any

Availability Zone Any

- 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 using the 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 using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **INSTANCES**, choose **Scheduled Instances**.
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 using the 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, C5, 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.

Spot Instances

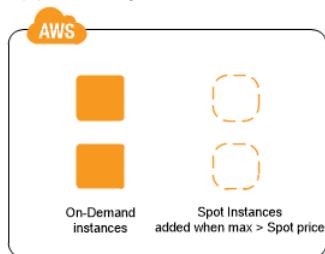
A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance is called a Spot price. The Spot price of each instance type in each Availability Zone is set by Amazon EC2, and 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](#).

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't stop and start an Amazon EBS-backed Spot Instance; only the Amazon EC2 Spot service can do this. The Amazon EC2 Spot service can interrupt (p. 325) 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 or terminated).

One strategy to maintain a minimum level of guaranteed compute resources for your applications is to launch a core group of On-Demand Instances , and supplement them with Spot Instances when the opportunity arises.



Another strategy is to launch Spot Instances with a required duration (also known as Spot blocks), which are not interrupted due to changes in the Spot price. For more information, see [Specifying a Duration for Your Spot Instances \(p. 290\)](#).

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, operating system, Availability Zone, and network platform (EC2-Classic or EC2-VPC).
- *Spot price* – The current price of a Spot Instance per hour.
- *Spot Instance 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 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.

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 \(p. 19\)](#)
- [Getting Started with Amazon EC2 Linux Instances \(p. 26\)](#)

Spot basics

- [How Spot Instances Work \(p. 281\)](#)
- [How Spot Fleet Works \(p. 282\)](#)

Working with Spot Instances

- [Preparing for Interruptions \(p. 327\)](#)
- [Creating a Spot Instance Request \(p. 292\)](#)
- [Getting Request Status Information \(p. 323\)](#)

Working with Spot Fleets

- [Spot Fleet Prerequisites \(p. 299\)](#)
- [Creating a Spot Fleet Request \(p. 302\)](#)

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 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](#) 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

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, or the Spot price exceeds your maximum price.

Spot Instances with a predefined duration use a fixed hourly price that remains in effect for the Spot Instance while it runs.

View Prices

To view the current (updated every five minutes) lowest Spot price per 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 \(p. 288\)](#).

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 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](#).

How Spot Instances Work

To use Spot Instances, create a *Spot Instance request* or a *Spot Fleet request*. The request can include the maximum price that you are willing to pay per hour per instance (the default is the On-Demand price), and other constraints such as the instance type and Availability Zone. If your maximum price exceeds the current Spot price for the specified instance, and capacity is available, your request is fulfilled immediately. Otherwise, the request is fulfilled whenever the maximum price exceeds the Spot price and the capacity is available. Spot Instances run until you terminate them or until Amazon EC2 must interrupt them (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 \(p. 325\)](#).

You can't stop and start an Amazon EBS-backed instance if it is a Spot Instance (only the Spot service can stop and start a Spot Instance), but you can reboot or terminate a Spot Instance.

Contents

- [Launching Spot Instances in a Launch Group \(p. 281\)](#)
- [Launching Spot Instances in an Availability Zone Group \(p. 281\)](#)
- [Launching Spot Instances in a VPC \(p. 282\)](#)

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, if your launch group includes instances in multiple Availability Zones and capacity in one of these Availability Zone 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 whether you specified the EC2-Classic network, a default VPC, or a nondefault VPC. For more information, see [Supported Platforms \(p. 637\)](#).

EC2-Classic

Amazon EC2 finds the lowest-priced Availability Zone in the region and launches your Spot Instances in that Availability Zone if the lowest price for the group is higher than the current Spot price in that Availability Zone. Amazon EC2 waits until there is enough capacity to launch your Spot Instances together, as long as the Spot price remains lower than the lowest price for the group.

Default VPC

Amazon EC2 uses the Availability Zone for the specified subnet, or if you don't specify a subnet, it selects an Availability Zone and its default subnet, but it might not be the lowest-priced Availability 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

To take advantage of the features of EC2-VPC when you use Spot Instances, specify in your Spot request that your Spot Instances are to be launched in a VPC. You specify a subnet for your Spot Instances the same way that you specify a subnet for your On-Demand Instances.

The process for making a Spot Instance request that launches Spot Instances in a VPC is the same as the process for making a Spot Instance request that launches Spot Instances in EC2-Classic—except for the following differences:

- 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 the Spot price exceeds the current Spot price and there is available capacity. The Spot Fleet also attempts to maintain its target capacity fleet if your Spot Instances are interrupted due to a change in Spot prices or available capacity.

A *Spot Instance pool* is a set of unused EC2 instances with the same instance type, operating system, Availability Zone, and network platform (EC2-Classic or EC2-VPC). 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 \(p. 283\)](#)

- [Spot Fleet Allocation Strategy \(p. 283\)](#)
- [Spot Price Overrides \(p. 284\)](#)
- [Spot Fleet Instance Weighting \(p. 284\)](#)
- [Walkthrough: Using Spot Fleet with Instance Weighting \(p. 286\)](#)

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 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.

Spot Fleet Allocation Strategy

The allocation strategy for 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.

`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 cheapest combination of instance types and Availability Zones based on the current Spot price.

For On-Demand Instance target capacity, Spot Fleet always selects the cheapest instance type based on the public On-Demand price, while continuing to follow the allocation strategy (either `lowestPrice` 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.

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 will 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.

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.large`, `c4.large`, and `c5.large`. The On-Demand price for `c5.large` is less than 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`.

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.

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 (.10 divided by 8)	2 (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 1 instance in each of the three pools, and the fourth instance in whichever of the three pools 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
<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 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 Prerequisites \(p. 299\)](#).

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 following [request-spot-fleet](#) command:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

For more information, see [Spot Fleet Requests \(p. 297\)](#).

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).

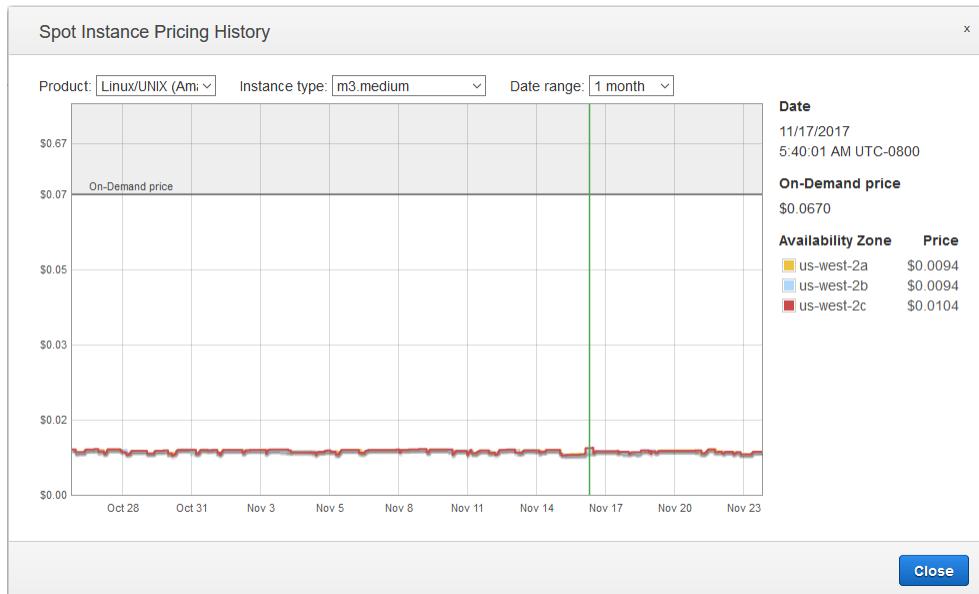
If Example Corp used the `diversified` strategy, the Spot Instances would come from all three pools. The Spot Fleet would launch 6 `r3.2xlarge` instances (which provide 6 units), 3 `r3.4xlarge` instances (which provide 6 units), and 2 `r3.8xlarge` instances (which provide 8 units), for a total of 20 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.

To view the Spot price history using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On 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**. By default, the page displays a graph of the data for Linux `t1.micro` instances in all Availability Zones over the past day. Move your pointer over the graph to display the prices at specific times in the table below the graph.



5. (Optional) To review the Spot price history for a specific Availability Zone, select an Availability Zone from the list. You can also select a different product, instance type, or date range.

To view the Spot price history using the command line

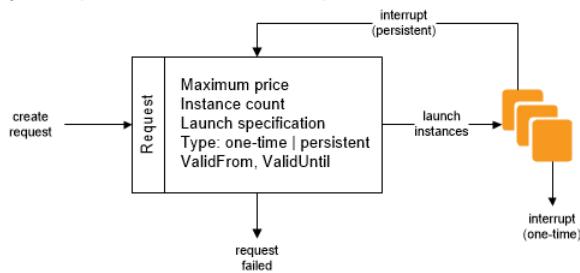
You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-spot-price-history](#) (AWS CLI)
- [Get-EC2SpotPriceHistory](#) (AWS Tools for Windows PowerShell)

Spot Instance Requests

To use Spot Instances, you create a Spot Instance request that includes the 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 action taken for a Spot Instance interruption depends on the request type (one-time or persistent) and the interruption behavior (hibernate, stop, or terminate). If the request is a persistent request, the request is opened again after your Spot Instance is interrupted.



Contents

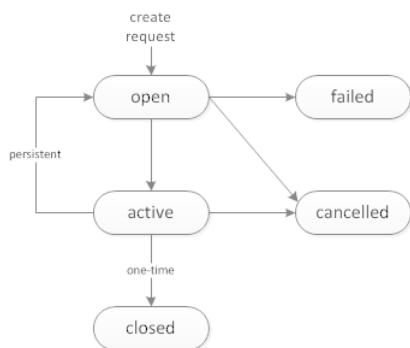
- [Spot Instance Request States \(p. 289\)](#)
- [Specifying a Duration for Your Spot Instances \(p. 290\)](#)
- [Specifying a Tenancy for Your Spot Instances \(p. 291\)](#)
- [Service-Linked Role for Spot Instance Requests \(p. 291\)](#)
- [Creating a Spot Instance Request \(p. 292\)](#)
- [Finding Running Spot Instances \(p. 294\)](#)
- [Tagging Spot Instance Requests \(p. 295\)](#)
- [Canceling a Spot Instance Request \(p. 295\)](#)
- [Spot Request Example Launch Specifications \(p. 296\)](#)

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.
- **cancelled**—You cancelled 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 the maximum price exceeds the Spot price or capacity becomes available again, the Spot Instance is started (if stopped), the Spot Instance is resumed (if hibernated), or the Spot Instance request is opened again and Amazon EC2 launches a new Spot Instance (if terminated).

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 \(p. 319\)](#).

Specifying a Duration for Your Spot Instances

Amazon EC2 does not terminate Spot Instances with a specified duration (also known as Spot blocks) when the Spot price changes. 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 specify 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 specify 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.

To launch Spot Instances with a specified duration using the console

Select the appropriate request type. For more information, see [Creating a Spot Instance Request \(p. 292\)](#).

To launch Spot Instances with a specified duration using the 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 specified duration using the 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 \(p. 345\)](#) 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 \(p. 292\)](#).
- 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 \(p. 347\)](#). 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 creates a service-linked role when you request Spot Instances. 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 **AWSServiceRoleForEC2Spot** to complete the following actions:

- `ec2:DescribeInstances` - Describe Spot Instances
- `ec2:StopInstances` - Stop Spot Instances
- `ec2:StartInstances` - Start Spot Instances

If you specify encrypted EBS snapshots for your Spot Instances and you use customer managed CMKs for encryption, you must grant the **AWSServiceRoleForEC2Spot** role access to the CMKs so that Amazon EC2 can launch Spot Instances on your behalf. The principal is the Amazon Resource Name (ARN) of the **AWSServiceRoleForEC2Spot** role. For more information, see [Using Key Policies in AWS KMS](#).

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*.

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.

Creating a Spot Instance Request

The process for requesting a Spot Instance is similar to the process for launching an On-Demand Instance. You can't change the parameters of your Spot request, including your maximum price, after you've submitted the request.

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 requests, see [Spot Request Status \(p. 319\)](#).

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 \(p. 288\)](#).

To create a Spot Instance request using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, choose **Spot Requests**.
3. If you are new to Spot Instances, you see a welcome page; choose **Get started**. Otherwise, choose **Request Spot Instances**.
4. For **Request type**, the default is **Request**, which specifies a one-time Spot request created using a Spot Fleet. To use Spot blocks instead, choose **Reserve for duration** and select the number of hours for the job to complete.

If you prefer to use **Request and Maintain**, see [Creating a Spot Fleet Request \(p. 302\)](#).

5. For **Target capacity**, enter the number of units to request. You can choose instances or performance characteristics that are important to your application workload, such as vCPUs, memory, and storage.
6. For **Requirements**, do the following:
 - a. [Spot Fleet] (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.
 - b. For **AMI**, choose one of the basic AMIs provided by AWS, or choose **Use custom AMI** to specify your own AMI.
 - c. For **Instance type(s)**, choose **Select**. Select the instance types that have the minimum hardware specifications that you need (vCPUs, memory, and storage).

- d. For **Network**, your account supports either the EC2-Classic and EC2-VPC platforms, or the EC2-VPC platform only. To find out which platforms your account supports, see [Supported Platforms \(p. 637\)](#).

[Existing VPC] Select the VPC.

[New VPC] Choose **Create new VPC** to go to the Amazon VPC console. When you are done, return to the wizard and refresh the list.

[EC2-Classic] Choose **EC2-Classic**.

- e. (Optional) For **Availability Zones**, the default is to let AWS choose the Availability Zones for your Spot Instances. If you prefer, you can specify specific Availability Zones.

[EC2-VPC] Select one or more Availability Zones. If you have more than one subnet in an Availability Zone, select the appropriate subnet from **Subnet**. To add subnets, select **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and refresh the list.

[EC2-Classic] Choose **Select specific zone/subnet**, and then select one or more Availability Zones.

- f. (Optional) To add storage, specify additional instance store volumes or EBS volumes, depending on the instance type. You can also enable EBS optimization.
- g. (Optional) By default, basic monitoring is enabled for your instances. To enable detailed monitoring, choose **Enable CloudWatch detailed monitoring**.
- h. (Optional) To run a Dedicated Spot Instance, for **Tenancy**, choose **Dedicated - run a dedicated instance**.
- i. For **Security groups**, select one or more security groups.
- j. [EC2-VPC] To connect to your instances in a VPC, enable **Auto-assign IPv4 Public IP**.
- k. (Optional) To connect to your instances, specify your key pair for **Key pair name**.
- l. (Optional) To launch your Spot Instances with an IAM role, for **IAM instance profile**, specify the role.
- m. (Optional) To run a start-up script, copy it to **User data**.
- n. [Spot Fleet] To add a tag, choose **Add new tag** and type the key and value for the tag. Repeat for each tag.

7. For **Spot request fulfillment**, do the following:

- a. [Spot Fleet] For **Allocation strategy**, choose the strategy that meets your needs. For more information, see [Spot Fleet Allocation Strategy \(p. 283\)](#).
- b. [Spot Fleet] For **Maximum price**, you can use the default maximum price (the On-Demand price) or specify the maximum price that 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.
- c. (Optional) To create a request that is valid only during a specific time period, edit the values for **Request valid from** and **Request valid until**.
- d. [Spot Fleet] By default, we terminate your Spot Instances when the request expires. To keep them running after your request expires, clear **Terminate instances at expiration**.

8. (Optional) To register your Spot Instances with a load balancer, choose **Receive traffic from one or more load balancers** and select 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**.

[Spot Fleet] The 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`.

[Spot block] The request type is `block` and the initial state is `open`. When the request is fulfilled, the state is `active` and the status is `fulfilled`.

To create a Spot Instance request using the AWS CLI

Use the following [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 following [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 Request Example Launch Specifications \(p. 296\)](#). 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).

Amazon EC2 launches your Spot Instance when the maximum price exceeds the Spot price and capacity is available. The Spot Instance runs until it is interrupted or you terminate it yourself. Use the following [describe-spot-instance-requests](#) command to monitor your Spot Instance request:

```
aws ec2 describe-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

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 will remain running, depending on demand.

To find running Spot Instances using the 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 then select the **Instances** tab.

3. Alternatively, in the navigation pane, choose **Instances**. In the top right corner, choose the **Show/Hide** icon, and then select **Lifecycle**. For each instance, **Lifecycle** is either `normal`, `spot`, or `scheduled`.

To find running Spot Instances using the AWS CLI

To enumerate your Spot Instances, use the [describe-spot-instance-requests](#) command with the `--query` option as follows:

```
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 as follows:

```
aws ec2 describe-instances --filters "Name=instance-lifecycle,Values=spot"
```

Tagging Spot Instance Requests

To help categorize and manage your Spot Instance requests, you can tag them with metadata of your choice. For more information, see [Tagging Your Amazon EC2 Resources \(p. 887\)](#).

You can assign a tag to a Spot Instance request after you create it. The tags that you create for your Spot Instance requests only apply to the requests. These tags are not added automatically to the Spot Instance that the Spot service launches to fulfill the request. You must add tags to a Spot Instance yourself after the Spot Instance is launched.

To add a tag to your Spot Instance request or Spot Instance using the AWS CLI

Use the following [create-tags](#) command to tag your resources:

```
aws ec2 create-tags --resources sir-08b93456 i-1234567890abcdef0 --tags  
Key=purpose,Value=test
```

Cancelling a Spot Instance Request

If you no longer want your Spot request, you can cancel it. You can only cancel Spot Instance requests that are `open` or `active`. Your Spot request is `open` when your request has not yet been fulfilled and no instances have been launched. Your Spot request is `active` when your request has been fulfilled, and Spot Instances have launched as a result. If your Spot request is `active` and has an associated running Spot Instance, canceling the request does not terminate the instance; you must terminate the running Spot Instance manually.

If the Spot request is a persistent Spot request, it 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 Spot request can launch a new instance.

To cancel a Spot Instance request using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**, and then select the Spot request.
3. Choose **Actions**, and then choose **Cancel spot request**.
4. (Optional) If you are finished with the associated Spot Instances, you can terminate them. In the navigation pane, choose **Instances**, select the instance, choose **Actions**, choose **Instance State**, and then choose **Terminate**.

To cancel a Spot Instance request using the AWS CLI

Use the following [cancel-spot-instance-requests](#) command to cancel the specified Spot request:

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

If you are finished with the associated Spot Instances, you can terminate them manually using the following [terminate-instances](#) command:

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7
```

Spot 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 \(p. 292\)](#).

1. [Launch Spot Instances \(p. 296\)](#)
2. [Launch Spot Instances in the specified Availability Zone \(p. 296\)](#)
3. [Launch Spot Instances in the specified subnet \(p. 297\)](#)
4. [Launch a Dedicated Spot Instance \(p. 297\)](#)

Example 1: Launch Spot Instances

The following example does not include an Availability Zone or subnet. Amazon EC2 selects an Availability Zone for you. If your account supports EC2-VPC only, Amazon EC2 launches the instances in the default subnet of the selected Availability Zone. If your account supports EC2-Classic, Amazon EC2 launches the instances in EC2-Classic in 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"
    }
}
```

You can specify security groups for EC2-Classic either by ID or by name (using the `SecurityGroups` field). You must specify security groups for EC2-VPC by ID.

Example 2: Launch Spot Instances in the Specified Availability Zone

The following example includes an Availability Zone. If your account supports EC2-VPC only, Amazon EC2 launches the instances in the default subnet of the specified Availability Zone. If your account supports EC2-Classic, Amazon EC2 launches the instances in EC2-Classic in 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 \(p. 282\)](#).

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

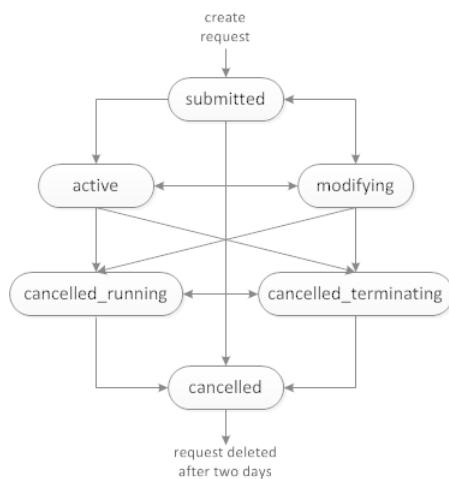
- [Spot Fleet Request States \(p. 298\)](#)
- [Spot Fleet Prerequisites \(p. 299\)](#)
- [Spot Fleet and IAM Users \(p. 299\)](#)
- [Spot Fleet Health Checks \(p. 300\)](#)
- [Planning a Spot Fleet Request \(p. 301\)](#)
- [Service-Linked Role for Spot Fleet Requests \(p. 301\)](#)
- [Creating a Spot Fleet Request \(p. 302\)](#)
- [Monitoring Your Spot Fleet \(p. 304\)](#)
- [Modifying a Spot Fleet Request \(p. 304\)](#)
- [Canceling a Spot Fleet Request \(p. 305\)](#)
- [Spot Fleet Example Configurations \(p. 306\)](#)

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 Spot 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 cancelled.
- `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 cancelled. A one-time `request` cannot be modified, and this state does not apply to such Spot requests.
- `cancelled_running` – The Spot Fleet is cancelled and will 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 cancelled and its Spot Instances are terminating. The request remains in this state until all instances are terminated.
- `cancelled` – The Spot Fleet is cancelled 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 cancelled immediately.



Spot Fleet Prerequisites

If you use the Amazon EC2 console to create a Spot Fleet, it creates a role named `aws-ec2-spot-fleet-tagging-role` that grants the Spot Fleet permission to request, launch, terminate, and tag instances on your behalf. This role is selected when you create your Spot Fleet request. If you use the AWS CLI or an API instead, you must ensure that this role exists. You can either use the Request Spot Instances wizard (the role is created when you advance to the second page of the wizard) or use the IAM console as follows.

To create the IAM role for 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**, and then choose **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 then choose **Create role**.

Spot Fleet and IAM Users

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 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 Spot 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`

4. On the **Review policy** page, type a policy name and description, and then choose **Create policy**.
5. In the navigation pane, choose **Users**, and then choose the user.
6. On the **Permissions** tab, choose **Add permissions**.
7. Choose **Attach existing policies directly**. Select the policy you created above and choose **Next: Review**.
8. Choose **Add permissions**.

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 \(p. 474\)](#).

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 target capacity in instances or in custom units. For more information, see [Spot Fleet Instance Weighting \(p. 284\)](#).
- Determine what portion of the Spot Fleet target capacity must be On-Demand capacity. You can specify 0 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 \(p. 306\)](#).

Service-Linked Role for Spot Fleet Requests

Amazon EC2 creates a service-linked role when you request a Spot Fleet. 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 **AWSServiceRoleForEC2SpotFleet** to complete the following actions:

- `ec2:RequestSpotInstances` - Request Spot Instances
- `ec2:TerminateInstances` - Terminate Spot Instances
- `ec2:DescribeImages` - Describe Amazon Machine Images (AMI) for the Spot Instances
- `ec2:DescribeInstanceStatus` - Describe the status of the Spot Instances
- `ec2:DescribeSubnets` - Describe the subnets for Spot Instances
- `ec2:CreateTags` - Add system tags to Spot Instances

Amazon EC2 also creates the **AWSServiceRoleForEC2Spot** role when you request a Spot Fleet. For more information, see [Service-Linked Role for Spot Instance Requests \(p. 291\)](#).

If you had an active Spot Fleet request before November 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 Account](#) in the *IAM User Guide*.

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.

Creating a Spot Fleet Request

When you create a Spot Fleet request, you must specify information about the Spot Instances to launch, such as the instance type and the maximum price you are willing to pay.

To create a Spot Fleet request using the 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 **Request type**, select **Request or Request and Maintain**.
4. For **Total required capacity**, type the number of units to request for target capacity. You can choose instances or performance characteristics that are important to your application workload, such as vCPUs, memory, and storage. If the request type is **Request and Maintain**, you can specify a target capacity of 0 and add capacity later.
5. (Optional) For **Optional On-Demand portion**, type the number of On-Demand units to request. The number must be less than the **Total required capacity**. Amazon EC2 calculates the difference, and allocates the difference to Spot units to request.
6. For **Requirements**, 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 specify **Optional On-Demand portion**, you must choose a launch template.
 - b. For **AMI**, choose one of the basic Amazon Machine Images (AMI) provided by AWS, or choose **Use custom AMI** to use an AMI from our user community, the AWS Marketplace, or one of your own.
 - c. For **Instance type(s)**, choose **Select**. Select the instance types that have the minimum hardware specifications that you need (vCPUs, memory, and storage).
 - d. For **Network**, your account supports either the EC2-Classic and EC2-VPC platforms, or the EC2-VPC platform only. To find out which platforms your account supports, see [Supported Platforms \(p. 637\)](#).

[Existing VPC] Select the VPC.

[New VPC] Select **Create new VPC** to go the Amazon VPC console. When you are done, return to the wizard and refresh the list.

[EC2-Classic] Select **EC2-Classic**.

- e. (Optional) For **Availability Zones**, the default is to let AWS choose the Availability Zones for your Spot Instances. If you prefer, you can specify specific Availability Zones.

[EC2-VPC] Select one or more Availability Zones. If you have more than one subnet in an Availability Zone, select the appropriate subnet from **Subnet**. To add subnets, select **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and refresh the list.

[EC2-Classic] Select **Select specific zone/subnet**, and then select one or more Availability Zones.

- f. (Optional) To add storage, specify additional instance store volumes or EBS volumes, depending on the instance type. You can also enable EBS optimization.
- g. (Optional) By default, basic monitoring is enabled for your instances. To enable detailed monitoring, select **Enable CloudWatch detailed monitoring**.

- h. (Optional) To replace unhealthy instances in a **Request and Maintain** Spot Fleet, select **Replace unhealthy instances**.
 - i. (Optional) To run a Dedicated Spot Instance, choose **Dedicated - run a dedicated instance for Tenancy**.
 - j. (Optional) 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. To do so, choose the corresponding option from **Interruption behavior**.
 - k. For **Security groups**, select one or more security groups.
 - l. [EC2-VPC] If you need to connect to your instances in a VPC, you can enable **Auto-assign IPv4 Public IP**.
 - m. (Optional) If you need to connect to your instances, specify your key pair using **Key pair name**.
 - n. (Optional) To launch your Spot Instances with an IAM role, choose the role for **IAM instance profile**.
 - o. (Optional) To run a start-up script, copy it to **User data**.
 - p. (Optional) To add a tag, choose **Add new tag** and type the key and value for the tag. Repeat for each tag.
7. For **Spot request fulfillment**, do the following:
 - a. For **Allocation strategy**, choose the strategy that meets your needs. For more information, see [Spot Fleet Allocation Strategy \(p. 283\)](#).
 - b. For **Maximum price**, you can use the default maximum price (the On-Demand price) or specify the maximum price 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 selected.
 - c. (Optional) To create a request that is valid only during a specific time period, edit **Request valid from** and **Request valid until**.
 - d. (Optional) By default, we terminate your Spot Instances when the request expires. To keep them running after your request expires, clear **Terminate instances at expiration**.
 8. (Optional) To register your Spot Instances with a load balancer, select **Receive traffic from one or more load balancers** and select 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 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 following `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 \(p. 306\)](#).

The following is example output:

```
{  
    "SpotFleetRequestId": "sfr-73fb2ce-aa30-494c-8788-1cee4EXAMPLE"  
}
```

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 using the 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. The configuration details are available in the **Description** tab.
4. To list the Spot Instances for the Spot Fleet, choose the **Instances** tab.
5. To view the history for the Spot Fleet, choose the **History** tab.

To monitor your Spot Fleet using the AWS CLI

Use the following [describe-spot-fleet-requests](#) command to describe your Spot Fleet requests:

```
aws ec2 describe-spot-fleet-requests
```

Use the following [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 following [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
- Decrease the target capacity

Note

You can't modify a one-time Spot Fleet request.

You can only modify the Spot Instance portion of a Spot Fleet request; you can't modify the On-Demand Instance portion of a Spot Fleet request.

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 using the console

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot/home/fleet>.
2. Select your Spot Fleet request.
3. Choose **Actions**, and then choose **Modify target capacity**.
4. In **Modify target capacity**, do the following:
 - a. Enter the new target capacity.
 - b. (Optional) If you are decreasing the target capacity but want to keep the fleet at its current size, deselect **Terminate instances**.
 - c. Choose **Submit**.

To modify a Spot Fleet request using the AWS CLI

Use the following `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-73fdbd2ce-  
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-73fdbd2ce-  
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 using the console

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot/home/fleet>.
2. Select your Spot Fleet request.
3. Choose **Actions**, and then choose **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, deselect **Terminate instances**. When you are ready, choose **Confirm**.

To cancel a Spot Fleet request using the AWS CLI

Use the following `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-73fdbd2ce-  
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 \(p. 302\)](#).

1. [Launch Spot Instances using the lowest-priced Availability Zone or subnet in the region \(p. 306\)](#)
2. [Launch Spot Instances using the lowest-priced Availability Zone or subnet in a specified list \(p. 307\)](#)
3. [Launch Spot Instances using the lowest-priced instance type in a specified list \(p. 308\)](#)
4. [Override the price for the request \(p. 309\)](#)
5. [Launch a Spot Fleet using the diversified allocation strategy \(p. 310\)](#)
6. [Launch a Spot Fleet using instance weighting \(p. 311\)](#)
7. [Launch a Spot Fleet with On-Demand capacity \(p. 313\)](#)

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. If your account supports EC2-VPC only, the Spot Fleet launches the instances in the lowest-priced Availability Zone that has a default subnet. If your account supports EC2-Classic, the Spot Fleet launches the instances in EC2-Classic in the lowest-priced Availability Zone. The price you pay will 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

If your account supports EC2-VPC only, the Spot Fleet launches the instances in the default subnet of the lowest-priced Availability Zone that you specified. If your account supports EC2-Classic, the Spot Fleet launches the instances in 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 3 launch specifications, such that there are 10 instances of each type. For more information, see [Spot Fleet Allocation Strategy \(p. 283\)](#).

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"  
        }  
    ]  
}
```

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 \(p. 284\)](#).

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  
        }  
    ]  
}
```

Priority

You can also use instance weighting to give priority to an Availability Zone or subnet. For example, the following launch specifications are nearly identical, except that they specify different subnets and weights. The Spot Fleet finds the specification with the highest value for `WeightedCapacity`, and attempts to provision the request in the least expensive Spot Instance pool in that subnet. The second launch specification does not include a weight, so it defaults to 1.

```
{  
    "SpotPrice": "0.42",  
    "TargetCapacity": 40,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c3.2xlarge",  
            "SubnetId": "subnet-482e4972",  
            "WeightedCapacity": 2  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c3.2xlarge",  
            "SubnetId": "subnet-bb3337d"  
        }  
    ]  
}
```

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. The On-Demand request is always fulfilled if there is capacity, while 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 EC2 capacity and availability.

For more information, see [On-Demand in Spot Fleet \(p. 283\)](#).

```
{  
    "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 Disable Detailed Monitoring for Your Instances \(p. 483\)](#).

For more information about CloudWatch metrics provided by Amazon EC2, see [Monitoring Your Instances Using CloudWatch \(p. 483\)](#).

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 \(p. 485\)](#).

The `AWS/EC2Spot` namespace includes the following metrics.

Metric	Description
<code>AvailableInstancePoolsCount</code>	The Spot Instance pools specified in the Spot Fleet request. Units: Count
<code>BidsSubmittedForCapacity</code>	The capacity for which Amazon EC2 has submitted bids. Units: Count
<code>EligibleInstancePoolCount</code>	The Spot Instance pools specified in the Spot Fleet request where Amazon EC2 can fulfill bids. Amazon EC2 will not fulfill bids in pools where your bid price is less than the Spot price or the Spot price is greater than the price for On-Demand instances. Units: Count
<code>FulfilledCapacity</code>	The capacity that Amazon EC2 has fulfilled. Units: Count
<code>MaxPercentCapacityAllocation</code>	The maximum value of <code>PercentCapacityAllocation</code> across all Spot Instance pools specified in the Spot Fleet request. Units: Percent
<code>PendingCapacity</code>	The difference between <code>TargetCapacity</code> and <code>FulfilledCapacity</code> . Units: Count
<code>PercentCapacityAllocation</code>	The capacity allocated for the Spot Instance pool for the specified dimensions. To get the maximum value recorded across all Spot Instance pools, use <code>MaxPercentCapacityAllocation</code> .

Metric	Description
	Units: Percent
TargetCapacity	The target capacity of the Spot Fleet request. Units: Count
TerminatingCapacity	The capacity that is being terminated due to Spot Instance interruptions. 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, you can 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.

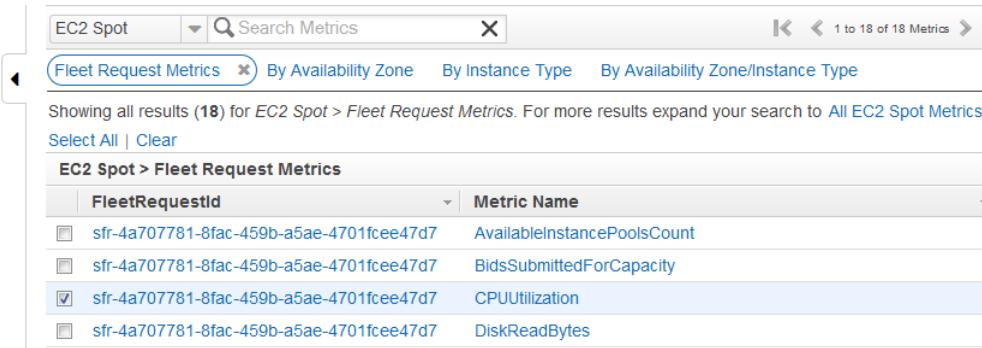
View the CloudWatch Metrics for Your Spot Fleet

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, under **Metrics**, choose the **EC2 Spot** namespace.
3. (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
4. To view the data for a metric, select the check box next to the metric.



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.

If you are using instance weighting, keep in mind that Spot Fleet can exceed the target capacity as needed, and that 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 auto 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 auto 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.

You can also configure the cooldown period for a scaling policy. 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, auto scaling scales out your scalable target immediately.

Spot Fleet supports the following types of scaling policies:

- [Target tracking scaling \(p. 316\)](#)—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 \(p. 317\)](#)—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.

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.

Note that 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.

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 configure a target tracking policy using the 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 then choose the **Auto Scaling** tab.
- 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**.
- Type 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 auto 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 auto scaling process adds 1 to 10 to get 11, so Spot Fleet launches 1 instance.

Note that when a Spot Fleet terminates an instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

Limits

- 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.
- If you use the AWS Management Console to enable automatic scaling for your Spot Fleet, it creates a role named `aws-ec2-spot-fleet-autoscale-role` that grants Amazon EC2 Auto Scaling permission to describe the alarms for your policies, monitor the current capacity of the fleet, and modify the capacity of the fleet. If you configure automatic scaling using the AWS CLI or an API, you can use this role if it exists, or manually create your own role for this purpose.

To create a role manually

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 **AWS service, EC2, EC2 - Spot Fleet Auto Scaling**, and then choose **Next: Permissions**.
5. On the **Attached permissions policy** page, choose **Next:Review**.
6. On the **Review** page, type a name for the role and then choose **Create role**.

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. For **CloudWatch Metrics by Category**, choose a category. For example, choose **EC2 Spot Metrics, Fleet Request Metrics**.
5. Select a metric, and then choose **Next**.
6. For **Alarm Threshold**, type a name and description for the alarm, and set the threshold value and number of time periods for the alarm.
7. (Optional) To receive notification of a scaling event, for **Actions**, choose **New list** and type your email address. Otherwise, you can delete the notification now and add one later as needed.

8. Choose **Create Alarm**.

To configure step scaling policies for your Spot Fleet using the 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 then choose the **Auto Scaling** tab.
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** to add a 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.

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.

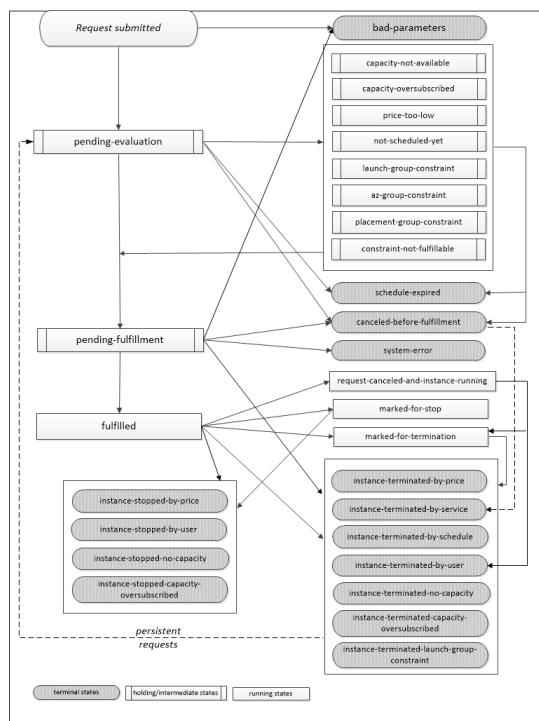
At each step of the process—also called the Spot request *lifecycle*, specific events determine successive request states.

Contents

- [Life Cycle of a Spot Request \(p. 319\)](#)
- [Getting Request Status Information \(p. 323\)](#)
- [Spot Request Status Codes \(p. 323\)](#)

Life Cycle 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 make a Spot Instance request, it goes into the `pending-evaluation` state unless one or more request parameters is not valid (`bad-parameters`).

Status Code	Request State	Instance State
<code>pending-evaluation</code>	<code>open</code>	n/a
<code>bad-parameters</code>	<code>closed</code>	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.

Status Code	Request State	Instance State
<code>capacity-not-available</code>	<code>open</code>	n/a
<code>capacity-oversubscribed</code>	<code>open</code>	n/a
<code>price-too-low</code>	<code>open</code>	n/a
<code>not-scheduled-yet</code>	<code>open</code>	n/a
<code>launch-group-constraint</code>	<code>open</code>	n/a

Status Code	Request State	Instance State
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, you cancel the request, or 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 cancelled by the user before a Spot Instance was launched, or 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 cancelled.

Status Code	Request State	Instance State
fulfilled	active	<code>pending</code> → <code>running</code>
fulfilled	active	<code>stopped</code> → <code>running</code>

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. For example, if your price equals the Spot price but Spot Instances are not available, the status code is `instance-terminated-capacity-oversubscribed`. 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>closed</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-capacity-oversubscribed</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>
<code>instance-terminated-by-user</code> †	<code>closed or cancelled *</code>	<code>terminated</code>
<code>instance-terminated-no-capacity</code>	<code>closed (one-time), open (persistent)</code>	<code>terminated</code>
<code>instance-terminated-capacity-oversubscribed</code>	<code>closed (one-time), open (persistent)</code>	<code>terminated</code>
<code>instance-terminated-launch-group-constraint</code>	<code>closed (one-time), open (persistent)</code>	<code>terminated</code>

† A Spot Instance can only get to this state if a user runs the shutdown command from the instance. We do not recommend that you do this, as the Spot service might restart the instance.

* 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. Note that 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 using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**, and then select the Spot request.
3. Check the value of **Status** in the **Description** tab.

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.

cancelled-before-fulfillment

The user cancelled the Spot request before it was fulfilled.

capacity-not-available

There is not enough capacity available for the instances that you requested.

capacity-oversubscribed

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 ran **shutdown -h** from the instance.

`instance-stopped-capacity-oversubscribed`

Your instance was stopped because the number of Spot requests with maximum prices equal to or higher than the Spot price exceeded the available capacity in this Spot Instance pool. (Note that the Spot price might not have changed.)

`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-capacity-oversubscribed`

Your instance was terminated because the number of Spot requests with maximum prices equal to or higher than the Spot price exceeded the available capacity in this Spot Instance pool. (Note that the Spot price might not have changed.)

`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 will not be 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 customer support for assistance.

Spot Instance Interruptions

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 will be interrupted. Therefore, you must ensure that your application is prepared for a Spot Instance interruption.

The following are the possible reasons that Amazon EC2 will 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 Spot 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.

An On-Demand Instance specified in a Spot Fleet cannot be interrupted.

Interruption Behavior

You can specify whether Amazon EC2 should hibernate, stop, or terminate Spot Instances when they are interrupted. You can choose the interruption behavior that meets your needs. The default is to terminate Spot Instances when they are interrupted. To change the interruption behavior, choose an option from

Interruption behavior in the console or `Instance Interruption Behavior` in the launch configuration or the launch template.

Stopping Interrupted Spot Instances

You can change the 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`, not `one-time`. You cannot specify a launch group in the Spot Instance request.
- For a Spot Fleet request, the type must be `maintain`, not `request`.
- The root volume must be an EBS volume, not an instance store volume.

After a Spot Instance is stopped by the Spot service, it can only be restarted by the Spot service, and only using the same launch configuration. When capacity is available that matches the Availability Zone and instance type of a Spot Instance that is stopped, the Spot Instance is started. With Spot Fleet, if capacity is available only with a different Availability Zone or instance type, Spot Fleet launches a new Spot Instance using the launch configuration with available 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 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 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 change the 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`, not `one-time`. You cannot specify a launch group in the Spot Instance request.
- For a Spot Fleet request, the type must be `maintain`, not `request`.
- 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
- 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 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. If your AMI does not have an encrypted root volume, you can copy it to a new AMI and request encryption. For more information, see [Amazon EBS Encryption \(p. 819\)](#) and [Copying an AMI \(p. 142\)](#).

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 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.

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 \(p. 328\)](#).

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 won't be 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 will be 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 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:

```
sudo apt-get install hibagent
```

2. Add the following to user data:

```
#!/bin/bash
/usr/bin/enable-ec2-spot-hibernation
```

Spot Instance Interruption Notices

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 interruption notices*, which provide a two-minute warning before Amazon EC2 must interrupt your Spot Instance. We recommend that you check for these warnings every 5 seconds.

This warning is made available as a CloudWatch event and as item in the [instance metadata \(p. 454\)](#) on the Spot Instance.

EC2 Spot Instance Interruption Warning

When Amazon EC2 interrupts your Spot Instance, it emits an event that can be detected by Amazon CloudWatch Events. For more information, see the [Amazon CloudWatch Events User Guide](#).

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 hibernated, stopped, or terminated by the Spot service, the `instance-action` item is present in your instance metadata. Otherwise, it is not present. You can retrieve `instance-action` as follows.

```
[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 that this instance will be stopped:

```
{"action": "stop", "time": "2017-09-18T08:22:00Z"}
```

The following example indicates the time that this instance will be terminated:

```
{"action": "terminate", "time": "2017-09-18T08:22:00Z"}
```

The following example indicates that hibernation has started immediately:

```
{"action": "hibernate", "time": "2017-11-28T08:22:00Z"}
```

If Amazon EC2 is not preparing to hibernate, stop, or terminate the instance, or if you terminated the instance yourself, `instance-action` is not present and you receive an HTTP 404 error.

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.

```
[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 will receive 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`. Note that `termination-time` remains in the instance metadata with the original approximate time, which is now in the past.

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 very 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 won't receive a data feed file for that hour.

Contents

- [Data Feed File Name and Format \(p. 330\)](#)
- [Amazon S3 Bucket Requirements \(p. 331\)](#)
- [Subscribing to Your Spot Instance Data Feed \(p. 331\)](#)
- [Deleting Your Spot Instance Data Feed \(p. 331\)](#)

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 `myawsbucket` and your prefix is `myprefix`, your file names are similar to the following:

```
myawsbucket.s3.amazonaws.com/myprefix/111122223333.2014-03-17-20.001.pwBdGTJG.gz
```

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 <code>m1.small</code> Spot Instances, this field is set to <code>SpotUsage</code> . For all other instance types, this field is set to <code>SpotUsage:{instance-type}</code> . For example, <code>SpotUsage:c1.medium</code> .
Operation	The product being charged for. For Linux Spot Instances, this field is set to <code>RunInstances</code> . For Windows Spot Instances, this field is set to <code>RunInstances:0002</code> . 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.

Field	Description
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 following `create-spot-datafeed-subscription` command:

```
aws ec2 create-spot-datafeed-subscription --bucket myawsbucket [--prefix myprefix]
```

The following is example output:

```
{  
    "SpotDatafeedSubscription": {  
        "OwnerId": "111122223333",  
        "Prefix": "myprefix",  
        "Bucket": "myawsbucket",  
        "State": "Active"  
    }  
}
```

Deleting Your Spot Instance Data Feed

To delete your data feed, use the following `delete-spot-datafeed-subscription` command:

```
aws ec2 delete-spot-datafeed-subscription
```

Spot Instance Limits

Spot Instance requests are subject to the following limits:

Limits

- [Spot Request Limits \(p. 332\)](#)
- [Spot Fleet Limits \(p. 332\)](#)
- [T2 Instances \(p. 332\)](#)

Spot Request Limits

By default, there is an account limit of 20 Spot Instances per region. If you terminate your Spot Instance but do not cancel the request, the request counts against this limit until Amazon EC2 detects the termination and closes the request.

Spot Instance limits are dynamic. When your account is new, your limit might be lower than 20 to start, but increase over time. In addition, your account might have limits on specific Spot Instance types. If you submit a Spot Instance request and you receive the error `Max spot instance count exceeded`, you can complete the AWS Support Center [Create Case](#) form to request an Amazon EC2 instance limit increase. For **Use Case Description**, indicate that you need an increase in your limits for Spot Instance requests. For more information, see [Amazon EC2 Service Limits \(p. 896\)](#).

Spot Fleet Limits

The usual Amazon EC2 limits apply to instances launched by a Spot Fleet, such as Spot request price limits, instance limits, and volume limits. In addition, the following limits apply:

- The number of active Spot Fleets per region: 1,000
- The number of launch specifications per fleet: 50
- The size of the user data in a launch specification: 16 KB
- The target capacity per Spot Fleet: 3,000
- The target capacity across all Spot Fleets in a region: 5,000
- A Spot Fleet request can't span regions.
- A Spot Fleet request can't span different subnets from the same Availability Zone.

T2 Instances

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 [T2 Unlimited \(p. 180\)](#), or use an instance type with dedicated CPU (for example, `c4.large`).

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, Linux Enterprise Server, and so on.

Contents

- [Differences between Dedicated Hosts and Dedicated Instances \(p. 333\)](#)

- [Bring Your Own License \(p. 333\)](#)
- [Dedicated Host Instance Capacity \(p. 334\)](#)
- [Dedicated Hosts Limitations and Restrictions \(p. 334\)](#)
- [Pricing and Billing \(p. 334\)](#)
- [Working with Dedicated Hosts \(p. 335\)](#)
- [Tracking Configuration Changes \(p. 344\)](#)

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. 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	Not supported	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, but 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've imported 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 may 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 \(p. 344\)](#).

Dedicated Host Instance Capacity

Dedicated Hosts are configured to support a single instance type and size capacity. The number of instances you can launch onto a Dedicated Host depends on the instance type that the Dedicated Host is configured to support. For example, if you allocated a `c3.xlarge` Dedicated Host, you'd have the right to launch up to eight `c3.xlarge` instances on the Dedicated Host. To determine the number of instance type sizes that you can run on a particular Dedicated Host, see [Amazon EC2 Dedicated Hosts Pricing](#).

Dedicated Hosts Limitations and Restrictions

Before you allocate Dedicated Hosts, take note of the following limitations and restrictions:

- RHEL, SUSE Linux, and Windows AMIs offered by AWS or on the AWS Marketplace cannot be used with Dedicated Hosts.
- Amazon EC2 instance recovery is not supported.
- 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.
- Host limits are independent from instance limits. Instances that you are running on Dedicated Hosts do not count towards your instance limits.
- Auto Scaling groups are not supported.
- 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. Placement groups are not supported for Dedicated Hosts.

Pricing and Billing

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 are charged an hourly rate for the Dedicated Host, regardless of the quantity or the size of instances that you choose to launch on it. In other words, you are charged for the entire Dedicated Host, and not the individual instances that you choose to run 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 \(p. 342\)](#).

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](#).

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.

Contents

- [Understanding Auto-Placement and Affinity \(p. 335\)](#)
- [Allocating Dedicated Hosts \(p. 336\)](#)
- [Launching Instances onto Dedicated Hosts \(p. 337\)](#)
- [Modifying Dedicated Host Auto-Placement \(p. 338\)](#)
- [Modifying Instance Tenancy and Affinity \(p. 339\)](#)
- [Viewing Dedicated Hosts \(p. 340\)](#)
- [Tagging Dedicated Hosts \(p. 340\)](#)
- [Monitoring Dedicated Hosts \(p. 341\)](#)
- [Releasing Dedicated Hosts \(p. 342\)](#)
- [Purchasing Dedicated Host Reservations \(p. 342\)](#)
- [Viewing Dedicated Host Reservations \(p. 343\)](#)

Understanding Auto-Placement and Affinity

Placement control happens on both the instance level and host level.

Auto-Placement

Auto-placement allows you to manage whether instances that you launch are launched onto a specific host, or onto any available host that has matching configurations. Auto-placement must be configured at the host level.

When a Dedicated Host's auto-placement 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 a Dedicated Host's auto-placement 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 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).

Allocating Dedicated Hosts

To begin using Dedicated Hosts, they need to be allocated to your account. You can allocate Dedicated Hosts to your account using the Amazon EC2 console or the command line tools.

To allocate Dedicated Hosts 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**, and then choose **Allocate Dedicated Host**.
3. Configure the following Dedicated Host options:
 - a. **Instance type**—The type of instance you want to launch on the Dedicated Host.
 - b. **Availability Zone**—The Availability Zone in which the Dedicated Host is located.
 - c. **Allow instance auto-placement**—Choose one of the following settings:
 - Yes—The Dedicated Host accepts untargeted instance launches that match its instance type configuration.
 - No—The Dedicated Host accepts *host* tenancy instances launches that specify its unique host ID only. This is the default setting.

For more information about auto-placement, see [Understanding Auto-Placement and Affinity \(p. 335\)](#).

- d. **Quantity**—The number of Dedicated Hosts to allocate with these options.
4. Choose **Allocate host**.

To allocate Dedicated Hosts using the command line tools

Use one of the following commands. The following examples allocate a Dedicated Host that supports *untargeted m4.large* instance launches in the `eu-west-1a` Availability Zone.

- [allocate-hosts \(AWS CLI\)](#)

```
aws ec2 allocate-hosts --instance-type "m4.large" --availability-zone "eu-west-1a" --  
auto-placement "off" --quantity 1
```

- [New-EC2Host \(AWS Tools for Windows PowerShell\)](#)

```
PS C:\> New-EC2Host -InstanceType m4.large -AvailabilityZone eu-west-1a -  
AutoPlacement Off -Quantity 1
```

The Dedicated Host capacity is made available in your account immediately.

If you launch instances with host tenancy but do not have any active Dedicated Host in your account, you receive an error and the instance launch fails.

Launching Instances onto Dedicated Hosts

After you have allocated a Dedicated Host, you can launch instances onto it. You cannot launch instances with host tenancy if you do not have active Dedicated Hosts with enough available capacity for the instance type 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 Limitations and Restrictions \(p. 334\)](#).

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, choose **Actions**, and then choose **Launch Instance(s) onto Host**.
4. Select an AMI from the list. Windows, SUSE, and RHEL AMIs provided by Amazon EC2 can't be used with Dedicated Hosts.
5. On the **Choose an Instance Type** page, keep the instance type that is selected by default, and then choose **Next: Configure Instance Details**.

The instance type is determined by the host you have selected.

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 \(p. 335\)](#).

Note

The **Tenancy** and **Host** options are pre-configured based on the host 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**, and then choose **Launch Instance**.
3. Select an AMI from the list. Windows, 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 Dedicated Host-specific settings:
 - 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. Dedicated Hosts will be disabled in the list if they do not 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 about auto-placement and Affinity, see [Understanding Auto-Placement and Affinity \(p. 335\)](#).

Note

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**.

To launch an instance onto a Dedicated Host using the command line tools

Use one of the following commands and specify the instance affinity, tenancy, and host in the **Placement** request parameter:

- [run-instances](#) (AWS CLI)
- [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

Modifying Dedicated Host Auto-Placement

You can modify a Dedicated Host's auto-placement settings after you have allocated it to your AWS account.

To modify a Dedicated Host's auto-placement using the Amazon EC2 console

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, choose **Actions**, and then choose **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 about auto-placement, see [Understanding Auto-Placement and Affinity \(p. 335\)](#).
5. Choose **Save**.

To modify a Dedicated Host's auto-placement using the command line tools

Use one of the following commands. The following examples enable auto-placement for the specified Dedicated Host.

- [modify-hosts](#) (AWS CLI)

```
aws ec2 modify-hosts --auto-placement on --host-ids h-012a3456b7890cdef
```

- [Edit-EC2Host](#) (AWS Tools for Windows PowerShell)

```
PS C:\> Edit-EC2Host --AutoPlacement 1 --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've launched it.

To modify instance tenancy and affinity using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Instances**, then 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 \(p. 345\)](#).
 - 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 may not have available, compatible Dedicated Hosts in your account.

For more information about affinity, see [Understanding Auto-Placement and Affinity \(p. 335\)](#).

6. Choose **Save**.

To modify instance tenancy and affinity using the command line tools

Use one of the following commands. The following examples change the specified instance's affinity from default to host and specifies the Dedicated Host that the instance has affinity with.

- [modify-instance-placement](#) (AWS CLI)

```
aws ec2 modify-instance-placement --instance-id i-1234567890abcdef0 --affinity host --host-id h-012a3456b7890cdef
```

- [Edit-EC2InstancePlacement](#) (AWS Tools for Windows PowerShell)

```
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.

To view details of instances on a Dedicated Host using the Amazon EC2 console

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 host to view more information about.
4. For information about the host, choose **Description**. For information about instances running on the host, choose **Instances**.

To view details of instances on a Dedicated Host using the command line tools

Use one of the following commands:

- [describe-hosts](#) (AWS CLI)

```
aws ec2 describe-hosts --host-id host_id
```

- [Get-EC2Host](#) (AWS Tools for Windows PowerShell)

```
PS C:\> Get-EC2Host -HostId host_id
```

Tagging Dedicated Hosts

You can assign custom tags to your 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've assigned. Dedicated Host tags can also be used for cost allocation tracking.

You can only tag a Dedicated Host using the command line tools.

To tag a Dedicated Host using the command line

Use one of the following commands:

- [create-tags](#) (AWS CLI)

The following command tags the specified Dedicated Host with `Owner=TeamA`.

```
aws ec2 create-tags --resources h-abc12345678909876 --tags Key=Owner,Value=TeamA
```

- [New-EC2Tag](#) (AWS Tools for Windows PowerShell)

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 also obtain information about your Dedicated Hosts using the command line tools.

To view the state of a 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. Locate the Dedicated Host in the list and review the value in the **State** column.

To view the state of a Dedicated Host using the command line tools

Use one of the following commands and then review the `state` property in the `hostSet` response element:

- [describe-hosts](#) (AWS CLI)

```
aws ec2 describe-hosts --host-id host_id
```

- [Get-EC2Host](#) (AWS Tools for Windows PowerShell)

```
PS C:\> Get-EC2Host -HostId host_id
```

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 cannot 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 cannot be launched onto a Dedicated Host in this state.
permanent-failure	An unrecoverable failure has been detected. You receive an eviction notice through your instances and by email. Your instances may continue to run. If you stop or terminate all instances on a Dedicated Host with this state, AWS retires the host. Instances cannot 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 need to 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.

To release a Dedicated Host using the Amazon EC2 console

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.

To release a Dedicated Host using the command line tools

Use one of the following commands:

- **release-hosts** (AWS CLI)

```
aws ec2 release-hosts --host-ids host_id
```

- **Remove-EC2Hosts** (AWS Tools for Windows PowerShell)

```
PS C:\> Remove-EC2Hosts -HostId host_id
```

After you release a Dedicated Host, you cannot reuse the same host or host ID again, and you are no longer charged On-Demand billing rates for it. The Dedicated Host's state is changed to `released` and you are not able to launch any instances onto that host.

Note

If you've recently released Dedicated Hosts, it may take some time for them to stop counting towards your limit. During this time, you may 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 console or the Amazon EC2 console or command line tools.

To purchase reservations using the Amazon EC2 console

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 assigned to a reservation.
 - **Availability Zone**—The Availability Zone of the Dedicated Hosts in your account that aren't assigned to a reservation.
 - **Payment option**—The payment option for the offering.

- **Term**—The term of the reservation. 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 choose **Review**.
 6. Review your order and choose **Purchase**.

To purchase reservations using the command line tools

1. Use one of the following commands 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 31536000 seconds, and a three-year term includes 94608000 seconds.

- [describe-host-reservation-offerings \(AWS CLI\)](#)

```
aws ec2 describe-host-reservation-offerings --filter Name=instance-family,Values=m4  
--max-duration 31536000
```

- [Get-EC2HostReservationOffering \(AWS Tools for Windows PowerShell\)](#)

```
PS C:\> $filter = @{Name="instance-family"; Value="m4"}
```

```
PS C:\> Get-EC2HostReservationOffering -filter $filter -MaxDuration 31536000
```

Both commands return a list of offerings that match your criteria. Note the `offeringId` of the offering to purchase.

2. Use one of the following commands to purchase the offering and provide the `offeringId` noted in the previous step. The following examples purchase the specified reservation and associate it with a specific Dedicated Host already allocated in the AWS account.

- [purchase-host-reservation \(AWS CLI\)](#)

```
aws ec2 purchase-host-reservation --offering-id hro-03f707bf363b6b324 --host-id-  
set h-013abcd2a00cbd123
```

- [New-EC2HostReservation \(AWS Tools for Windows PowerShell\)](#)

```
PS C:\> New-EC2HostReservation -OfferingId hro-03f707bf363b6b324 -  
HostIdSet h-013abcd2a00cbd123
```

Viewing Dedicated Host Reservations

You can view information about the Dedicated Hosts associated with your reservation, the term of the reservation, the payment option selected, and the start and end dates of the reservation.

To view details of reservations using the Amazon EC2 console

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**.

4. Choose the reservation from the list provided.
5. Choose **Details** for information about the reservation.
6. Choose **Hosts** for information about the Dedicated Hosts the reservation is associated with.

To view details of reservations using the command line tools

Use one of the following commands:

- [describe-host-reservations](#) (AWS CLI)

```
aws ec2 describe-host-reservations
```

- [Get-EC2HostReservation](#) (AWS Tools for Windows PowerShell)

```
PS C:\> Get-EC2HostReservation
```

Tracking Configuration Changes

You can use AWS Config to record configuration changes for Dedicated Hosts, and 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](#).

To activate AWS Config using the command line or API

- Using the AWS CLI, see [Viewing Configuration Details in the AWS Config Console](#) in the *AWS Config Developer Guide*.
- Using the Amazon EC2 API, see [GetResourceConfigHistory](#).

For more information, see [Viewing Configuration Details in the AWS Config Console](#).

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 the hardware level. In addition, Dedicated Instances that belong to AWS accounts that are linked to a single payer account are also physically isolated at the hardware level. 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 \(p. 332\)](#).

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 \(p. 349\)](#).

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 to dedicated.

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 \(p. 240\)](#).

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. However, if you purchase a Reserved Instance with a default tenancy value, you won't get a Dedicated Reserved Instance if you launch an instance with dedicated instance 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 \(p. 426\)](#).

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 \(p. 291\)](#).

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](#).

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 \(p. 347\)](#)
- [Launching Dedicated Instances into a VPC \(p. 348\)](#)
- [Displaying Tenancy Information \(p. 348\)](#)
- [Changing the Tenancy of an Instance \(p. 349\)](#)
- [Changing the Tenancy of a VPC \(p. 350\)](#)

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 Dedicated Hosts \(p. 337\)](#).

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 \(p. 335\)](#). 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 instance tenancy attribute of a VPC to dedicated.

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

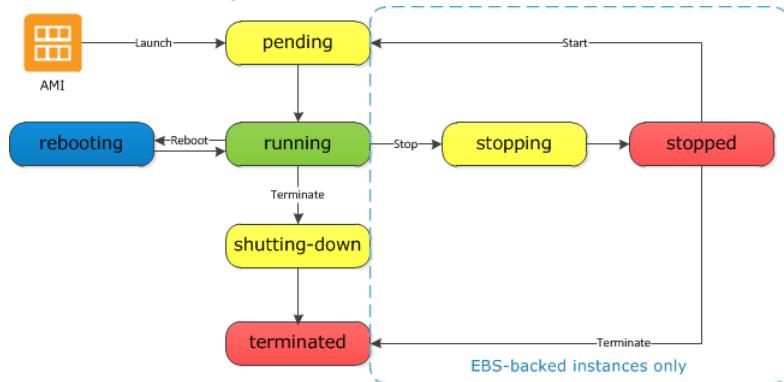
- Use the [modify-vpc-tenancy](#) command to specify the ID of the VPC and instance tenancy value. The only supported value is default.

```
aws ec2 modify-vpc-tenancy --vpc-id vpc-1a2b3c4d --instance-tenancy default
```

Instance Lifecycle

By working with Amazon EC2 to manage your instances from the moment you launch them through their termination, you ensure that your customers have the best possible experience with the applications or sites that you host on your instances.

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 \(p. 84\)](#).



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 <code>running</code> state. An instance enters the <code>pending</code> state when it launches for the first time, or when it is restarted after being in the <code>stopped</code> state.	Not billed
running	The instance is running and ready for use.	Billed
stopping	The instance is preparing to be stopped.	Not billed
stopped	The instance is shut down and cannot be used. The instance can be restarted 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 restarted.	Not billed

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 \(p. 354\)](#) and [Connect to Your Linux Instance \(p. 397\)](#).

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 in most cases, we move the instance to a new host computer. (Your instance may stay on the same host computer if there are no problems with the host computer.) When you stop and start your instance, you lose any data on the instance store volumes on the previous host computer.

If your instance is running in EC2-Classic, it receives a new private IPv4 address, which means that an Elastic IP address associated with the private IPv4 address is no longer associated with your instance. If your instance is running in EC2-VPC, it 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 restart your instance.

For more information, see [Stop and Start Your Instance \(p. 415\)](#).

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 \(p. 418\)](#).

Instance Retirement

An instance is scheduled to be retired when AWS detects 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 \(p. 419\)](#).

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 \(p. 421\)](#).

Differences Between Reboot, Stop, and Terminate

The following table summarizes the key differences between rebooting, stopping, and terminating your instance.

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Terminate
Host computer	The instance stays on the same host computer	In most cases, we move the instance to a new host computer. Your instance may stay on the same host computer if there are no problems with the host computer.	None
Private and public IPv4 addresses	These addresses stay the same	EC2-Classic: The instance gets new private and public IPv4 addresses EC2-VPC: The instance keeps its private IPv4 address. The instance gets a new public IPv4 address, unless it has an Elastic IP address, which doesn't change during a stop/start.	None
Elastic IP addresses (IPv4)	The Elastic IP address remains associated with the instance	EC2-Classic: The Elastic IP address is disassociated from the instance	The Elastic IP address is disassociated from the instance

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Terminate
		EC2-VPC: The Elastic IP address remains associated with the instance	
IPv6 address (EC2-VPC only)	The address stays the same	The instance keeps its IPv6 address	None
Instance store volumes	The data is preserved	The data is erased	The data is erased
Root device volume	The volume is preserved	The volume is preserved	The volume is deleted by default
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 restart 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 \(p. 423\)](#).

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 micro instance for free for 12 months. 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. 355)
[Amazon EC2 console] Create a launch template and launch the instance from the launch template	Launching an Instance from a Launch Template (p. 361)
[Amazon EC2 console] Use an existing instance as the base	Launching an Instance Using Parameters from an Existing Instance (p. 370)

Method	Documentation
[Amazon EC2 console] Use an Amazon EBS snapshot that you created	Launching a Linux Instance from a Backup (p. 371)
[Amazon EC2 console] Use an AMI that you purchased from the AWS Marketplace	Launching an AWS Marketplace Instance (p. 372)
[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 an EC2 Fleet (p. 374)

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. 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 Amazon EC2 network (EC2-Classic or EC2-VPC) can use to contact the instance. For more information about connecting to your instance, see [Connect to Your Linux Instance \(p. 397\)](#).

When you are finished with an instance, be sure to terminate it. For more information, see [Terminate Your Instance \(p. 421\)](#).

Launching an Instance Using the Launch Instance Wizard

Before you launch your instance, be sure that you are set up. For more information, see [Setting Up with Amazon EC2 \(p. 19\)](#).

Your AWS account might support both the EC2-Classic and EC2-VPC platforms, depending on when you created your account and which regions you've used. To find out which platform your account supports, see [Supported Platforms \(p. 637\)](#). If your account supports EC2-Classic, you can launch an instance into either platform. If your account supports EC2-VPC only, you can launch an instance into a VPC only.

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.

Launching Your Instance from an 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: for example, Linux, Apache, and your website.

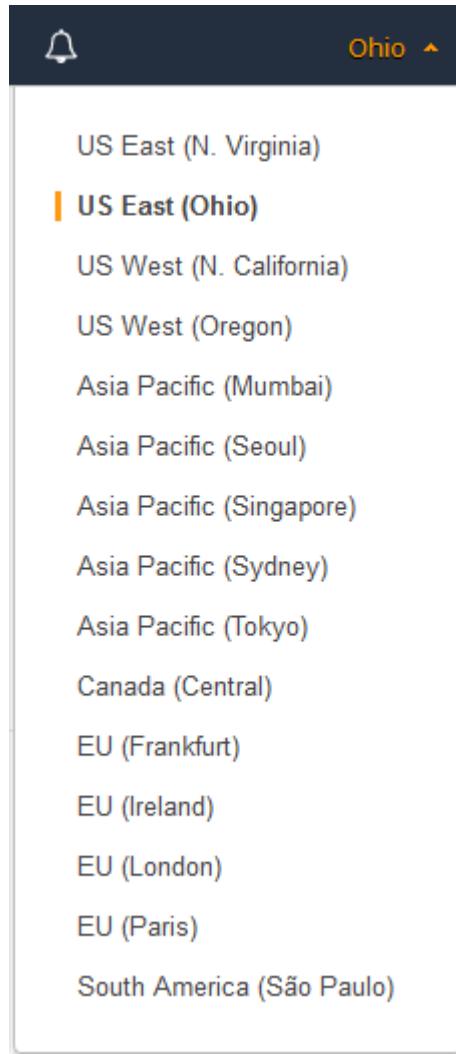
Tip

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.

To launch an instance

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. 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 \(p. 878\)](#).



3. From the Amazon EC2 console dashboard, choose **Launch Instance**.
4. On the **Choose an Amazon Machine Image (AMI)** page, choose an AMI as follows:
 - a. 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 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 \(p. 372\)](#).

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.

- b. 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 \(p. 84\)](#).
 - c. 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 \(p. 86\)](#).
 - d. Choose an AMI that meets your needs, and then choose **Select**.
5. 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 \(p. 163\)](#).

To remain eligible for the free tier, choose the **t2.micro** instance type. For more information, see [T2 Instances \(p. 168\)](#).

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**.

6. 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.

Note

To help ensure that you maintain the correct number of instances to handle 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:** Select **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 \(p. 292\)](#).
- Your account may support the EC2-Classic and EC2-VPC platforms, or EC2-VPC only. To find out which platform your account supports, see [Supported Platforms \(p. 637\)](#). If your account supports EC2-VPC only, you can launch your instance into your default VPC or a nondefault VPC. Otherwise, you can launch your instance into EC2-Classic or a nondefault VPC.

Note

Some instance types must be launched into a VPC. If you don't have a VPC, you can let the wizard create one for you.

To launch into EC2-Classic:

- **Network:** Select **Launch into EC2-Classic**.
- **Availability Zone:** Select the Availability Zone to use. To let AWS choose an Availability Zone for you, select **No preference**.

To launch into a VPC:

- **Network:** Select the VPC, or to create a new VPC, choose **Create new VPC** to go the Amazon VPC console. When you have finished, return to the wizard and choose **Refresh** to load your VPC in the list.
- **Subnet:** Select the subnet into which to launch your instance. If your account is EC2-VPC only, 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.
- **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 \(p. 660\)](#).
- **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*.
- **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 \(p. 620\)](#).
- **Shutdown behavior:** Select whether the instance should stop or terminate when shut down. For more information, see [Changing the Instance Initiated Shutdown Behavior \(p. 423\)](#).
- **Enable termination protection:** To prevent accidental termination, select this check box. For more information, see [Enabling Termination Protection for an Instance \(p. 422\)](#).
- **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 \(p. 483\)](#).
- **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 \(p. 812\)](#).
- **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 \(p. 345\)](#) and [Dedicated Hosts \(p. 332\)](#).
- **T2 Unlimited:** (Only valid for T2 instances) 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 [T2 Instances \(p. 168\)](#).
- **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 \(p. 683\)](#). 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 \(p. 664\)](#).

- **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.
 - **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 \(p. 701\)](#).
 - **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.
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**. You can configure the following options for each volume:
- **Type:** Select instance store or Amazon EBS volumes to associate with your instance. The type of volume available in the list depends on the instance type you've chosen. For more information, see [Amazon EC2 Instance Store \(p. 847\)](#) and [Amazon EBS Volumes \(p. 733\)](#).
 - **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 public snapshots by typing text into the **Snapshot** field. Snapshot descriptions are case-sensitive.
 - **Size:** For Amazon EBS-backed 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 keep under 30 GiB of total storage.

Note

Linux AMIs require GPT partition tables and GRUB 2 for boot volumes 2 TiB (2048 GiB) or larger. Many Linux AMIs today use the MBR partitioning scheme, which only supports up to 2047 GiB boot volumes. If your instance does not boot with a boot volume that is 2 TiB or larger, the AMI you are using may be limited to a 2047 GiB boot volume size. Non-boot volumes do not have this limitation on Linux instances.

Note

If you increase the size of your root volume at this point (or any other volume created from a snapshot), you need to extend the file system on that volume in order to use the extra space. For more information about extending your file system after your instance has launched, see [Modifying the Size, IOPS, or Type of an EBS Volume on Linux \(p. 769\)](#).

- **Volume Type:** For Amazon EBS volumes, select either a General Purpose SSD, Provisioned IOPS SSD, or Magnetic volume. For more information, see [Amazon EBS Volume Types \(p. 735\)](#).

Note

If you select a Magnetic boot volume, you'll be prompted when you complete the wizard to make General Purpose SSD volumes the default boot volume for this instance and future console launches. (This preference persists in the browser session, and does not affect AMIs with Provisioned IOPS SSD boot volumes.) We recommend that you make General Purpose SSD volumes the default because they provide a much faster

boot experience and they are the optimal volume type for most workloads. For more information, see [Amazon EBS Volume Types \(p. 735\)](#).

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.

- **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 \(p. 424\)](#).
- **Encrypted:** Select a value in this menu to configure the encryption state of new Amazon EBS volumes. The default value is **Not encrypted**. Additional options include using your AWS managed customer master key (CMK) or a customer-managed CMK that you have created. Available keys are listed in the menu. You can also hover over the field and paste the Amazon Resource Name (ARN) of a key directly into the text box. For information about creating customer-managed CMKs, see [AWS Key Management Service Developer Guide](#).

Note

Encrypted volumes may only be attached to [supported instance types \(p. 820\)](#).

When done configuring your volumes, choose **Next: Add Tags**.

8. On the **Add Tags** page, specify [tags \(p. 887\)](#) 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.
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 \(p. 531\)](#).) Select or create a security group as follows, and then choose **Review and Launch**.
 - a. To select an existing security group, choose **Select an existing security group**, and select your security group. If you are launching into EC2-Classic, the security groups are for EC2-Classic. If you are launching into a VPC, the security groups are for that VPC.

Note

(Optional) 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.

- b. 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).
- c. 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.

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**.

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 you created when getting set up.

To launch your instance, select the acknowledgment check box, then choose **Launch Instances**.

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.

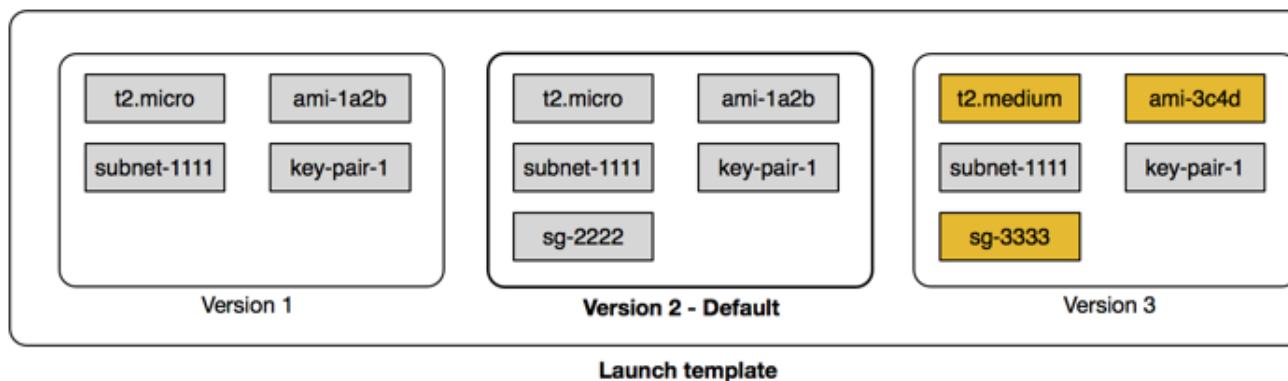
12. (Optional) You can create a status check alarm for the instance (additional fees may apply). (If you're not sure, you can always add one later.) On the confirmation screen, choose **Create status check alarms** and follow the directions. For more information, see [Creating and Editing Status Check Alarms \(p. 477\)](#).
13. If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see [Troubleshooting Instance Launch Issues \(p. 907\)](#).

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 \(p. 362\)](#)
- [Using Launch Templates to Control Launch Parameters \(p. 362\)](#)
- [Controlling the Use of Launch Templates \(p. 363\)](#)
- [Creating a Launch Template \(p. 363\)](#)
- [Managing Launch Template Versions \(p. 367\)](#)
- [Launching an Instance from a Launch Template \(p. 368\)](#)
- [Using Launch Templates with Amazon EC2 Auto Scaling \(p. 369\)](#)
- [Using Launch Templates with Spot Fleet \(p. 370\)](#)
- [Deleting a Launch Template \(p. 370\)](#)

Launch Template Restrictions

The following rules apply to launch templates and launch template versions:

- You are limited to creating 1,000 launch templates per region and 10,000 versions per launch template.
- Launch 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 validated when you create the 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 permission to use the `ec2:RunInstances` action, and they must have permission 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, or you can grant users permission to launch an instance using a launch template instead. 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 \(p. 603\)](#).

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 [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 557\)](#) and the following example policies: [13. Working with Launch Templates \(p. 612\)](#).

Take care when granting users permissions to use the `ec2:CreateLaunchTemplate` and `ec2:CreateLaunchTemplateVersion` actions. These actions do not support resource-level permissions that enable you 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

You can create a new launch template using parameters that you define, or you can use an existing instance as the basis for a new launch template.

To create a new 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 and description for the launch template.
4. For **Launch template contents**, provide the following information.
 - **AMI ID:** Specify an AMI ID from which to launch the instance. You can use an AMI that you own, or you can [find a suitable AMI \(p. 87\)](#).
 - **Instance type:** Choose the instance type. Ensure that the instance type is compatible with the AMI you've specified. For more information, see [Instance Types \(p. 163\)](#).
 - **Key pair name:** Specify the key pair for the instance. For more information, see [Amazon EC2 Key Pairs \(p. 521\)](#).
 - **Network type:** If applicable, choose whether to launch the instance into EC2-Classic or a VPC. This option is not available if your account supports EC2-VPC only. If you choose EC2-Classic, ensure that the specified instance type is supported in EC2-Classic and specify the Availability Zone for the instance. If you choose EC2-VPC, specify the subnet in the **Network interfaces** section.
5. For **Network interfaces**, you can specify up to two [network interfaces \(p. 683\)](#) for the instance.
 - **Device:** Specify 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:** Specify the ID of the network interface or leave blank to let AWS create a new network interface.
 - **Description:** Optionally enter a description for a new network interface.
 - **Subnet:** Specify 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 specified an existing

network interface for eth0, the instance is launched in the subnet in which the network interface is located.

- **Auto-assign public IP:** Specify 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:** Enter a private IPv4 address from the range of your subnet, or leave blank to let AWS choose a private IPv4 address for you.
 - **Secondary IP:** Enter a secondary private IPv4 address from the range of your subnet, or leave blank to let AWS choose one for you.
 - **(IPv6-only) IPv6 IPs:** Enter an IPv6 address from the range of the subnet.
 - **Security group ID:** Enter the ID of a security group in your VPC with which to associate the network interface.
 - **Delete on termination:** Choose whether the network interface is deleted when the instance is deleted.
6. For **Storage (Volumes)**, specify volumes to attach to the instance besides the volumes specified by the AMI.
- **Volume type:** Specify 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 \(p. 847\)](#) and [Amazon EBS Volumes \(p. 733\)](#).
 - **Device name:** Specify a device name for the volume.
 - **Snapshot:** Enter the ID of the snapshot from which to create the volume.
 - **Size:** For Amazon EBS-backed volumes, specify a storage size.
 - **Volume type:** For Amazon EBS volumes, the volume type. For more information, see [Amazon EBS Volume Types \(p. 735\)](#).
 - **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 \(p. 424\)](#).
 - **Encrypted:** Select this check box to encrypt new Amazon EBS volumes. Amazon EBS volumes that are restored from encrypted snapshots are automatically encrypted. Encrypted volumes may only be attached to [supported instance types \(p. 820\)](#).
7. For **Tags**, specify [tags \(p. 887\)](#) by providing key and value combinations. You can tag the instance, the volumes, or both.
8. For **Security groups**, specify one or more security groups to associate with the instance. For more information, see [Amazon EC2 Security Groups for Linux Instances \(p. 531\)](#).
9. For **Advanced Details**, expand the section to view the fields and specify any additional parameters for the instance.
- **IAM instance profile:** Specify an AWS Identity and Access Management (IAM) instance profile to associate with the instance. For more information, see [IAM Roles for Amazon EC2 \(p. 620\)](#).
 - **Shutdown behavior:** Select whether the instance should stop or terminate when shut down. For more information, see [Changing the Instance Initiated Shutdown Behavior \(p. 423\)](#).
 - **Termination protection:** Select whether to prevent accidental termination. For more information, see [Enabling Termination Protection for an Instance \(p. 422\)](#).
 - **Monitoring:** Select whether to enable detailed monitoring of the instance using Amazon CloudWatch. Additional charges apply. For more information, see [Monitoring Your Instances Using CloudWatch \(p. 483\)](#).
 - **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 \(p. 701\)](#).

- **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 \(p. 812\)](#).
- **Tenancy:** Specify whether 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 \(p. 345\)](#) and [Dedicated Hosts \(p. 332\)](#). If you specify a Dedicated Host, you can choose a specific host and the affinity for the instance.
- **RAM disk ID:** A RAM disk for the instance. If you have selected a kernel, you may need to select 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 \(p. 448\)](#).

10. Choose **Create launch template**.

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 and description 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 choose **Create launch template**.

To create a launch template using the command line

- Use the `create-launch-template` (AWS CLI) command. The following example creates a launch template that specifies the instance type (`r4.4xlarge`) and AMI (`ami-8c1be5f6`) to launch, specifies the number of cores (4) and threads per core (2) for a total of 8 vCPUs (4 cores x 2 threads), specifies the subnet in which to launch the instance (`subnet-7b16de0c`), 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 --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"  
    }  
}
```

To get instance data for a launch template using the command line

- Use the [get-launch-template-data](#) (AWS CLI) 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",  
            "Status": "in-use",  
            "MacAddress": "0A:1B:2C:3D:4E:5F",  
            "OwnerId": "123456789012",  
            "PrivateDnsName": "ip-10-0-0-72.us-east-1.compute.internal",  
            "PrivateIpAddress": "10.0.0.72",  
            "Association": {  
                "AllocationId": "alloc-12345678901234567890123456789012",  
                "AssociationTime": "2017-11-27T09:13:24.000Z",  
                "PublicDnsName": "ip-10-0-0-72.us-east-1.compute.amazonaws.com",  
                "PublicIpAddress": "10.0.0.72",  
                "Region": "us-east-1",  
                "RouteTableId": "rtb-12345678901234567890123456789012",  
                "State": "associated",  
                "VpcId": "vpc-12345678901234567890123456789012"  
            }  
        }  
    ]  
}
```

```
        "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-0123d646e8048bab --query
"LaunchTemplateData" >> instance-data.json
```

Managing Launch Template Versions

You can create launch template versions for a specific launch template, set the default version, and delete versions that you no longer require.

Tasks

- [Creating a Launch Template Version \(p. 367\)](#)
- [Setting the Default Launch Template Version \(p. 368\)](#)
- [Deleting a Launch Template Version \(p. 368\)](#)

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 \(p. 363\)](#).

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**, type 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**.

To create a launch template version using the command line

- Use the `create-launch-template-version` (AWS CLI) 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.

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**.

To set the default launch template version using the command line

- Use the [modify-launch-template](#) (AWS CLI) 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
```

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.

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**.

To delete a launch template version using the command line

- Use the [delete-launch-template-versions](#) (AWS CLI) 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.

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**.

To launch an instance from a launch template using the command line

- Use the `run-instances` AWS CLI 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 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 \(p. 907\)](#).

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*.

To create or update an Amazon EC2 Auto Scaling group with a launch template using the command line

- Use the [create-auto-scaling-group](#) or the [update-auto-scaling-group](#) AWS CLI command and specify the --launch-template parameter.

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 \(p. 297\)](#).

To create a Spot Fleet request with a launch template using the command line

- Use the [request-spot-fleet](#) AWS CLI 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.

To delete a launch template

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**.

To delete a launch template using the command line

- 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 \(p. 361\)](#) 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 \(p. 660\)](#).
- 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:

- (VPC only) 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. On the Instances page, select the instance you want to use.
2. Choose **Actions**, and then **Launch More Like This**.
3. 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.
4. If the instance fails to launch or the state immediately goes to terminated instead of running, see [Troubleshooting Instance Launch Issues \(p. 907\)](#).

Launching a Linux Instance from a Backup

With an Amazon EBS-backed Linux instance, you can back up the root device volume of the instance by creating a snapshot. When you have a snapshot of the root device volume of an instance, you can terminate that instance and then later launch a new instance from the snapshot. This can be useful if you don't have the original AMI that you launched an instance from, but you need to be able to launch an instance using the same image.

Important

Although you can create a Windows AMI from a snapshot, you can't successfully launch an instance from that AMI.

Some Linux distributions, such as Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES), use the billing product code associated with an AMI to verify subscription status for package updates. Creating an AMI from an EBS snapshot does not maintain this billing code, and subsequent instances launched from such an AMI are not able to connect to the package update infrastructure. To retain the billing product codes, create the AMI from the instance not from a snapshot. For more information, see [Creating an Amazon EBS-Backed Linux AMI \(p. 101\)](#) or [Creating an Instance Store-Backed Linux AMI \(p. 105\)](#).

Use the following procedure to create an AMI from the root volume of your instance using the console. If you prefer, you can use one of the following commands instead: [register-image](#) (AWS CLI) or [Register-EC2Image](#) (AWS Tools for Windows PowerShell). You specify the snapshot using the block device mapping.

To create an AMI from your root volume 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, Snapshots**.
3. Choose **Create Snapshot**.
4. For **Volumes**, start typing the name or ID of the root volume, and then select it from the list of options.
5. Choose the snapshot that you just created, and then choose **Actions, Create Image**.
6. In the **Create Image from EBS Snapshot** dialog box, provide the following information and 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 \(p. 864\)](#).
 - **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 \(p. 86\)](#).
 - (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 are required to 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 the Volume \(p. 787\)](#).
7. In the navigation pane, choose **AMIs**.
8. Choose the AMI that you just created, and then choose **Launch**. Follow the wizard to launch your instance. For more information about how to configure each step in the wizard, see [Launching an Instance Using the Launch Instance Wizard \(p. 355\)](#).

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 \(p. 98\)](#). 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 \(p. 101\)](#).

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 \(p. 355\)](#). 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 \(p. 397\)](#).

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 \(p. 101\)](#).

10. If the instance fails to launch or the state immediately goes to **terminated** instead of **running**, see [Troubleshooting Instance Launch Issues \(p. 907\)](#).

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:

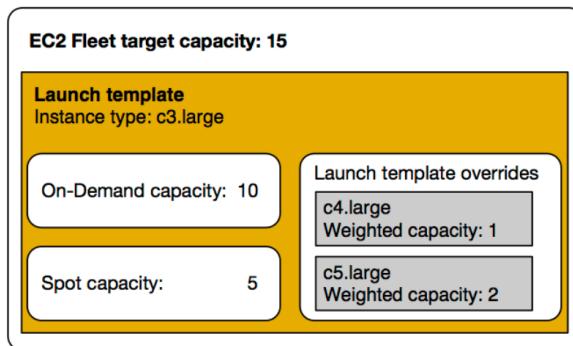
Method	Documentation
AWS CLI	Use the run-instances command, or see the following topic for more information: Launching an Instance .

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 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 models together. Using EC2 Fleet, you can define separate On-Demand and Spot capacity targets, specify the instance types that work best for your applications, and specify how Amazon EC2 should distribute your fleet capacity within each purchasing model.

The EC2 Fleet attempts to launch the number of instances that are required to meet the target capacity specified in your request. The fleet also attempts to maintain its target Spot capacity if your Spot Instances are interrupted due to a change in Spot prices or available capacity. For more information, see [How Spot Instances Work \(p. 281\)](#).



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 \(p. 375\)](#)
- [EC2 Fleet Configuration Strategies \(p. 375\)](#)

- [Managing an EC2 Fleet \(p. 382\)](#)

EC2 Fleet Limitations

The following limitations apply to EC2 Fleet:

- EC2 Fleet is available only through the API or AWS CLI.
- You need to create a separate EC2 Fleet for each region.

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 On-Demand Instances and Spot Instances to meet the specified target capacity. The request for Spot Instances is fulfilled if the specified Spot price exceeds the current Spot price and there is available capacity. The fleet also attempts to maintain its target capacity if your Spot Instances are interrupted due to a change in Spot prices or available capacity.

A *Spot Instance pool* is a set of unused EC2 instances with the same instance type, operating system, Availability Zone, and network platform (EC2-Classic or EC2-VPC). 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. Or you can request 1000 cores with a minimum of 2 GB of RAM per core. The fleet determines the combination of Amazon EC2 options to launch that capacity at the absolute lowest cost.

Use the appropriate configuration strategies to create an EC2 Fleet that meets your needs.

Contents

- [Planning an EC2 Fleet \(p. 375\)](#)
- [EC2 Fleet Request Types \(p. 376\)](#)
- [Allocation Strategy for Spot Instances \(p. 376\)](#)
- [Configuring EC2 Fleet for On-Demand Backup \(p. 377\)](#)
- [Maximum Price Overrides \(p. 378\)](#)
- [EC2 Fleet Instance Weighting \(p. 378\)](#)
- [Walkthrough: Using EC2 Fleet with Instance Weighting \(p. 379\)](#)
- [Walkthrough: Using EC2 Fleet with On-Demand as the Primary Capacity \(p. 381\)](#)

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 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 \(p. 376\)](#).
- 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 \(p. 378\)](#).
- 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.
- Review the possible options for your EC2 Fleet. For more information, see the [EC2 Fleet JSON Configuration File Reference \(p. 387\)](#). For EC2 Fleet configuration examples, see [EC2 Fleet Example Configurations \(p. 394\)](#).

EC2 Fleet Request Types

There are two types of EC2 Fleet requests: `request` and `maintain`. You can create a fleet to submit a one-time request for your desired capacity, or require it to maintain the target capacity over time.

If you configure the fleet type as `request`, EC2 Fleet places a one-time request for the target capacity, and does not attempt to replenish Spot Instances if capacity is diminished. If capacity is not available, the fleet does not submit requests in alternative Spot Instance pools.

If you configure the fleet type as `maintain`, EC2 Fleet places requests to meet the target capacity, and automatically replenishes any interrupted Spot Instances.

You cannot modify the target capacity of a one-time request after it's been submitted. To change the target capacity of a one-time request, delete the EC2 Fleet and create a new one.

Both types of requests benefit from an allocation strategy. For more information, see [Allocation Strategy for Spot Instances \(p. 376\)](#).

Allocation Strategy 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:

`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.

`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, an EC2 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 EC2 Fleet for Cost Optimization

To optimize the costs for your use of Spot Instances, specify the `lowestPrice` allocation strategy so that EC2 Fleet automatically deploys the cheapest 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 `lowestPrice` 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 `lowestPrice` allocation strategy in combination with `InstancePoolsToUseCount`. EC2 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.

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 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 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 `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.

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 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 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`.

Maximum Price Overrides

Each EC2 Fleet can 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.

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, and adjust your maximum price for each launch specification 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. 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
<code>r3.xlarge</code>	2	10	5 (10 divided by 2)	\$0.05	.025 (.05 divided by 2)
<code>r3.8xlarge</code>	8	10	2 (10 divided by 8, result rounded up)	\$0.10	.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 `lowestPrice` 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.

Walkthrough: Using EC2 Fleet with Instance Weighting

This walkthrough 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

Instance type	Memory (GiB)	vCPUs
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 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 \(p. 384\)](#).

Creating the EC2 Fleet

Example Corp creates a file, config.json, with the following configuration for its EC2 Fleet:

```
{  
    "LaunchTemplateConfigs": [  
        {  
            "LaunchTemplateSpecification": {  
                "LaunchTemplateId": "lt-07b3bc7625cdab851",  
                "LaunchTemplateName": "my-launch-template",  
                "LaunchTemplateVersion": 1  
            }  
        }  
    ]  
}
```

```
        "Version": "1"
    },
    "Overrides": [
        {
            "InstanceType": "r3.2xlarge",
            "SubnetId": "subnet-482e4972",
            "WeightedCapacity": 1
        }
        {
            "InstanceType": "r3.4xlarge",
            "SubnetId": "subnet-482e4972",
            "WeightedCapacity": 2
        }
        {
            "InstanceType": "r3.8xlarge",
            "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 \(p. 389\)](#).

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 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.

Walkthrough: Using EC2 Fleet with On-Demand as the Primary Capacity

This walkthrough 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 purchase models 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 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 \(p. 384\)](#).

Creating the EC2 Fleet

ABC Online creates a file, `config.json`, with the following configuration for its EC2 Fleet:

```
{  
    "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 \(p. 389\)](#).

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 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 attempts 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 may specify whether deletion terminates the instances in that fleet.

Contents

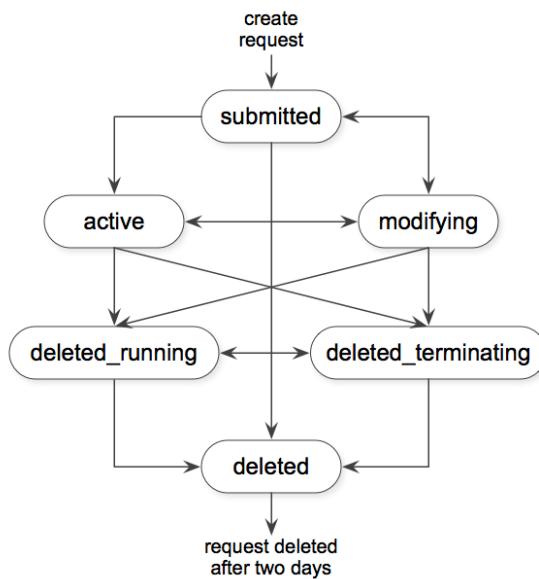
- [EC2 Fleet Request States \(p. 383\)](#)
- [EC2 Fleet Prerequisites \(p. 384\)](#)
- [EC2 Fleet Health Checks \(p. 386\)](#)
- [Generating an EC2 Fleet JSON Configuration File \(p. 386\)](#)
- [Creating an EC2 Fleet \(p. 389\)](#)
- [Tagging an EC2 Fleet \(p. 390\)](#)
- [Monitoring Your EC2 Fleet \(p. 382\)](#)
- [Modifying an EC2 Fleet \(p. 392\)](#)
- [Deleting an EC2 Fleet \(p. 393\)](#)
- [EC2 Fleet Example Configurations \(p. 394\)](#)

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. A one-time request cannot be modified, and this state does not apply to such requests.
- **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 \(p. 384\)](#)
- [IAM role for EC2 Fleet \(p. 384\)](#)
- [Service-linked role for EC2 Fleet \(p. 385\)](#)

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 \(p. 361\)](#).

IAM Role for EC2 Fleet

The `aws-ec2-fleet-tagging-role` grants the EC2 Fleet permission to request, launch, terminate, and tag instances on your behalf. Ensure that this role exists before you use the AWS CLI or an API to create an EC2 Fleet. To create the role, use the IAM console as follows.

To create the IAM role for EC2 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 - Fleet Tagging**, and then choose **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-fleet-tagging-role`) and choose **Create role**.

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, Create policy**.
3. 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": [  
                "lambda:InvokeFunction"  
            ],  
            "Resource": "  
        }  
    ]  
}
```

```
        "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`
4. On the **Review policy** page, type a policy name and description, and then choose **Create policy**.
 5. In the navigation pane, choose **Users** and select the user.
 6. On the **Permissions** tab, choose **Add permissions**.
 7. Choose **Attach existing policies directly**. Select the policy that you created earlier and choose **Next: Review**.
 8. Choose **Add permissions**.

Service-Linked Role for EC2 Fleet Requests

Before you create an EC2 Fleet, you need to create a service-linked role named **AWSServiceRoleForEC2Fleet**. A service-linked role includes all the permissions that Amazon EC2 requires to call other AWS services on your behalf. If you try to create a fleet before the role is created, you get an error. Create the role using the IAM console. For more information, see [Using Service-Linked Roles](#) in the *IAM User Guide*.

Amazon EC2 uses the service-linked role named **AWSServiceRoleForEC2Fleet** to complete the following actions:

- `ec2:RequestSpotInstances` – Request Spot Instances.
- `ec2:TerminateInstances` – Terminate Spot Instances.
- `ec2:DescribeImages` – Describe Amazon Machine Images (AMI) for the Spot Instances.
- `ec2:DescribeInstanceStatus` – Describe the status of the Spot Instances.
- `ec2:DescribeSubnets` – Describe the subnets for Spot Instances.
- `ec2>CreateTags` - Add system tags to Spot Instances.

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.

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 \(p. 474\)](#).

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, target capacity, and whether the default purchasing model 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  
    },  
    "OnDemandOptions": {  
        "AllocationStrategy": "prioritized"  
    },  
    "ExcessCapacityTerminationPolicy": "termination",  
    "LaunchTemplateConfigs": [  
        {  
            "LaunchTemplateSpecification": {  
                "LaunchTemplateId": "",  
                "LaunchTemplateName": "",  
                "Version": ""  
            },  
            "Overrides": [  
                {  
                    "InstanceType": "t2.micro",  
                    "Weight": 1  
                }  
            ]  
        }  
    ]  
}
```

```
        "MaxPrice": "",  
        "SubnetId": "",  
        "AvailabilityZone": "",  
        "WeightedCapacity": null,  
        "Priority": null  
    }  
}  
]  
}  
],  
"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` and `diversified`. The default is `lowest-price`. Specify the allocation strategy that meets your needs. For more information, see [Allocation Strategy for Spot Instances \(p. 376\)](#).

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 `lowestPrice`. EC2 Fleet selects the cheapest Spot pools and evenly allocates your target Spot capacity across the number of Spot pools that you specify.

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.

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 more information about creating launch templates, see [Launching an Instance from a Launch Template \(p. 361\)](#).

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 \(p. 361\)](#).

Version

The version number of the launch template.

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. Your account supports either the EC2-Classic and EC2-VPC platforms, or the EC2-VPC platform only. To find out which platforms your account supports, see [Supported Platforms \(p. 637\)](#).

[New VPC] 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.

[EC2-VPC] 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 model 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 EC2 Fleet only requests the target capacity, or also attempts to maintain it. Valid values are `maintain` and `request`. The default value is `maintain`. If the value is `request`, EC2 Fleet only places the required requests. It does not attempt to replenish instances if capacity is diminished, and does not submit requests in alternative capacity pools if capacity is unavailable. If the value is `maintain`, the fleet places the required requests to meet the target capacity. It also automatically replenishes any interrupted Spot Instances.

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 \(p. 363\)](#). For information about tagging after launch, see [Tagging Your Resources \(p. 889\)](#).

[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 \(p. 387\)](#).

To create an EC2 Fleet using the AWS CLI

- Use the following [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 \(p. 394\)](#).

The following is example output:

```
{  
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"  
}
```

Tagging an EC2 Fleet

To help categorize and manage your EC2 Fleet requests, you can tag them with custom metadata. For more information, see [Tagging Your Amazon EC2 Resources \(p. 887\)](#).

You can assign a tag to an EC2 Fleet request when you create it, or afterward. Tags assigned to the fleet request are not assigned to the instances launched by the 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 \(p. 386\)](#) used to create the fleet. The value for `ResourceType` must be `fleet`. If you specify another value, the fleet request fails.

To tag instances launched by an EC2 Fleet

To tag instances when they are launched by the fleet, specify the tags in the [launch template \(p. 363\)](#) referenced in the EC2 Fleet request.

To tag an existing EC2 Fleet request and instance using the AWS CLI

Use the following [create-tags](#) command to tag existing resources:

```
aws ec2 create-tags --resources fleet-12a34b55-67cd-8ef9-  
ba9b-9208dEXAMPLE i-1234567890abcdef0 --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 using the AWS CLI

Use the following [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": "lowestPrice"
            },
            "FleetState": "active",
            "ExcessCapacityTerminationPolicy": "termination",
            "CreateTime": "2018-04-10T16:46:03.000Z"
        }
    ]
}
```

Use the following [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-dvx17fsm"
    },
],
"FleetId": "fleet-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

Use the following [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 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`.

Note

You can only modify an EC2 Fleet that has `Type=maintain`.

When you increase the target capacity, the EC2 Fleet launches the additional instances according to the instance purchasing model 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 `lowestPrice`, 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 `lowestPrice`, 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 using the AWS CLI

Use the following [modify-fleet](#) command to update the target capacity of the specified EC2 Fleet:

```
aws ec2 modify-fleet --fleet-id fleet-73fbcd2ce-aa30-494c-8788-1ceeeEXAMPLE --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-73fbcd2ce-aa30-494c-8788-1ceeeEXAMPLE --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.

To delete an EC2 Fleet using the 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-73fbcd2ce-aa30-494c-8788-1ceeeEXAMPLE --terminate-instances
```

The following is example output:

```
{  
    "UnsuccessfulFleetDeletions": [],  
    "SuccessfulFleetDeletions": [  
        {  
            "CurrentFleetState": "deleted_terminating",  
            "PreviousFleetState": "active",  
            "FleetId": "fleet-73fbcd2ce-aa30-494c-8788-1ceeeEXAMPLE"  
        }  
    ]  
}
```

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-73fbcd2ce-aa30-494c-8788-1ceeeEXAMPLE --no-terminate-instances
```

The following is example output:

```
{  
    "UnsuccessfulFleetDeletions": [],  
    "SuccessfulFleetDeletions": [  
        {  
            "CurrentFleetState": "deleted_running",  
            "PreviousFleetState": "active",  
            "FleetId": "fleet-4b8aaae8-dfb5-436d-a4c6-3dafa4c6b7dcEXAMPLE"  
        }  
    ]  
}
```

```
}
```

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, see the [EC2 Fleet JSON Configuration File Reference \(p. 387\)](#).

1. [Launch Spot Instances as the default purchasing model \(p. 394\)](#)
2. [Launch On-Demand Instances as the default purchasing model \(p. 394\)](#)
3. [Launch On-Demand Instances as the primary capacity \(p. 395\)](#)
4. [Launch Spot Instances using the lowest-price allocation strategy \(p. 395\)](#)

Example 1: Launch Spot Instances as the Default Purchasing Model

The following example specifies the minimum parameters required in an EC2 Fleet: a launch template, target capacity, and default purchasing model. 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 model 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 Model

The following example specifies the minimum parameters required in an EC2 Fleet: a launch template, target capacity, and default purchasing model. 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 model 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 model is spot. The fleet launches 1 On-Demand Instance as specified, but needs to launch one more instance to fulfil the total target capacity. The purchasing model 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 model 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"  
}
```

Connect to Your Linux Instance

Learn how to 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*.

Your Computer	Topic
Linux	Connecting to Your Linux Instance Using SSH (p. 397)
Windows	Connecting to Your Linux Instance from Windows Using PuTTY (p. 402) Connecting to Your Linux Instance from Windows Using Windows Subsystem for Linux (p. 408)
All	Connecting to Your Linux Instance Using MindTerm (p. 414)

After you connect to your instance, you can try one of our tutorials, such as [Tutorial: Install a LAMP Web Server with the Amazon Linux AMI \(p. 41\)](#) or [Tutorial: Hosting a WordPress Blog with Amazon Linux \(p. 51\)](#).

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.

Note

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.

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 \(p. 909\)](#).

Prerequisites

Before you connect to your Linux instance, complete the following prerequisites:

- **Install an SSH client**

Your Linux computer most likely includes an SSH client by default. You can check for an SSH client by typing **ssh** at the command line. If your computer doesn't recognize the command, the OpenSSH project provides a free implementation of the full suite of SSH tools. For more information, see <http://www.openssh.com>.

- **Install the AWS CLI Tools**

(Optional) If you're using a public AMI from a third party, you can use the command line tools to verify the fingerprint. For more information about installing the AWS CLI, see [Getting Set Up](#) in the *AWS Command Line Interface User Guide*.

- **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 **Show/Hide** icon 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 \(p. 661\)](#).

- **Locate the private key and verify permissions**

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. Verify that the `.pem` file has permissions of 0400, not 0777. For more information, see [Error: Unprotected Private Key File \(p. 914\)](#).

- **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` or `root`.
- 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 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 does not allow incoming SSH traffic by default. For more information, see [Authorizing Inbound Traffic for Your Linux Instances \(p. 628\)](#).

Connecting to Your Linux Instance

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 \(p. 909\)](#).

To connect to your instance using SSH

1. (Optional) You can verify the RSA key fingerprint on your running instance by using one of the following commands on your local system (not on the instance). This is useful if you've launched your instance from a public AMI from a third party. Locate the `SSH HOST KEY FINGERPRINTS` section, and note the RSA fingerprint (for example, `1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f`) and compare it to the fingerprint of the instance.
 - [get-console-output](#) (AWS CLI)

```
aws ec2 get-console-output --instance-id instance_id
```

Ensure that the instance is in the `running` state, not the `pending` state. The `SSH HOST KEY FINGERPRINTS` section is only available after the first boot of the instance.

2. In a command-line shell, change directories to the location of the private key file that you created when you launched the instance.
3. Use the following command to set the permissions of your private key file so that only you can read it.

```
chmod 400 /path/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 \(p. 914\)](#).

4. Use the `ssh` command to connect to the instance. You specify the private key (`.pem`) file and `user_name@public_dns_name`. For example, if you used Amazon Linux 2 or the Amazon Linux AMI, the user name is `ec2-user`.

```
ssh -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
```

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)?
```

5. (IPv6 only) Alternatively, 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. For example, if you used Amazon Linux 2 or the Amazon Linux AMI, the user name is `ec2-user`.

```
ssh -i /path/my-key-pair.pem ec2-user@2001:db8:1234:1a00:9691:9503:25ad:1761
```

6. (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you obtained in step 1. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.
7. 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

- **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>.

- **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 **Show/Hide** icon 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 \(p. 661\)](#).

- **Locate the private key and verify permissions**

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. Verify that the `.pem` file has permissions of 0400, not 0777. For more information, see [Error: Unprotected Private Key File \(p. 914\)](#).

- **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` or `root`.
- 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 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 does not allow incoming SSH traffic by default. For more information, see [Authorizing Inbound Traffic for Your Linux Instances \(p. 628\)](#).

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. (Optional) You can verify the RSA key fingerprint on your instance by using one of the following commands on your local system (not on the instance). This is useful if you've launched your instance from a public AMI from a third party. Locate the `SSH HOST KEY FINGERPRINTS` section, and note the RSA fingerprint (for example, `1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f`) and compare it to the fingerprint of the instance.

- [get-console-output \(AWS CLI\)](#)

```
aws ec2 get-console-output --instance-id instance_id
```

The SSH HOST KEY FINGERPRINTS section is only available after the first boot of the instance.

2. In a command shell, change directories to the location of the private key file that you specified when you launched the instance.
3. Use the **chmod** command to make sure that your private key file isn't publicly viewable. For example, if the name of your private key file is `my-key-pair.pem`, use the following command:

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

4. 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 `ec2-user`, and the public DNS name of the instance is `ec2-198-51-100-1.compute-1.amazonaws.com`, use the following command to copy the file to the `ec2-user` home directory.

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt ec2-
user@ec2-198-51-100-1.compute-1.amazonaws.com:~
```

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)?
```

5. (IPv6 only) Alternatively, you can transfer a file using the IPv6 address for the instance. The IPv6 address must be enclosed in square brackets ([]), which must be escaped (\).

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt ec2-user@
\[2001:db8:1234:1a00:9691:9503:25ad:1761\]:~
```

6. (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you obtained in step 1. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.
7. 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
```

8. 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 the following command on your local computer:

```
scp -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com:~/  
SampleFile.txt ~/SampleFile2.txt
```

9. (IPv6 only) Alternatively, you can transfer files in the other direction using the instance's IPv6 address.

```
scp -i /path/my-key-pair.pem ec2-user@[2001:db8:1234:1a00:9691:9503:25ad:1761]:~/  
SampleFile.txt ~/SampleFile2.txt
```

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.

Note

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.

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:

- **Install PuTTY**

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.

- **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 **Show/Hide** icon 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 \(p. 661\)](#).

- **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.

- **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 or root.
- 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 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 does not allow incoming SSH traffic by default. For more information, see [Authorizing Inbound Traffic for Your Linux Instances \(p. 628\)](#).

Converting Your Private Key Using PuTTYgen

PuTTY does not natively support the private key format (.pem) generated by Amazon EC2. PuTTY has a tool named PuTTYgen, which can convert keys to the required PuTTY format (.ppk). You must convert your private key into this format (.ppk) before attempting to connect to your instance using PuTTY.

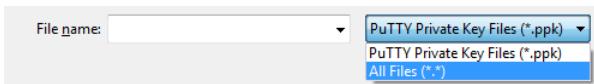
To convert your private key

1. Start PuTTYgen (for example, 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, select 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 then choose **Open**. Choose **OK** to dismiss the confirmation dialog box.
5. Choose **Save private key** to save the key in the format that PuTTY can use. 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, so 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 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). 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.

Starting a PuTTY Session

Use the following procedure to connect to your Linux instance using PuTTY. You need the `.ppk` file that you created for your private key. If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](#).

To start a PuTTY session

1. (Optional) You can verify the RSA key fingerprint on your instance using the `get-console-output` (AWS CLI) command on your local system (not on the instance). This is useful if you've launched your instance from a public AMI from a third party. Locate the `SSH HOST KEY FINGERPRINTS` section, and note the RSA fingerprint (for example, `1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f`) and compare it to the fingerprint of the instance.

```
aws ec2 get-console-output --instance-id instance_id
```

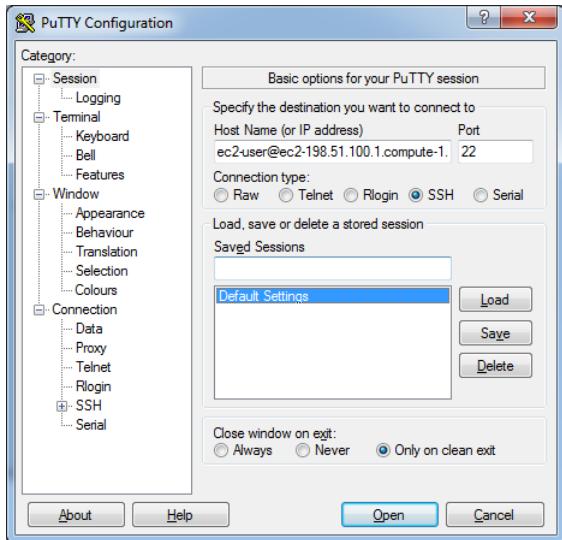
Here is an example of what you should look for:

```
-----BEGIN SSH HOST KEY FINGERPRINTS-----  
... 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f ...  
-----END SSH HOST KEY FINGERPRINTS-----
```

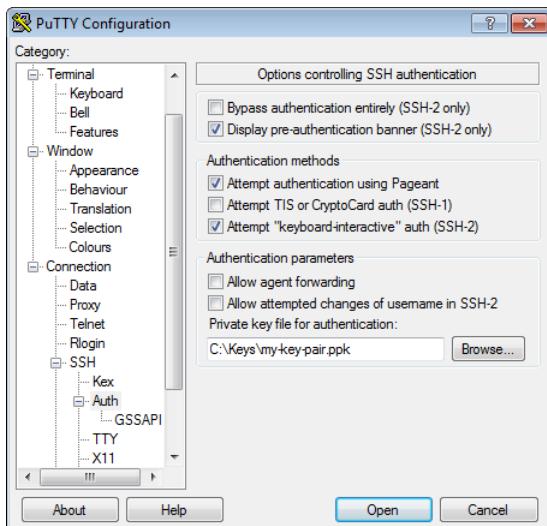
The `SSH HOST KEY FINGERPRINTS` section is only available after the first boot of the instance.

2. Start PuTTY (from the **Start** menu, choose **All Programs > PuTTY > PuTTY**).
3. In the **Category** pane, choose **Session** and complete the following fields:
 - a. In the **Host Name** box, enter `user_name@public_dns_name`. Be sure to specify the appropriate user name for your AMI. For example:
 - 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` or `root`.
 - 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.
 - b. (IPv6 only) To connect using your instance's IPv6 address, enter `user_name@ipv6_address`. Be sure to specify the appropriate user name for your AMI. For example:
 - 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` or `root`.
 - 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.
 - c. Under **Connection type**, select **SSH**.

- d. Ensure that **Port** is 22.



4. (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.
5. In the **Category** pane, expand **Connection**, expand **SSH**, and then choose **Auth**. Complete the following:
 - a. Choose **Browse**.
 - b. Select the .ppk file that you generated for your key pair, and then choose **Open**.
 - c. (Optional) If you plan to start this session again later, you can save the session information for future use. Choose **Session** in the **Category** tree, enter a name for the session in **Saved Sessions**, and then choose **Save**.
 - d. Choose **Open** to start the PuTTY session.



6. 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 you are connecting to.
7. (Optional) Verify that the fingerprint in the security alert dialog box matches the fingerprint that you previously obtained in step 1. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.
8. 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 \(p. 406\)](#).

To use PSCP, you need the private key you generated in [Converting Your Private Key Using PuTTYgen \(p. 403\)](#). You also need the public DNS address of your Linux instance.

The following example transfers the file `Sample_file.txt` from the `C:\` drive on a Windows computer to the `ec2-user` home directory on an Amazon Linux instance:

```
pscp -i C:\\path\\my-key-pair.ppk C:\\path\\Sample_file.txt ec2-user@public_dns:/home/ec2-user/Sample_file.txt
```

(IPv6 only) The following example transfers the file `Sample_file.txt` using the instance's IPv6 address. The IPv6 address must be enclosed in square brackets (`[]`).

```
pscp -i C:\\path\\my-key-pair.ppk C:\\path\\Sample_file.txt ec2-user@[ipv6-address]:/home/ec2-user/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 machine to your Linux instance or synchronize entire directory structures between the two systems.

To use WinSCP, you need the private key you generated in [Converting Your Private Key Using PuTTYgen \(p. 403\)](#). You also need the public DNS address of your Linux instance.

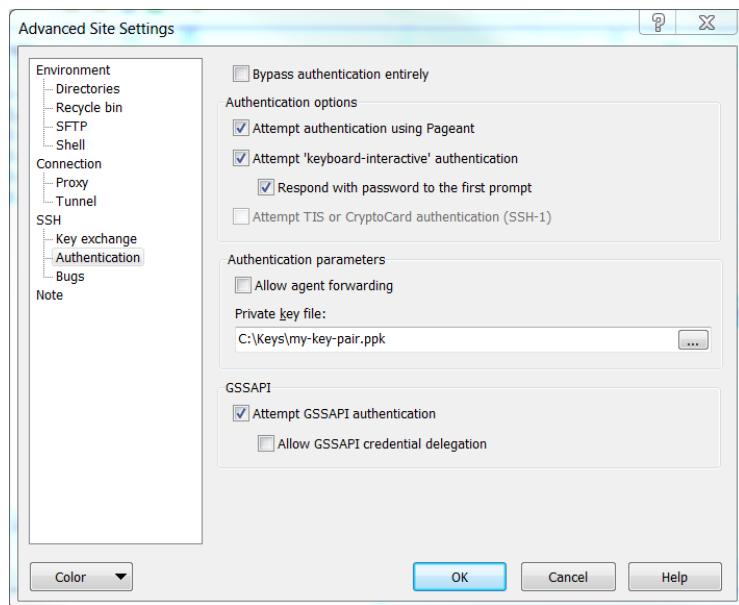
1. Download and install WinSCP from <http://winscp.net/eng/download.php>. For most users, the default installation options are OK.
2. Start WinSCP.
3. At the **WinSCP login** screen, for **Host name**, enter the public DNS hostname or public IPv4 address for your instance.

(IPv6 only) 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.

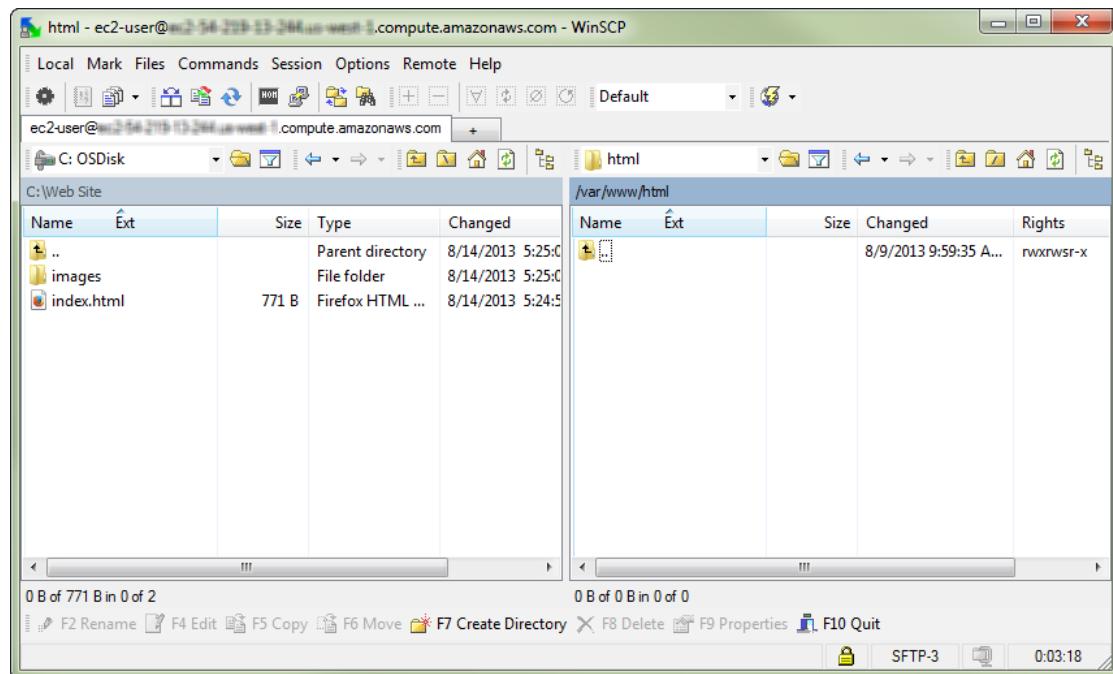
- For Amazon Linux 2 or the Amazon Linux AMI, the user name is `ec2-user`.
 - For a Centos AMI, the user name is `centos`.
 - For a Debian AMI, the user name is `admin` or `root`.
 - For a Fedora AMI, the user name is `ec2-user` or `fedora`.
 - For a RHEL AMI, the user name is `ec2-user` or `root`.
 - 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. For newer versions of WinSCP, choose **Advanced** to open the advanced site settings and then under **SSH**, choose **Authentication** to find the **Private key file** setting.

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 [Converting Your Private Key Using PuTTYgen \(p. 403\)](#).

6. (Optional) In the left panel, choose **Directories**, and then, for **Remote directory**, enter the path for the directory you want to add files to. For newer versions of WinSCP, choose **Advanced** to open the advanced site settings and then under **Environment**, choose **Directories** to find the **Remote directory** setting.
7. Choose **Login** to connect, and choose **Yes** to add the host fingerprint to the host cache.



- 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

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](#).

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.

Note

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.

If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](#).

Contents

- [Prerequisites \(p. 397\)](#)
- [Connecting to Your Linux Instance using the Windows Subsystem for Linux \(p. 410\)](#)
- [Transferring Files to Linux Instances from Linux Using SCP \(p. 411\)](#)
- [Uninstalling Windows Subsystem for Linux \(p. 413\)](#)

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 \(p. 397\)](#), and the experience is just like using native Linux.

Prerequisites

Before you connect to your Linux instance, complete the following prerequisites:

- **Install the Windows Subsystem for Linux (WSL) and a Linux distribution**

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.

- **Install the AWS CLI Tools**

(Optional) If you're using a public AMI from a third party, you can use the command line tools to verify the fingerprint. For more information about installing the AWS CLI, see [Getting Set Up](#) in the [AWS Command Line Interface User Guide](#).

- **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 **Show/Hide** icon 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 \(p. 661\)](#).

- **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?](#).

```
cp /mnt/<Windows drive letter>/path/my-key-pair.pem ~/WSL-path/my-key-pair.pem
```

- **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` or `root`.
 - 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 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 does not allow incoming SSH traffic by default. For more information, see [Authorizing Inbound Traffic for Your Linux Instances \(p. 628\)](#).

Connecting to Your Linux Instance using the Windows Subsystem for Linux

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. (Optional) You can verify the RSA key fingerprint on your running instance by using one of the following commands on your local system (not on the instance). This is useful if you've launched your instance from a public AMI from a third party. Locate the `SSH HOST KEY FINGERPRINTS` section, and note the RSA fingerprint (for example, `1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f`) and compare it to the fingerprint of the instance.
 - [get-console-output \(AWS CLI\)](#)

```
aws ec2 get-console-output --instance-id instance_id
```

Ensure that the instance is in the `running` state, not the `pending` state. The `SSH HOST KEY FINGERPRINTS` section is only available after the first boot of the instance.

2. In a command-line shell, change directories to the location of the private key file that you created when you launched the instance.
3. Use the `chmod` command to make sure that your private key file isn't publicly viewable. For example, if the name of your private key file is `my-key-pair.pem`, use the following command:

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

4. Use the `ssh` command to connect to the instance. You specify the private key (`.pem`) file and `user_name@public_dns_name`. For example, if you used Amazon Linux 2 or the Amazon Linux AMI, the user name is `ec2-user`.

```
ssh -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
```

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)?
```

5. (IPv6 only) Alternatively, 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. For example, if you used Amazon Linux 2 or the Amazon Linux AMI, the user name is **ec2-user**.

```
ssh -i /path/my-key-pair.pem ec2-user@2001:db8:1234:1a00:9691:9503:25ad:1761
```

6. (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you obtained in step 1. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.
7. 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

- **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>.

- **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 **Show/Hide** icon 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 \(p. 661\)](#).

- **Locate the private key and verify permissions**

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. Verify that the `.pem` file has permissions of 0400, not 0777. For more information, see [Error: Unprotected Private Key File \(p. 914\)](#).

- **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` or `root`.
- 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 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 does not allow incoming SSH traffic by default. For more information, see [Authorizing Inbound Traffic for Your Linux Instances \(p. 628\)](#).

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. (Optional) You can verify the RSA key fingerprint on your instance by using one of the following commands on your local system (not on the instance). This is useful if you've launched your instance from a public AMI from a third party. Locate the `SSH HOST KEY FINGERPRINTS` section, and note the RSA fingerprint (for example, `1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f`) and compare it to the fingerprint of the instance.

- [get-console-output](#) (AWS CLI)

```
aws ec2 get-console-output --instance-id instance_id
```

The `SSH HOST KEY FINGERPRINTS` section is only available after the first boot of the instance.

2. In a command shell, change directories to the location of the private key file that you specified when you launched the instance.
3. Use the `chmod` command to make sure that your private key file isn't publicly viewable. For example, if the name of your private key file is `my-key-pair.pem`, use the following command:

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

4. 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 `ec2-user`, and the public DNS name of the instance is `ec2-198-51-100-1.compute-1.amazonaws.com`, use the following command to copy the file to the `ec2-user` home directory:

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt ec2-
user@ec2-198-51-100-1.compute-1.amazonaws.com:~
```

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)?
```

5. (IPv6 only) Alternatively, you can transfer a file using the IPv6 address for the instance. The IPv6 address must be enclosed in square brackets ([]), which must be escaped (\).

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt ec2-user@
\[2001:db8:1234:1a00:9691:9503:25ad:1761\]:~
```

6. (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you obtained in step 1. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.
7. 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
```

8. 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 the following command on your local computer:

```
scp -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com:~/
SampleFile.txt ~/SampleFile2.txt
```

9. (IPv6 only) Alternatively, you can transfer files in the other direction using the instance's IPv6 address:

```
scp -i /path/my-key-pair.pem ec2-user@\[2001:db8:1234:1a00:9691:9503:25ad:1761\]:~
SampleFile.txt ~/SampleFile2.txt
```

Uninstalling Windows Subsystem for Linux

For information about uninstalling Windows Subsystem for Linux, see [How do I uninstall WSL?](#).

Connecting to Your Linux Instance Using MindTerm

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.

Note

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.

The following instructions explain how to connect to your instance using MindTerm through the Amazon EC2 console. If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](#).

Important

The Chrome browser does not support the NPAPI plugin, and therefore cannot run the MindTerm client. For more information, go to the Chromium [NPAPI deprecation article](#). You can use Firefox, Safari, or Internet Explorer 9 or higher instead.

Prerequisites

- **Install Java**

Your Linux computer most likely includes Java. If not, see [How do I enable Java in my web browser?](#). On a Windows or macOS client, you must run your browser using administrator credentials. For Linux, additional steps may be required if you are not logged in as `root`.

- **Enable Java in your browser**

For instructions, see https://java.com/en/download/help/enable_browser.xml.

- **Locate the private key and verify permissions**

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. Verify that the `.pem` file has permissions of 0400, not 0777. For more information, see [Error: Unprotected Private Key File \(p. 914\)](#).

- **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` or `root`.
- 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 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 does not allow incoming SSH traffic by default. For more information, see [Authorizing Inbound Traffic for Your Linux Instances \(p. 628\)](#).

Starting MindTerm

To connect to your instance using a web browser with MindTerm

1. In the Amazon EC2 console, choose **Instances** in the navigation pane.

2. Select the instance, and then choose **Connect**.
3. Choose **A Java SSH client directly from my browser (Java required)**.
4. Amazon EC2 automatically detects the public DNS name of your instance and then populates **Public DNS** for you. It also detects the name of the key pair that you specified when you launched the instance. Complete the following, and then choose **Launch SSH Client**.
 - a. In **User name**, enter the user name to log in to 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` or `root`.
 - 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.
 - b. In **Private key path**, enter the fully qualified path to your private key (`.pem`) file, including the key pair name; for example:
`C:\KeyPairs\my-key-pair.pem`
- c. (Optional) Choose **Store in browser cache** to store the location of the private key in your browser cache. This enables Amazon EC2 to detect the location of the private key in subsequent browser sessions, until you clear your browser's cache.
5. If necessary, choose **Yes** to trust the certificate, and choose **Run** to run the MindTerm client.
6. If this is your first time running MindTerm, a series of dialog boxes asks you to accept the license agreement, to confirm setup for your home directory, and to confirm setup of the known hosts directory. Confirm these settings.
7. A dialog prompts you to add the host to your set of known hosts. If you do not want to store the host key information on your local computer, choose **No**.
8. A window opens and you are connected to your instance.

If you chose **No** in the previous step, you see the following message, which is expected:

Verification of server key disabled in this session.

Stop and Start Your Instance

You can stop and restart 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 \(p. 416\)](#) 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 \(p. 421\)](#).

Contents

- [Overview \(p. 416\)](#)
- [Stopping and Starting Your Instances \(p. 417\)](#)
- [Modifying a Stopped Instance \(p. 418\)](#)
- [Troubleshooting \(p. 418\)](#)

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 \(p. 85\)](#).

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.
- EC2-Classic: We release the public and private IPv4 addresses for the instance when you stop the instance, and assign new ones when you restart it.

EC2-VPC: The instance retains its private IPv4 addresses and any IPv6 addresses when stopped and restarted. We release the public IPv4 address and assign a new one when you restart it.

- EC2-Classic: We disassociate any Elastic IP address that's associated with the instance. You're charged for Elastic IP addresses that aren't associated with an instance. When you restart the instance, you must associate the Elastic IP address with the instance; we don't do this automatically.

EC2-VPC: The instance retains its associated Elastic IP addresses. You're charged for any Elastic IP addresses associated with a stopped instance.

- 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 restarting it. For more information about ClassicLink, see [ClassicLink \(p. 638\)](#).

For more information, see [Differences Between Reboot, Stop, and Terminate \(p. 353\)](#).

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.

Stopping and Starting Your Instances

You can start and stop 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 \(p. 423\)](#).

To stop and start an Amazon EBS-backed instance using the console

1. In the navigation pane, choose **Instances**, and select the instance.
2. [EC2-Classic] If the instance has an associated Elastic IP address, write down the Elastic IP address and the instance ID shown in the details pane.
3. Choose **Actions**, select **Instance State**, and then choose **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. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. In the confirmation dialog box, choose **Yes, Stop**. It can take a few minutes for the instance to stop.

[EC2-Classic] When the instance state becomes stopped, the **Elastic IP**, **Public DNS (IPv4)**, **Private DNS**, and **Private IPs** fields in the details pane are blank to indicate that the old values are no longer associated with the instance.

5. While your instance is stopped, you can modify certain instance attributes. For more information, see [Modifying a Stopped Instance \(p. 418\)](#).
6. To restart 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.

[EC2-Classic] When the instance state becomes `running`, the **Public DNS (IPv4)**, **Private DNS**, and **Private IPs** fields in the details pane contain the new values that we assigned to the instance.

8. [EC2-Classic] If your instance had an associated Elastic IP address, you must re-associate it as follows:
 - a. In the navigation pane, choose **Elastic IPs**.
 - b. Select the Elastic IP address that you wrote down before you stopped the instance.
 - c. Choose **Actions**, and then select **Associate address**.
 - d. Select the instance ID that you wrote down before you stopped the instance, and then choose **Associate**.

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 \(p. 228\)](#).
- To change the user data for your instance, see [Working with Instance User Data \(p. 457\)](#).
- To enable or disable EBS-optimization for your instance, see [Modifying EBS-Optimization \(p. 818\)](#).
- To change the `DeleteOnTermination` attribute of the root volume for your instance, see [Updating the Block Device Mapping of a Running Instance \(p. 873\)](#).

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 \(p. 916\)](#).

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 remains on the same physical host, so your instance 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 restarting 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 \(p. 479\)](#).

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 four 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.

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.

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 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. 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.

Topics

- [Identifying Instances Scheduled for Retirement \(p. 419\)](#)
- [Working with Instances Scheduled for Retirement \(p. 420\)](#)

For more information about types of instance events, see [Scheduled Events for Your Instances \(p. 479\)](#).

Identifying Instances Scheduled for Retirement

If your instance is scheduled for retirement, you'll receive an email prior to the event with the instance ID and retirement date. This email is sent to the address that's associated with your account; the same email address that you use to log in to the AWS Management Console. If you use an email account that you do not check regularly, then you can use the Amazon EC2 console or the command line to determine if any of your instances are scheduled for retirement. To update the contact information for your account, go to the [Account Settings](#) page.

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 associated with your Amazon EC2 instances and volumes, organized by region.

Scheduled Events



US East (N. Virginia):

1 instances have scheduled events

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 with 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)

Working with Instances Scheduled for Retirement

There are a number of actions available to you when your instance is scheduled for retirement. The action you take depends on whether your instance root device is an Amazon EBS volume, or an instance store volume. If you do not know what your instance root device type is, you can find out using the Amazon EC2 console or the command line.

Determining Your Instance Root Device Type

To determine your instance root device type using the console

1. In the navigation pane, select **Events**. Use the filter lists to identify retiring instances, as demonstrated in the procedure above, [Identifying instances scheduled for retirement \(p. 419\)](#).
2. In the **Resource Id** column, select the instance ID to go to the **Instances** page.
3. Select the instance and locate 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.

To determine your instance root device type 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)

Managing Instances Scheduled for Retirement

You can perform one of the actions listed below in order to preserve the data on your retiring instance. 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's 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 they are attached to an EBS-backed instance.

Instance Root Device Type	Action
EBS	Wait for the scheduled retirement date - when the instance is stopped - or stop the instance yourself before the retirement date. You can start

Instance Root Device Type	Action
	the instance again at any time. 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 associated with your instance, see Stop and Start Your Instance (p. 415) .
EBS	Create an EBS-backed AMI from your instance, and launch a replacement instance. For more information, see Creating an Amazon EBS-Backed Linux AMI (p. 101) .
Instance store	Create an instance store-backed AMI from your instance using the AMI tools, and launch a replacement instance. For more information, see Creating an Instance Store-Backed Linux AMI (p. 105) .
Instance store	Convert your instance to an EBS-backed instance by transferring your data to an EBS volume, taking a snapshot of the volume, and then creating an 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. 116) .

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 restart an instance after you've terminated it. However, you can launch additional instances using the same AMI. If you'd rather stop and restart your instance, see [Stop and Start Your Instance \(p. 415\)](#). For more information, see [Differences Between Reboot, Stop, and Terminate \(p. 353\)](#).

Contents

- [Instance Termination \(p. 421\)](#)
- [Terminating an Instance \(p. 422\)](#)
- [Enabling Termination Protection for an Instance \(p. 422\)](#)
- [Changing the Instance Initiated Shutdown Behavior \(p. 423\)](#)
- [Preserving Amazon EBS Volumes on Instance Termination \(p. 424\)](#)
- [Troubleshooting \(p. 426\)](#)

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, therefore 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 \(p. 424\)](#).

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 for an Instance \(p. 422\)](#).

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 \(p. 423\)](#).

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.

Terminating an Instance

You can terminate an instance using the AWS Management Console or the command line.

To terminate an instance using the console

1. Before you terminate the 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 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 for an Instance

By default, you can terminate your instance using the Amazon EC2 console, command line interface, or API. If you want 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 \(p. 423\)](#).

Limits

You can't enable termination protection for Spot instances — a Spot instance is terminated when the Spot price exceeds your bid price. However, you can prepare your application to handle Spot instance interruptions. For more information, see [Spot Instance Interruptions \(p. 325\)](#).

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, choose **Actions, Instance Settings**, and then choose **Change Termination Protection**.
2. Select **Yes, Enable**.

To disable termination protection for a running or stopped instance

1. Select the instance, choose **Actions, Instance Settings**, and then choose **Change Termination Protection**.
2. Select **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

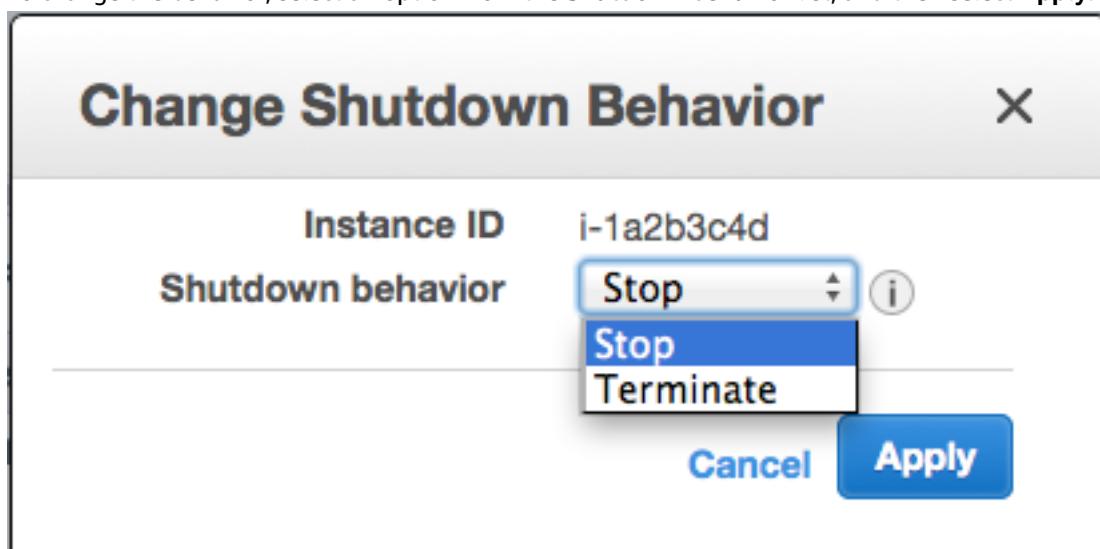
By default, when you initiate a shutdown from an Amazon EBS-backed instance (using a command such as `shutdown`, `halt`, or `poweroff`), the instance stops. You can change this behavior using the `InstanceInitiatedShutdownBehavior` attribute for the instance so that it terminates instead. You can update this attribute while the instance is running or stopped.

Note that instance store-backed instances can be terminated but they can't be 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, select **Actions**, **Instance Settings**, and then choose **Change Shutdown Behavior**. The current behavior is already selected.
4. To change the behavior, select an option from the **Shutdown behavior** list, and then select **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.

By default, the `DeletionOnTermination` attribute for the root volume of an instance is set to `true`. Therefore, the default is to delete the root volume of an instance when the instance terminates.

By default, when you attach an EBS volume to an instance, 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.

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 \(p. 873\)](#).

You can change 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 \(p. 425\)](#)
- [Changing the Root Volume to Persist at Launch Using the Command Line \(p. 425\)](#)
- [Changing the Root Volume of a Running Instance to Persist Using the Command Line \(p. 426\)](#)

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**, click 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 your instance is in the `shutting-down` state for longer than usual, it will eventually be cleaned up (terminated) by automated processes within the Amazon EC2 service. For more information, see [Troubleshooting Terminating \(Shutting Down\) Your Instance \(p. 917\)](#).

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. For more information about using Amazon CloudWatch alarms to recover an instance, see [Create Alarms That Stop, Terminate, Reboot, or Recover an Instance \(p. 502\)](#). To troubleshoot issues with instance recovery failures, see [Troubleshooting Instance Recovery Failures](#) in the [Amazon EC2 User Guide for Linux Instances](#).

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

The recover action can also be triggered when an instance is scheduled by AWS to stop or retire due to degradation of the underlying hardware. For more information about scheduled events, see [Scheduled Events for Your Instances \(p. 479\)](#).

The recover action is supported only on instances with the following characteristics:

- Use a C3, C4, C5, M3, M4, M5, R3, R4, R5, T2, or X1 instance
- Run in a VPC (not EC2-Classic)
- 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 IPv4 address, it retains the public IPv4 address after recovery.

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 \(p. 427\)](#)
- [Managing Software on Your Linux Instance \(p. 428\)](#)
- [Managing User Accounts on Your Linux Instance \(p. 432\)](#)
- [Processor State Control for Your EC2 Instance \(p. 434\)](#)
- [Setting the Time for Your Linux Instance \(p. 439\)](#)
- [Changing the Hostname of Your Linux Instance \(p. 444\)](#)
- [Setting Up Dynamic DNS on Your Linux Instance \(p. 447\)](#)
- [Running Commands on Your Linux Instance at Launch \(p. 448\)](#)
- [Instance Metadata and User Data \(p. 454\)](#)

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 Linux Instance \(p. 428\)](#).

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 Linux Instance \(p. 432\)](#).

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 \(p. 439\)](#).

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 Linux Instance \(p. 444\)](#). You can configure an instance to use a dynamic DNS service provider. For more information, see [Setting Up Dynamic DNS on Your Linux Instance \(p. 447\)](#).

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 \(p. 448\)](#).

Managing Software on Your 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 \(p. 428\)](#)
- [Adding Repositories \(p. 430\)](#)
- [Finding Software Packages \(p. 431\)](#)
- [Installing Software Packages \(p. 431\)](#)
- [Preparing to Compile Software \(p. 432\)](#)

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 \(p. 428\)](#).

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 \(p. 430\)](#). For help finding packages in enabled repositories, see [Finding Software Packages \(p. 431\)](#). For information about installing software on an Amazon Linux instance, see [Installing Software Packages \(p. 431\)](#).

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 \(p. 432\)](#).

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

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

These procedures are intended for use with Amazon Linux. For more 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

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

These procedures are intended for use with Amazon Linux. For more 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

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

These procedures are intended for use with Amazon Linux. For more information about other distributions, see their specific documentation.

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.

```
[ec2-user ~]$ sudo yum search "find"
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
```

Installing Software Packages

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

These procedures are intended for use with Amazon Linux. For more 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

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

These procedures are intended for use with Amazon Linux. For more 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 Linux Instance

Each Linux instance type launches with a default Linux system user account. For Amazon Linux 2 or the Amazon Linux, the user name is `ec2-user`. For Centos, the user name is `centos`. For Debian, the user name is `admin` or `root`. For Fedora, the user name is `ec2-user` or `fedora`. For RHEL, the user name

is `ec2-user` or `root`. For SUSE, the user name is `ec2-user` or `root`. For Ubuntu, the user name is `ubuntu`. Otherwise, if `ec2-user` and `root` don't work, check with your 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*.

Using the default user account is adequate for many applications, but you may choose to add user accounts so that individuals can have their own files and workspaces. Creating user accounts for new users is much more secure than granting multiple (possibly inexperienced) users access to the `ec2-user` account, because that account can cause a lot of damage to a system when used improperly.

After you add the user account, you must set up access keys that allow the user to log in.

Prerequisites

Create a key pair for the user or use an existing key pair. For more information, see [Creating a Key Pair Using Amazon EC2 \(p. 523\)](#). To retrieve a public key from an existing key pair, see [Retrieving the Public Key for Your Key Pair on Linux \(p. 525\)](#).

To add a user account

1. Use the following `adduser` command to add the `newuser` account to the system (with an entry in the `/etc/passwd` file). This command also creates a group and a home directory for the account.

```
[ec2-user ~]$ sudo adduser newuser
```

[Ubuntu] When adding a user to an Ubuntu system, include the `--disabled-password` option with this command to avoid adding a password to the account.

```
[ubuntu ~]$ sudo adduser newuser --disabled-password
```

2. Switch to the new account so that newly created files have the proper ownership.

```
[ec2-user ~]$ sudo su - newuser  
[newuser ~]$
```

Notice that the prompt changes from `ec2-user` to `newuser` to indicate that you have switched the shell session to the new account.

3. 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.

4. 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.

5. Open the `authorized_keys` file using your favorite text editor (such as `vim` or `nano`).

```
[newuser ~]$ nano .ssh/authorized_keys
```

Paste the public key for your key pair into the file and save the changes. For example:

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBITntckiJ7FbtXJMLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
gaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb70z1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

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.

To remove a user from the system

If a user account is no longer needed, you can remove that account so that it may no longer be used. When you specify the `-r` option, the user's home directory and mail spool are deleted. To keep the user's home directory and mail spool, omit the `-r` option.

```
[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
- Compute optimized: c4.8xlarge
- Memory optimized: r4.8xlarge | r4.16xlarge | x1.16xlarge | x1.32xlarge | x1e.8xlarge | x1e.16xlarge | x1e.32xlarge
- Storage optimized: d2.8xlarge | i3.8xlarge | i3.16xlarge | i3.metal | h1.8xlarge | h1.16xlarge
- Accelerated computing: f1.16xlarge | g3.16xlarge | 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
- Compute optimized: c5.9xlarge | c5.18xlarge | c5d.9xlarge | c5d.18xlarge
- Memory optimized: r5.12xlarge | r5.24xlarge | r5d.12xlarge | r5d.24xlarge | z1d.6xlarge | z1d.12xlarge

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 \(p. 435\)](#)
- [High Performance and Low Latency by Limiting Deeper C-states \(p. 436\)](#)
- [Baseline Performance with the Lowest Variability \(p. 438\)](#)

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 ~]$ 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   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   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"
```

7. 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

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
```

5. 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"
```

6. 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
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
  65.33 19.02 100.00 0.00
  0  0  18  0.04 2.90 2.90   0 99.96  0.00  0.00  0.00
  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  0.00  0.00  0.00
  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  0.00  0.00  0.00
  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 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 you can access from your instance. 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. 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.

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
```

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. Start the `chrony` daemon (`chronyd`).

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

```
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
||          |----- xxxx = adjusted offset,
||          |----- 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
```

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
|| |----- xxxx = adjusted offset,
|| |----- 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.

5. 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       : 0.0000000011 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
```

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, but you may wish to change the time on an instance to the local time or to another time zone in your network.

Important

These procedures are intended for use with Amazon Linux. For more information about other distributions, see their specific documentation.

To change the time zone on an instance

1. 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       GMT0       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. In this example, you can use the time zone file for Los Angeles, `/usr/share/zoneinfo/America/Los_Angeles`.

2. Update the `/etc/sysconfig/clock` file with the new time zone.
 - 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**.
 - 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 your 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
```

Changing the Hostname of Your 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. Instances launched into EC2-Classic also receive a new private IPv4 address, private DNS hostname, and system hostname when they're stopped and restarted; instances launched into a VPC don't.

Important

These procedures are intended for use with Amazon Linux. For more 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.

1. • 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
```

2. 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**).

3. 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

1. • 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
```

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` file in your favorite text editor (such as **vim** or **nano**). You need to use **sudo** with the editor command because `/etc/bashrc` is 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` (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]\\\$ "
# You might want to have e.g. tty in prompt (e.g. more virtual machines)
# 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?](#)
- [How do I assign a static hostname to a private Amazon EC2 instance running SuSe Linux?](#)
- [How do I assign a static hostname to a private Amazon EC2 instance running Ubuntu Linux?](#)

Setting Up Dynamic DNS on Your 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

These procedures are intended for use with Amazon Linux. For more 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 \(p. 32\)](#) 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.

By default, user data and cloud-init directives only run during the first boot cycle when you launch an instance. However, AWS Marketplace vendors and owners of third-party AMIs may have made their own customizations for how and when scripts run.

Contents

- [Prerequisites \(p. 449\)](#)
- [User Data and Shell Scripts \(p. 449\)](#)
- [User Data and the Console \(p. 449\)](#)
- [User Data and cloud-init Directives \(p. 451\)](#)
- [User Data and the AWS CLI \(p. 452\)](#)

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 \(p. 27\)](#). 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 \(p. 19\)](#).

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, and 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.

Important

User data scripts and cloud-init directives run only during the first boot cycle when an instance is launched.

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).

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.

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 Your Instance from an AMI \(p. 355\)](#), but when you get to [Step 6 \(p. 357\)](#) 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 0664 {} \;
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 \(p. 537\)](#).

(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. Therefore, if you have any data on instance store volumes that you want to keep, 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 \(p. 151\)](#).

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

User data scripts and cloud-init directives only run during the first boot cycle when an instance is launched.

The Amazon Linux version of cloud-init does not support all of the directives that are available in the base package, and some of the directives have been renamed (such as `repo_update` instead of `apt-upgrade`).

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 Your Instance from an AMI \(p. 355\)](#), but when you get to [Step 6 \(p. 357\)](#) 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 \(p. 537\)](#).

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 \(p. 457\)](#).

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 Linux, use the base64 command to encode the user data.

```
base64 my_script.txt >my_script_base64.txt
```

On Windows, 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 --user-data 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: 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":
    "IyEvYmluL2Jhc2gKeXVtIHVwZGF0ZSAtOpzzXJ2aN1IGH0dHBkIHN0YXJ0CmNoa2NvbmZpZyBodHRwZCBvbg=="
  },
  "InstanceId": "i-1234567890abcdef0"
}
```

On Linux, 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 Windows, 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 more information, see [Instance Metadata Categories \(p. 460\)](#).

Important

Although you can only access instance metadata and user data from within the instance itself, the data is not protected by cryptographic methods. Anyone who can access the instance can view its metadata. Therefore, you should take suitable precautions to protect sensitive data (such as long-lived encryption keys). You should not store sensitive data, such as passwords, as user data.

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 attach a simple script. You can also use this data to build more generic AMIs that can be modified by configuration files supplied at launch time. For example, if you run web servers for various small businesses, they can all use the same AMI and retrieve their content from the Amazon S3 bucket you specify in the user data at launch. To add a new customer at any time, simply create a bucket for the customer, add their content, and launch your AMI. If you launch more than one instance at the same time, the user data is available to all instances in that reservation.

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 \(p. 466\)](#).

Contents

- [Retrieving Instance Metadata \(p. 454\)](#)
- [Working with Instance User Data \(p. 457\)](#)
- [Retrieving Dynamic Data \(p. 458\)](#)
- [Example: AMI Launch Index Value \(p. 458\)](#)
- [Instance Metadata Categories \(p. 460\)](#)
- [Instance Identity Documents \(p. 466\)](#)

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.

To view all categories of instance metadata from within a running instance, use the following URI:

```
http://169.254.169.254/latest/meta-data/
```

Note that you are not billed for HTTP requests used to retrieve instance metadata and user data.

You can use a tool such as cURL, or if your instance supports it, the GET command; for example:

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
```

```
[ec2-user ~]$ GET 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 without having to type out the full URI or category names.

All instance metadata is returned as text (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).

Examples of Retrieving 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.

```
[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
```

This example gets the top-level metadata items. Some items are only available for instances in a VPC. For more information about each of these items, see [Instance Metadata Categories \(p. 460\)](#).

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/  
ami-id  
ami-launch-index  
ami-manifest-path  
block-device-mapping/  
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/
```

These examples get the value of some of the metadata items from the preceding example.

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-id  
ami-12345678
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/reservation-id  
r-fea54097
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-hostname  
ip-10-251-50-12.ec2.internal
```

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-hostname  
ec2-203-0-113-25.compute-1.amazonaws.com
```

This example gets the list of available public keys.

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/  
0=my-public-key
```

This example shows the formats in which public key 0 is available.

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/  
openssh-key
```

This example gets public key 0 (in the OpenSSH key format).

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key  
ssh-rsa MIICiTCFAICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC  
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1M98wDQYDVQQKEwZBbWF6  
b24xFDASBgNVBAstC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXN0Q21sYWMxHzAd  
BgkqhkiG9w0BCQEWEG5vb25lGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhCN  
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD  
VQHHEwdTZWF0dGx1M98wDQYDVQQKEwZBbWF6b24xFDASBgNVBAstC01BTSBDb25z  
b2x1MRIwEAYDVQQDEw1UZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGft  
YXpvbi5jb20wgDQYJKoZIhvcNAQEBBQADgY0AMIGJAOGBAMaK0dn+a4GmWIW  
21uUSfwfEvySwtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T  
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSov/c7ugFFDzQGBzZswY6786gPE  
Ibb3OhjZnzcvQAaRHd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4  
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb  
FFBjvSfpJ1J00zbhNY5f6GuoEdmFJ10ZxBHjNyp378D8uTs7fLvjx79LjSTb  
NYiytVbZPQQU5Yaxu2jXnimvv3rrszlaEXAMPLE my-public-key
```

This example shows the information available for a specific network interface (indicated by the MAC address) on an NAT instance in the EC2-Classic platform.

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/network/interfaces/  
macs/02:29:96:8f:6a:2d/  
device-number  
local-hostname  
local-ipv4s  
mac  
owner-id  
public-hostname  
public-ipv4s
```

This example gets the subnet ID for an instance launched into a VPC.

```
[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 may lead to throttling. Instead, we recommend that you cache the credentials until they start approaching their expiry time.

If you're throttled while accessing the instance metadata service, retry your query with an exponential backoff strategy.

Working with Instance User Data

When working with instance user data, keep the following in mind:

- 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.
- User data is limited to 16 KB. This limit applies to the data in raw form, not base64-encoded form.
- User data must be base64-encoded. The Amazon EC2 console can perform the base64 encoding for you or accept base64-encoded input.
- User data must be decoded when you retrieve it. The data is decoded when you retrieve it using instance metadata and the console.
- 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 \(p. 355\)](#) and [Running Commands on Your Linux Instance at Launch \(p. 448\)](#).

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 \(p. 450\)](#).

Retrieve Instance User Data

To retrieve user data from within a running instance, use the following URL:

```
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:

```
[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:

```
[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 \(p. 452\)](#)

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:

```
[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 \(p. 466\)](#).

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 master and the remaining three acting as replicas. When she launches them, she wants to add user data about the replication strategy for each replicant. 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 replicant's configuration, `replicate-every=5min` defines the second replicant'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-12345678 --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-12345678
- 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
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

Metadata	Value
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.

1. She connects to one of the instances, and retrieves the `ami-launch-index` for that instance to ensure it is one of the replicants:

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-launch-index
2
```

2. She saves the `ami-launch-index` as a variable:

```
[ec2-user ~]$ ami_launch_index=`curl http://169.254.169.254/latest/meta-data/ami-launch-index`
```

3. She saves the user data as a variable:

```
[ec2-user ~]$ user_data=`curl http://169.254.169.254/latest/user-data/`
```

4. Finally, Alice uses the `cut` command to extract the portion of the user data that is applicable to that instance:

```
[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.

Important

Category names that are formatted in red text 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	Version Introduced
ami-id	The AMI ID used to launch the instance.	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.	1.0

Data	Description	Version Introduced
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.	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 AMI manifest file contained an <code>ancestor-amis</code> key.	2007-10-10
block-device-mapping/ami	The virtual device that contains the root/boot file system.	2007-12-15
block-device-mapping/ebs <i>N</i>	The virtual devices associated with Amazon EBS volumes, if any are present. 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
block-device-mapping/ephemeral <i>N</i>	The virtual devices associated with ephemeral devices, if any are present. The <i>N</i> indicates the index of the ephemeral volume.	2007-12-15
block-device-mapping/root	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
block-device-mapping/swap	The virtual devices associated with swap. Not always present.	2007-12-15
elastic-gpus/associations/ <i>elastic-gpu-id</i>	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
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).	1.0

Data	Description	Version Introduced
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/ role-name	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. 621)). Otherwise, not present.	2012-01-12
instance-action	Notifies the instance that it should reboot in preparation for bundling. Valid values: none shutdown bundle-pending.	2008-09-01
instance-id	The ID of this instance.	1.0
instance-type	The type of instance. For more information, see Instance Types (p. 163) .	2007-08-29
kernel-id	The ID of the kernel launched with this instance, if applicable.	2008-02-01
local-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).	2007-01-19
local-ipv4	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).	1.0
mac	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

Data	Description	Version Introduced
<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 device-number 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
<code>network/interfaces/macs/mac/ipv4-associations/public-ip</code>	The private IPv4 addresses that are associated with each <code>public-ip</code> 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). If the instance is in a VPC, this category is only returned if the <code>enableDnsHostnames</code> attribute is set to <code>true</code> . For more information, see Using DNS with Your VPC .	2011-01-01
<code>network/interfaces/macs/mac/public-ipv4s</code>	The 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. Returned only for instances launched into a VPC.	2011-01-01

Data	Description	Version Introduced
<code>network/interfaces/macs/mac/security-group-ids</code>	The IDs of the security groups to which the network interface belongs. Returned only for instances launched into a VPC. For more information on security groups in the EC2-VPC platform, see Security Groups for Your VPC .	2011-01-01
<code>network/interfaces/macs/mac/subnet-id</code>	The ID of the subnet in which the interface resides. Returned only for instances launched into a VPC.	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. Returned only for instances launched into a VPC.	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. Returned only for instances launched into a VPC.	2016-06-30
<code>network/interfaces/macs/mac/vpc-id</code>	The ID of the VPC in which the interface resides. Returned only for instances launched into a VPC.	2011-01-01
<code>network/interfaces/macs/mac/vpc-ipv4-cidr-block</code>	The primary IPv4 CIDR block of the VPC. Returned only for instances launched into a VPC.	2011-01-01
<code>network/interfaces/macs/mac/vpc-ipv4-cidr-blocks</code>	The IPv4 CIDR blocks for the VPC. Returned only for instances launched into a VPC.	2016-06-30
<code>network/interfaces/macs/mac/vpc-ipv6-cidr-blocks</code>	The IPv6 CIDR block of the VPC in which the interface resides. Returned only for instances launched into a VPC.	2016-06-30
<code>placement/availability-zone</code>	The Availability Zone in which the instance launched.	2008-02-01
<code>product-codes</code>	Marketplace product codes associated with the instance, if any.	2007-03-01
<code>public-hostname</code>	The instance's public DNS. If the instance is in a VPC, this category is only returned if the <code>enableDnsHostnames</code> attribute is set to <code>true</code> . For more information, see Using DNS with Your VPC .	2007-01-19
<code>public-ipv4</code>	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

Data	Description	Version Introduced
public-keys/0/openssh-key	Public key. Only available if supplied at instance launch time.	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.	1.0
security-groups	<p>The names of the security groups applied to the instance.</p> <p>After launch, you can only change the security groups of instances running in a VPC. Such changes are reflected here and in network/interfaces/macs/<i>mac</i>/security-groups.</p>	1.0
services/domain	The domain for AWS resources for the region; for example, <code>amazonaws.com</code> for us-east-1.	2014-02-25
services/partition	The partition that the resource is in. For standard AWS regions, the partition is <code>aws</code> . If you have resources in other partitions, the partition is <code>aws-<i>partitionname</i></code> . For example, the partition for resources in the China (Beijing) region is <code>aws-cn</code> .	2015-10-20
spot/instance-action	<p>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.</p> <p>For more information, see instance-action (p. 329).</p>	2016-11-15
spot/termination-time	<p>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, <code>2015-01-05T18:02:00Z</code>) only if the Spot instance has been marked for termination by Amazon EC2.</p> <p>The termination-time item is not set to a time if you terminated the Spot instance yourself. For more information, see termination-time (p. 329).</p>	2014-11-05

Dynamic Data Categories

The following table lists the categories of dynamic data.

Data	Description	Version introduced
fws/instance-monitoring	Value showing whether the customer has enabled detailed one-minute monitoring in CloudWatch. Valid values: enabled disabled	2009-04-04
instance-identity/document	JSON containing instance attributes, such as instance-id, private IP address, etc. See Instance Identity Documents (p. 466) .	2009-04-04
instance-identity/pkcs7	Used to verify the document's authenticity and content against the signature. See Instance Identity Documents (p. 466) .	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. 466) .	2009-04-04

Instance Identity Documents

An instance identity document is a JSON file that describes an instance. The instance identity document is accompanied by a signature and a PKCS7 signature which can be used to verify the accuracy, origin, and authenticity of the information provided in the document. For example, you may have downloaded free software with paid updates.

The instance identity document is generated when the instance is launched, and exposed to the instance through [instance metadata \(p. 454\)](#). It validates the attributes of the instances, such as the subscribed software, instance size, instance type, operating system, and AMI.

Important

Due to the dynamic nature of instance identity documents and signatures, we recommend retrieving the instance identity document and signature regularly.

Obtaining the Instance Identity Document and Signatures

To retrieve the instance identity document, use the following command from your running instance:

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/document
```

The following is example output:

```
{  
    "devpayProductCodes" : null,  
    "marketplaceProductCodes" : [ "1abc2defghijklm3nopqrs4tu" ],  
    "availabilityZone" : "us-west-2b",  
    "privateIp" : "10.158.112.84",  
    "version" : "2017-09-30",  
    "instanceId" : "i-1234567890abcdef0",  
    "billingProducts" : null,  
    "instanceType" : "t2.micro",  
    "accountId" : "123456789012",  
    "imageId" : "ami-5fb8c835",
```

```
"pendingTime" : "2016-11-19T16:32:11Z",
"architecture" : "x86_64",
"kernelId" : null,
"ramdiskId" : null,
"region" : "us-west-2"
}
```

To retrieve the instance identity signature, use the following command from your running instance:

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/signature
```

The following is example output:

```
dExamplesjNQhhJan7pORLpLSr7lJEF4V2DhKGlyoYVBuOYrY9njyBCmhEayaGrhtS/AWY+LPx
1VSQURF5n0gwPNCu06ICT0fNrm5IH7w9ydyaxamplejJw8XvWPxbuRkcN0TAA1p4RtCAqm4ms
x2oALjWSCBExample=
```

To retrieve the PKCS7 signature, use the following command from your running instance:

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/pkcs7
```

The following is example output:

```
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVmxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1M98wDQYDVQQKEwZBbWF6
b24xFDASBgnVBAsTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEg5vb25lQGFTYXpvbi5jb20wHhcNMTEwNDI1MjAONTIxWhCN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQHQHewdTZWF0dGx1M98wDQYDVQQKEwZBbWF6b24xFDASBgnVBAsTC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEg5vb25lQGFT
YXpvbi5jb20wgZ8wDOYJKoZIhvvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4gMWIWJ
21uUsfwfEvyswTc2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/Mb0ITxOUSQv7c7ugFFDzQGBzzswY6786m86gpE
Ibb3OhjZnzcvQAArHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvvcNAQEFBQADgYEAtCu4
nUhVvxUntneD9+h8Mg9q6q+auNkyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNY5f6GuoEdmFJ10ZxBHjJnyp3780D8uTs7fLvjkx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE
```

Verifying the PKCS7 Signature

You can use the PKCS7 signature to verify your instance by validating it against the appropriate AWS public certificate.

The AWS public certificate for the regions provided by an AWS account is as follows:

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAQ0CCQCWukjZ5V4aZzAJBgqhkJOOAQDMFWxCzAJBgNVBAYTA1VTMRkw
FwYDVQQIExBXYXN0aW5ndG9uIFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYD
VQKQExdBbWF6b24gV2ViIFNlcnPzY2VzIExM0zAeFw0xMjAxMDUxMjU2MTJaFw0z
ODAxMDUxMjU2MTJaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQQIExBXYXN0aW5ndG9u
IFN0YXR1MRAwDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQExdBbWF6b24gV2ViIFN1
cnZpY2VzIExM0zCABcwgEsBgcqhkjOOAQBMIIBHwKBgQCjkvcS2bb1VQ4yt/5e
ih5O06kK/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6Ud1Z1gYipr58Kj3nssSNpI6bX3
VyIQzK7wLc1nd/YozqNNmgIyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1j
k+tKqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MbcJ1/U
hhy1KHVpCG19fueQ2s6IL0Ca0/buyC1CiYQk40KNHCChfNiZbd1x1E9rpUp7bnF
1Ra2v1ntMX3caRVDbtPEWmdxSCYSYFDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/Gf
MNmP9CM5eoVQOGx5ho8WqD+aTebs+k2tn92BBPqeZqpWRa5P/+jrdKml1qx4llHW
```

```
MXrs3IgIb6+hUIB+S8dz8/mm00bp7r76RoZVCYab2CZedFut7qc3WUH9+EUAH5mw
vSeDCOUMYQR7R9LINYwouHIZiqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6ROk0k9K
-----END CERTIFICATE-----
```

The AWS public certificate for the AWS GovCloud (US) region is as follows:

```
-----BEGIN CERTIFICATE-----
MIICuzCCAiQCCQDrSGn1RgvSazANBgkqhkiG9w0BAQUFADCBoTELMAkGA1UEBhMC
VVmxCzAJBgNVBAgTAlDbMRawDgYDVQQHEwdTZWF0dGx1MRMwEQQYDVQQKEwpBbWF6
b24uY29tMRYwFAYDVQQLEw1F9zIgQXV0aG9yaXR5MRowGAYDVQQDExFFQzIgQU1J
IEF1dGhvcml0eTEqMCggCSqGS1b3DQEJARYbZWMyLwlpcZEBhbWF6
b24uY29tM84XDTExDgxmjE3MTgwNvoXDTIxMDgwOTE3MTgwNVowgaExCzAJBgNV
BAYTA1VTM0swCQYDVQQIEwJXQTEQMA4GA1UEB2VhdHRsZTETMBEGA1UEChMK
QW1hem9uLmNvbTEWMBQGA1UECxMNRUUmIEF1dGhvcml0eTEaMBgGA1UEAxMRRUUm
IEFNSSBBdXRob3JpdHkxKjAoBgkqhkiG9w0BCQEWG2Vjm1pbmN0Yw5jZs1paWRA
YW1hem9uLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEaQaIcGFFTx/SO
1W5G91jHvyQdGP25n1Y91aXCuOOAUTvSvNGpXrI4AXNrQF+CmIOC4beBASnHCx0
82jYudWBBl9Wiza0psYc9flrczSzVLMmN8w/c78F/95NfiQdnUQPpvqgcMeJo82c
gHkLR7XoFWgMrZJqrcUK0gnsQcb6kakCAwEAATANBgkqhkiG9w0BAQFAAOBqQDF
VHO+UGZr1LCQ78pbBH0GreidQMFfa+W8xASDYUzrMvY3kcIelkoIazvi4VtP07Qc
yAiLr6nkk69Tr/MITnmmsZJZPetshqBndRyL+DaTRnF0/xvBQXj5tEh+AmRjvGtp
6iS1rQoNaN8oEcT2j4b48rmCmnDhRoBcFHwCYs/3w==
-----END CERTIFICATE-----
```

For other regions, contact [AWS Support](#) to get the AWS public certificate.

To verify the PKCS7 signature

- From your instance, create a temporary file for the PKCS7 signature:

```
[ec2-user ~]$ PKCS7=$(mktemp)
```

- Add the -----BEGIN PKCS7----- header to the temporary PKCS7 file:

```
[ec2-user ~]$ echo "-----BEGIN PKCS7-----" > $PKCS7
```

- Append the contents of the PKCS7 signature from the instance metadata, plus a new line:

```
[ec2-user ~]$ curl -s http://169.254.169.254/latest/dynamic/instance-identity/pkcs7 >>
$PKCS7
[ec2-user ~]$ echo "" >> $PKCS7
```

- Append the -----END PKCS7----- footer:

```
[ec2-user ~]$ echo "-----END PKCS7-----" >> $PKCS7
```

- Create a temporary file for the instance identity document:

```
[ec2-user ~]$ DOCUMENT=$(mktemp)
```

- Add the contents of the document from your instance metadata to the temporary document file:

```
[ec2-user ~]$ curl -s http://169.254.169.254/latest/dynamic/instance-identity/document
> $DOCUMENT
```

- Open a text editor and create a file named `AWSpubkey`. Copy and paste the contents of the AWS public certificate above to the file and save it.

- Use the OpenSSL tools to verify the signature as follows:

```
[ec2-user ~]$ openssl smime -verify -in $PKCS7 -inform PEM -content $DOCUMENT -certfile AWSpubkey -noverify > /dev/null  
Verification successful
```

Identify EC2 Linux Instances

You may benefit from being able to determine whether a system is an EC2 instance. There are two methods that you can use to identify 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 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. For a definitive approach, see [Inspecting the Instance Identity Document \(p. 470\)](#).

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-DDC1-B213-94ED-01234ABCDEF
```

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 \(p. 466\)](#).

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. 485)	
Network utilization	NetworkIn (p. 485) NetworkOut (p. 485)	
Disk performance	DiskReadOps (p. 485) DiskWriteOps (p. 485)	
Disk Reads/Writes	DiskReadBytes (p. 485) DiskWriteBytes (p. 485)	
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.

Topics

- [Automated Monitoring Tools \(p. 472\)](#)
- [Manual Monitoring Tools \(p. 473\)](#)

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 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 \(p. 474\)](#).

- **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 \(p. 474\)](#).

- **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 \(p. 483\)](#).

- **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 [What is Amazon CloudWatch Logs?](#).
- **Amazon EC2 Monitoring Scripts** - Perl scripts that can monitor memory, disk, and swap file usage in your instances. For more information, see [Monitoring Memory and Disk Metrics for Amazon EC2 Linux Instances](#).
- **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 click **Instances**, select an instance, and then click the **Monitoring** tab)
 - Volume metric details (In the navigation pane click **Volumes**, select a volume, and then click 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 on specific events scheduled for your instances. Events provide information about upcoming activities such as rebooting or retirement that are planned for your instances, along with the scheduled start and end time of each event.

Contents

- [Status Checks for Your Instances \(p. 474\)](#)
- [Scheduled Events for Your Instances \(p. 479\)](#)

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. This data augments the information that Amazon EC2 already provides about the intended state of each instance (such as pending, running, stopping) as well as the utilization metrics that Amazon CloudWatch monitors (CPU utilization, network traffic, and disk activity).

Status checks are performed every minute and each returns 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. You can, however create or delete 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 \(p. 477\)](#).

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 \(p. 426\)](#).

Contents

- [Types of Status Checks \(p. 475\)](#)

- [Viewing Status Checks \(p. 475\)](#)
- [Reporting Instance Status \(p. 476\)](#)
- [Creating and Editing Status Check Alarms \(p. 477\)](#)

Types of Status Checks

There are two types of status checks: system status checks and instance 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 migrates it to a new host computer. 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

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 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.

To view status checks using the 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.

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 [AWS Support](#) if you need technical assistance with this instance.

Instance Status Checks ⓘ

These checks monitor your software and network configuration for this instance.

Instance reachability check failed at October 7, 2013 11:52:11 AM UTC+2 (16 minutes ago)

[Learn more about this issue](#)

5. 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 \(p. 918\)](#).

Viewing Status Using the Command Line or API

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 a instance status of `impaired`:

```
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 \(p. 918\)](#).

Reporting Instance Status

You can provide feedback if you are having problems with an instance whose status is not shown as `impaired`, or 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

To report instance status 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. Select the **Status Checks** tab, and then choose **Submit feedback**.
5. Complete the **Report Instance Status** form, and then choose **Submit**.

Reporting Status Feedback Using the Command Line or API

Use the following `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 create instance status and system status alarms to notify you when an instance has a failed status check.

Creating a Status Check Alarm Using the Console

You can create status check alarms for an existing instance to monitor instance status or system status. You can configure the alarm to send you a notification by email or stop, terminate, or recover an instance when it fails an [instance status check or system status check \(p. 475\)](#).

To create a status check alarm

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. Select the **Status Checks** tab, and then choose **Create Status Check Alarm**.
5. Select **Send a notification to**. Choose an existing SNS topic, or click **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.
6. (Optional) Choose **Take the action**, and then select the action that you'd like to take.
7. In **Whenever**, select the status check that you want to be notified about.

Note

If you selected **Recover this instance** in the previous step, select **Status Check Failed (System)**.

8. 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.

9. (Optional) In **Name of alarm**, replace the default name with another name for the alarm.
10. 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 clicking 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.

To edit a status check alarm

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**, select **CloudWatch Monitoring**, and then choose **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 metric is `StatusCheckFailed`.

To create a status check alarm using the 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
```

Note

- `--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.
- `--evaluation-periods` is the number of consecutive periods for which the value of the metric must be compared to the threshold. This example uses 2.
- `--alarm-actions` is the list of actions to perform when this alarm is triggered. Each action is specified as an Amazon Resource Name (ARN). 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, with 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.

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 \(p. 479\)](#)
- [Viewing Scheduled Events \(p. 479\)](#)
- [Working with Instances Scheduled to Stop or Retire \(p. 481\)](#)
- [Working with Instances Scheduled for Reboot \(p. 481\)](#)
- [Working with Instances Scheduled for Maintenance \(p. 482\)](#)

Types of Scheduled Events

Amazon EC2 supports the following types of scheduled events for your instances:

- **Instance stop:** The instance will be stopped. When you start it again, it's migrated to a new host computer. Applies only to instances backed by Amazon EBS.
- **Instance retirement:** The instance will be stopped or terminated.
- **Reboot:** Either the instance will be rebooted (instance reboot) or the host computer for the instance will be rebooted (system reboot).
- **System maintenance:** 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.

To view scheduled events for your instances using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **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-3870335	instance-stop

Event: i-3870335

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

3. 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

4. Note that some events are also shown for affected resources. For example, in the navigation pane, choose **Instances**, and then 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. [\(i\)](#)

To view scheduled events for your instances using the command line or API

Use the following AWS CLI command:

```
aws ec2 describe-instance-status --instance-id i-1234567890abcdef0
```

The following is example output showing an instance retirement event:

```
{
    "InstanceStatuses": [
        {
            "InstanceState": {
                "Status": "ok",
                "Details": [
                    {
                        "Status": "passed",
                        "Name": "reachability"
                    }
                ]
            },
            "AvailabilityZone": "us-west-2a",
            "InstanceId": "i-1234567890abcdef0",
            "InstanceState": {
                "Code": 16,
                "Name": "running"
            },
            "SystemStatus": {
                "Status": "ok",
                "Details": [
                    {
                        "Status": "passed",
                        "Name": "cpu"
                    }
                ]
            }
        }
    ]
}
```

```
        "Status": "passed",
        "Name": "reachability"
    }
],
"Events": [
{
    "Code": "instance-stop",
    "Description": "The instance is running on degraded hardware",
    "NotBefore": "2015-05-23T00:00:00.000Z"
}
]
}
}
```

Alternatively, use the following commands:

- [Get-EC2InstanceState](#) (AWS Tools for Windows PowerShell)
- [DescribeInstanceState](#) (Amazon EC2 Query API)

Working with Instances Scheduled to Stop or Retire

When AWS detects irreparable failure of the underlying host computer 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 \(p. 419\)](#).

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 will 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 computer. For more information about stopping your instance, as well as information about the changes to your instance configuration when it's stopped, see [Stop and Start Your Instance \(p. 415\)](#).

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 needs to perform tasks such as installing updates or maintaining the underlying host computer, it can schedule an instance or the underlying host computer for the instance for a reboot. Regardless of any existing instances that are scheduled for reboot, a new instance launch does not require a reboot, as the updates are already applied on the underlying host.

You can determine whether the reboot event is an instance reboot or a system reboot.

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. Select **Instance resources** from the filter list, and then select your instance.
4. In the bottom pane, locate **Event type**. The value is either `system-reboot` or `instance-reboot`.

To view the type of scheduled reboot event using the AWS CLI

Use the following `describe-instance-status` command:

```
aws ec2 describe-instance-status --instance-ids i-1234567890abcdef0
```

Actions for Instance Reboot

You can wait for the instance reboot to occur within its scheduled maintenance window. Alternatively, you can reboot your instance yourself at a time that is convenient for you. For more information, see [Reboot Your Instance \(p. 418\)](#).

After you reboot your instance, the scheduled event for the instance reboot is canceled and the event's description is updated. The pending maintenance to the underlying host computer 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. We recommend that you wait for the system reboot to occur during its scheduled maintenance window. A system reboot typically completes in a matter of minutes, the instance retains its IP address and DNS name, and any data on local instance store volumes is preserved. After the system reboot has occurred, the scheduled event for the instance is cleared, and you can verify that the software on your instance is operating as you expect.

Alternatively, if it is necessary to maintain the instance at a different time, you can stop and start an EBS-backed instance, which migrates it to a new host. However, the data on the local instance store volumes would not be preserved. In the case of an instance store-backed instance, you can launch a replacement instance from your most recent AMI.

Working with Instances Scheduled for Maintenance

When AWS needs to maintain the underlying host computer 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 will be 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 the description of the scheduled event begins with **[Completed]**. It sometimes takes up to 1 hour for this instance status 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 computer. For more information about stopping your instance,

as well as information about the changes to your instance configuration when it's stopped, see [Stop and Start Your Instance \(p. 415\)](#).

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.

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 Disable Detailed Monitoring for Your Instances \(p. 483\)](#).

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 Disable Detailed Monitoring for Your Instances \(p. 483\)](#)
- [List the Available CloudWatch Metrics for Your Instances \(p. 485\)](#)
- [Get Statistics for Metrics for Your Instances \(p. 493\)](#)
- [Graph Metrics for Your Instances \(p. 500\)](#)
- [Create a CloudWatch Alarm for an Instance \(p. 501\)](#)
- [Create Alarms That Stop, Terminate, Reboot, or Recover an Instance \(p. 502\)](#)

Enable or Disable 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 table describes basic and detailed monitoring for instances.

Type	Description
Basic	Data is available automatically in 5-minute periods at no charge.
Detailed	Data is available in 1-minute periods for an additional cost. 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.

Type	Description
	For information about pricing, see the Amazon CloudWatch product page .

Enabling Detailed Monitoring

You can enable detailed monitoring on an instance as you launch it or after the instance is running or stopped.

To enable detailed monitoring 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**.
3. Select the instance, 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 using the 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 using the 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 using the 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...
```

Disabling Detailed Monitoring

You can disable detailed monitoring on an instance as you launch it or after the instance is running or stopped.

To disable detailed monitoring 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, choose **Actions, CloudWatch Monitoring, Disable Detailed Monitoring**.
4. In the **Disable Detailed Monitoring** dialog box, choose **Yes, Disable**.
5. Choose **Close**.

To disable detailed monitoring using the AWS CLI

Use the following [unmonitor-instances](#) command to disable 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 previous 5 minutes of activity for the instance. If you've enabled detailed monitoring, each data point covers the previous 1 minute of activity.

For information about getting the statistics for these metrics, see [Get Statistics for Metrics for Your Instances \(p. 493\)](#).

Instance Metrics

The AWS/EC2 namespace includes the following CPU credit metrics for your T2 instances.

Metric	Description
CPUCreditUsage	<p>[T2 instances] 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>[T2 instances] 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. Once 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 the instance stops, the CPUCreditBalance 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>

Metric	Description
CPUSurplusCreditBalance	<p>[T2 Unlimited instances] The number of surplus credits that have been spent by a T2 Unlimited instance when its CPUCreditBalance is zero.</p> <p>The CPUSurplusCreditBalance is paid down by earned CPU credits. If the number of surplus credits exceeds the maximum number of credits 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>
CPUSurplusCreditsCharged	<p>[T2 Unlimited instances] The number of spent surplus credits that are not paid down by earned CPU credits, and 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 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>Units: Credits (vCPU-minutes)</p>

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 upon a selected instance.</p> <p>To use the percentiles statistic, you must enable detailed monitoring.</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>

Metric	Description
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>
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>

Metric	Description
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>

The AWS/EC2 namespace includes the following status checks 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](#).

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>

Metric	Description
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>

The AWS/EC2 namespace includes the following Amazon EBS metrics for your C5, C5d, M5, M5d, R5, R5d, and z1d instances.

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> <p>Unit: Count</p>
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	Bytes written to all EBS volumes attached to the instance in a specified period of time.

Metric	Description
	<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 C5, C5d, M5, M5d, R5, R5d, and z1d 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 C5, C5d, M5, M5d, R5, R5d, and z1d 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 \(p. 755\)](#). For information about the metrics provided for your Spot fleets, see [CloudWatch Metrics for Spot Fleet \(p. 314\)](#).

Amazon EC2 Dimensions

You can use the following dimensions to refine the metrics returned for your instances.

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 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.
ImageId	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.
InstanceId	This dimension filters the data you request for the identified instance only. This helps you pinpoint an exact instance from which to monitor data.
InstanceType	This dimension filters the data you request for all instances running with this specified instance type. This helps you categorize

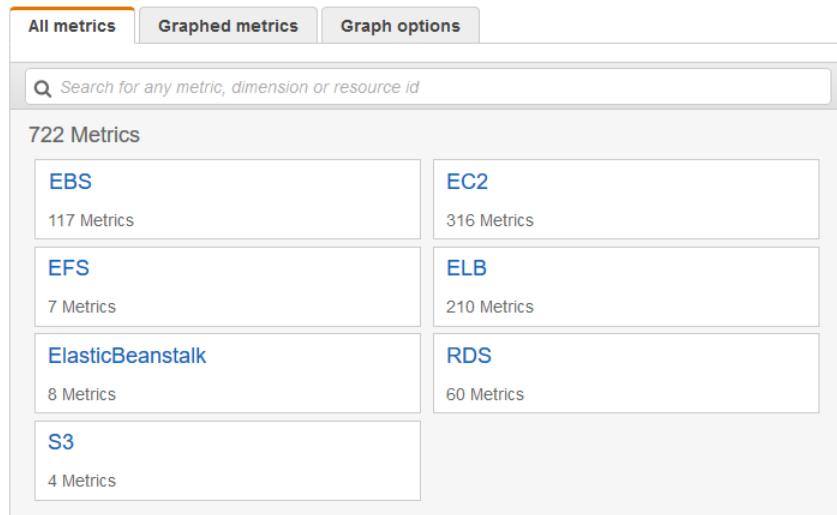
Dimension	Description
	your data by the type of instance running. For example, you might compare data from an m1.small instance and an m1.large instance to determine which has the better business value for your application. Available for instances with Detailed Monitoring enabled.

List Available Metrics

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

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Select the EC2 metric namespace.



4. Select a metric dimension (for example, Per-Instance Metrics).

All metrics Graphed metrics Graph options

All > EC2 Search for any metric, dimension or resource id

103 Metrics

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**.

	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

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": "CPUUtilization"  
        },  
        {  
            "Namespace": "AWS/EC2",  
            "Dimensions": [  
                {  
                    "Name": "InstanceId",  
                    "Value": "i-1234567890abcdef0"  
                }  
            ],  
            "MetricName": "NetworkIn"  
        },  
        ...  
    ]  
}
```

To list all the available metrics for an instance

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

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 \(p. 494\)](#)
- [Get Statistics for a Specific Instance \(p. 494\)](#)

- [Aggregate Statistics Across Instances \(p. 497\)](#)
- [Aggregate Statistics by Auto Scaling Group \(p. 499\)](#)
- [Aggregate Statistics by AMI \(p. 499\)](#)

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.
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 Disable Detailed Monitoring for Your Instances \(p. 483\)](#).

To display the CPU utilization for a specific instance using the console

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Select the EC2 metric namespace.

The screenshot shows the 'All metrics' tab selected in the top navigation bar. A search bar is present at the top. Below it, a summary section says '722 Metrics'. A grid of service metrics is displayed:

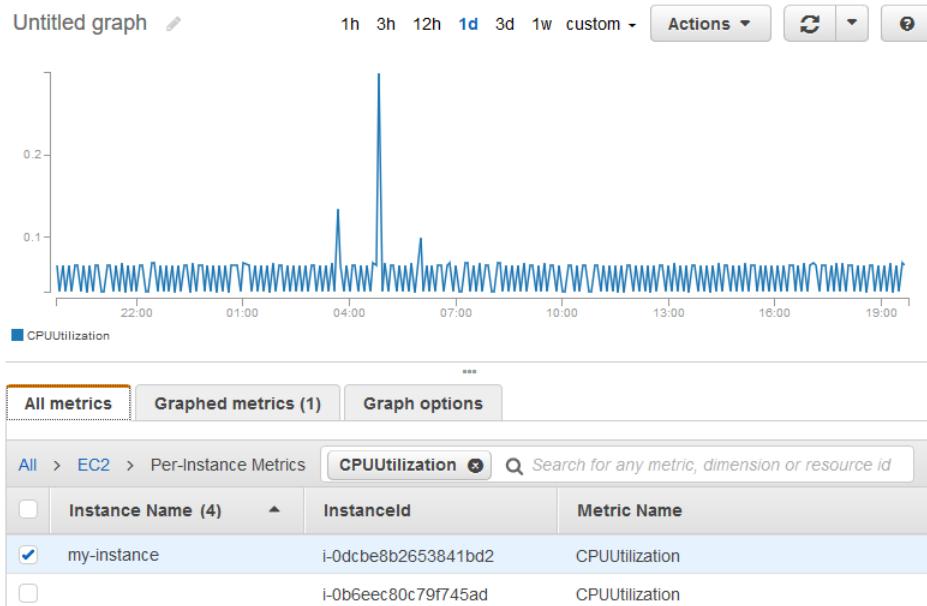
EBS	EC2
117 Metrics	316 Metrics
EFS	ELB
7 Metrics	210 Metrics
ElasticBeanstalk	RDS
8 Metrics	60 Metrics
S3	
4 Metrics	

4. Select the Per-Instance Metrics dimension.

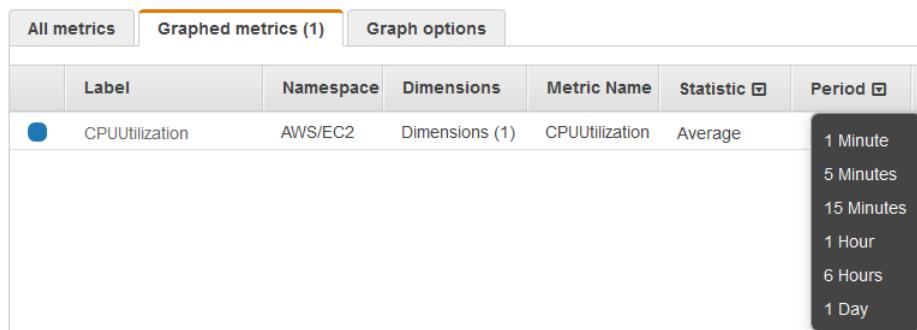
The screenshot shows the 'Graphed metrics' tab selected in the top navigation bar, with 'EC2' selected in the breadcrumb trail. A search bar is present at the top. Below it, a summary section says '103 Metrics'. A list of dimension options is displayed:

- By Auto Scaling Group
- By Image (AMI) Id
- Per-Instance Metrics
- Aggregated by Instance Type
- Across All Instances

5. In the search field, type **CPUUtilization** and press Enter. Select 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 using the 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.

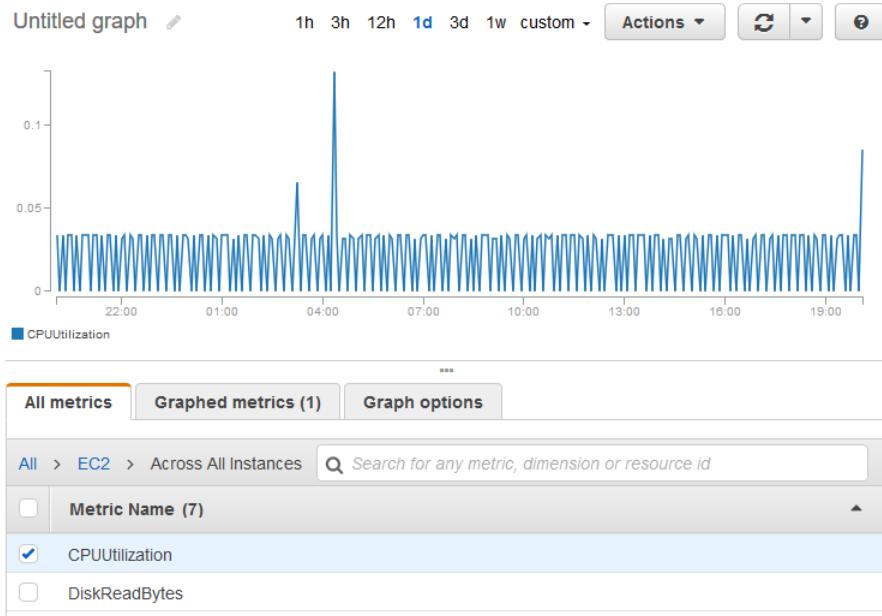
This example shows you how to use detailed monitoring to get the average CPU usage for your EC2 instances. Because no dimension is specified, CloudWatch returns statistics for all dimensions in the AWS/EC2 namespace.

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

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Select the **EC2** namespace and then select **Across All Instances**.
4. Select 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

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": 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 using the console

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Select the **EC2** namespace and then select **By Auto Scaling Group**.
4. Select 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 using the 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 Disable Detailed Monitoring for Your Instances \(p. 483\)](#).

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 using the console

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Select the **EC2** namespace and then select **By Image (AMI) Id**.
4. Select 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

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 \(p. 485\)](#).

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

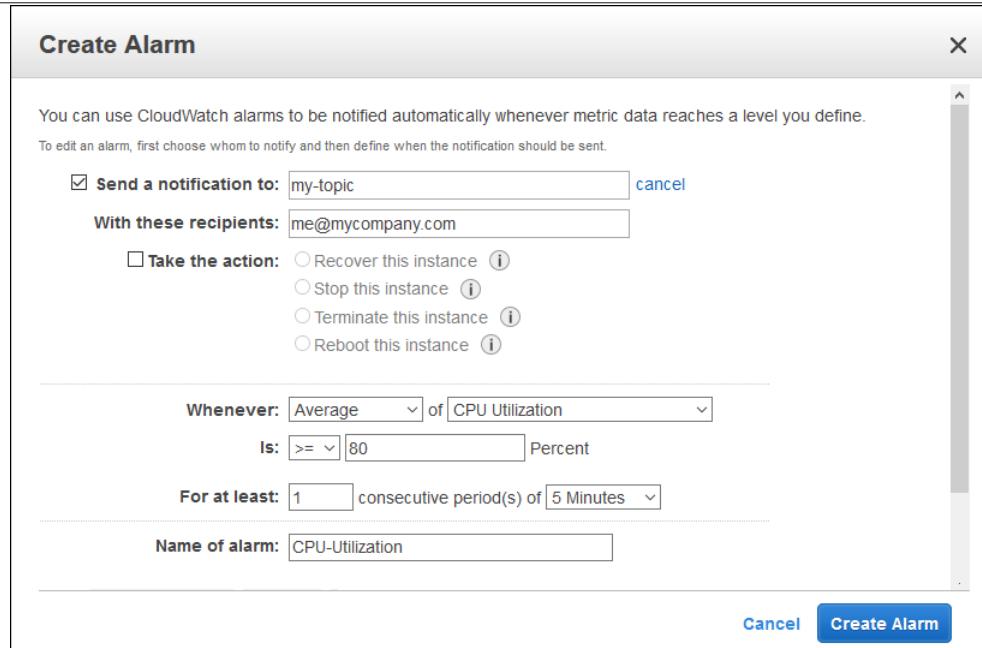
You can create a CloudWatch alarm that monitors CloudWatch metrics for one of your instances. CloudWatch will automatically send you a notification when the metric reaches a threshold you specify. You can create a CloudWatch alarm using the Amazon EC2 console, or using the more advanced options provided 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*.

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, choose **Create Alarm**.
5. In the **Create Alarm** dialog box, do the following:
 - a. Choose **create topic**. For **Send a notification to**, type a name for the SNS topic. For **With these recipients**, type 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 \geq and type 80 percent. For **For at least**, type 1 consecutive period of 5 Minutes.
 - c. Choose **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 `AWSServiceRoleForCloudWatchEvents` 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:

- `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 [Permissions and Policies](#) in the *IAM User Guide*.

Contents

- [Adding Stop Actions to Amazon CloudWatch Alarms \(p. 503\)](#)
- [Adding Terminate Actions to Amazon CloudWatch Alarms \(p. 504\)](#)
- [Adding Reboot Actions to Amazon CloudWatch Alarms \(p. 505\)](#)
- [Adding Recover Actions to Amazon CloudWatch Alarms \(p. 505\)](#)
- [Using the Amazon CloudWatch Console to View Alarm and Action History \(p. 507\)](#)
- [Amazon CloudWatch Alarm Action Scenarios \(p. 507\)](#)

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 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. 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**, type a name for the topic, and then for **With these recipients**, type 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, type **10** percent.
- e. For **For at least**, specify the evaluation period for the alarm. In this example, type **24** consecutive period(s) of **1 Hour**.
- f. To change the name of the alarm, for **Name of alarm**, type a new name. Alarm names must contain only ASCII characters.

If you don't type 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 for an Instance](#) in the *Amazon EC2 User Guide for Linux Instances*.

To create an alarm to terminate an idle instance 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. 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**, type a name for the topic, and then for **With these recipients**, type 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, type **10** percent.
 - e. For **For at least**, specify the evaluation period for the alarm. In this example, type **24** consecutive period(s) of **1 Hour**.
 - f. To change the name of the alarm, for **Name of alarm**, type a new name. Alarm names must contain only ASCII characters.

If you don't type 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 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. 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**, type a name for the topic, and for **With these recipients**, type 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, type **3 consecutive period(s) of 1 Minute**.
 - e. To change the name of the alarm, for **Name of alarm**, type a new name. Alarm names must contain only ASCII characters.

If you don't type 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.

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 a C3, C4, C5, M3, M4, M5, R3, R4, R5, T2, or X1 instance
- Run in a VPC (not EC2-Classic)
- 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 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. 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**, type a name for the topic, and for **With these recipients**, type 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.

- 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, type **2** consecutive period(s) of **1 Minute**.
- e. To change the name of the alarm, for **Name of alarm**, type a new name. Alarm names must contain only ASCII characters.

If you don't type 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

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 define.

To edit an alarm, first choose whom to notify and then define when the notification should be sent.

Send a notification to: [create topic](#)

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:

CPU Utilization Percent

7/21 22:00 7/22 00:00 7/22 02:00

Time	CPU Utilization (%)
7/21 22:00	0
7/22 00:00	0
7/22 02:00	0

[Cancel](#)

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

Setting	Value
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
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	<=

Setting	Value
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 AWS SMS 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

New CloudWatch Agent Available

A new multi-platform CloudWatch agent is available. You can use a single agent to collect both system metrics and log files from Amazon EC2 instances and on-premises servers. The new 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 you use the new agent instead of the older monitoring scripts to collect metrics and logs. For more information about the CloudWatch agent, see [Collect Metrics from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent](#) in the [Amazon CloudWatch User Guide](#).

The rest of this section is informational for customers who are still using the older Perl scripts for monitoring. You can download these [Amazon CloudWatch Monitoring Scripts for Linux](#) from the AWS sample code library.

CloudWatch Monitoring Scripts

The Amazon CloudWatch Monitoring Scripts for Amazon Elastic Compute Cloud (Amazon EC2) Linux-based instances demonstrate how to produce and consume Amazon CloudWatch custom metrics. 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 \(p. 512\)](#)
- [Package Contents \(p. 512\)](#)
- [Prerequisites \(p. 513\)](#)
- [Getting Started \(p. 514\)](#)
- [mon-put-instance-data.pl \(p. 515\)](#)
- [mon-get-instance-stats.pl \(p. 518\)](#)
- [Viewing Your Custom Metrics in the Console \(p. 519\)](#)
- [Troubleshooting \(p. 519\)](#)

Supported Systems

These monitoring scripts are intended for use with Amazon EC2 instances running Linux. The scripts have been tested on instances using the following Amazon Machine Images (AMIs), both 32-bit and 64-bit versions:

- Amazon Linux 2
- Amazon Linux AMI 2014.09.2 and later
- Red Hat Enterprise Linux 7.4 and 6.9
- SUSE Linux Enterprise Server 12
- Ubuntu Server 16.04 and 14.04

Note

On servers running SUSE Linux Enterprise Server 12, you may need to first download the `perl-Switch` package. You can download and install this package with 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
```

You can also monitor memory and disk metrics on Amazon EC2 instances running Windows by sending this data to CloudWatch Logs. For more information, see [Sending Logs, Events, and Performance Counters to Amazon CloudWatch](#) in the *Amazon EC2 User Guide for Windows Instances*.

Package Contents

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.

Prerequisites

With some versions of Linux, you must install additional modules before the monitoring scripts will work.

Amazon Linux 2 and Amazon Linux AMI

To install the required packages

1. Log on to your instance. For more information, see [Connect to Your Linux Instance \(p. 397\)](#).
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
```

Red Hat Enterprise Linux

You must install additional Perl modules.

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 \(p. 397\)](#).
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[1]> install LWP::Protocol::https  
cpan[1]> install Sys::Syslog  
cpan[1]> install Switch
```

To install the required packages on Red Hat Enterprise Linux 7.4

1. Log on to your instance. For more information, see [Connect to Your Linux Instance \(p. 397\)](#).
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
```

SUSE Linux Enterprise Server

You must install additional Perl modules.

To install the required packages on SUSE

1. Log on to your instance. For more information, see [Connect to Your Linux Instance \(p. 397\)](#).
2. At a command prompt, install packages as follows:

```
sudo zypper install perl-Switch perl-Datetime  
sudo zypper install -y "perl(LWP::Protocol::https)"
```

Ubuntu Server

You must configure your server as follows.

To install the required packages on Ubuntu

1. Log on to your instance. For more information, see [Connect to Your Linux Instance \(p. 397\)](#).
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
```

Getting Started

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
```

3. Ensure that the scripts have permission to perform CloudWatch operations using one of the following options:

- If you associated an IAM role (instance profile) with your instance, verify that it grants permissions to perform the following operations:
 - cloudwatch:PutMetricData
 - cloudwatch:GetMetricStatistics
 - cloudwatch>ListMetrics
 - ec2:DescribeTags
- Specify your AWS credentials in a credentials file. First, copy the `awscreds.template` file included with the monitoring scripts to `awscreds.conf` as follows:

```
cp awscreds.template awscreds.conf
```

Add the following content to the `awscreds.conf` file:

```
AWSAccessKeyId=my-access-key-id
AWSSecretKey=my-secret-access-key
```

For information about how to view your AWS credentials, see [Understanding and Getting Your Security Credentials](#) in the *Amazon Web Services General Reference*.

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
<code>--mem-util</code>	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 <code>--mem-used-incl-cache-buff</code> option.
<code>--mem-used</code>	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 <code>--mem-used-incl-cache-buff</code> option.
<code>--mem-used-incl-cache-buff</code>	If you include this option, memory currently used for cache and buffers is counted as "used" when the metrics are reported for <code>--mem-util</code> , <code>--mem-used</code> , and <code>--mem-avail</code> .
<code>--mem-avail</code>	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 <code>--mem-used-incl-cache-buff</code> option.
<code>--swap-util</code>	Collects and sends SwapUtilization metrics, reported in percentages.

Name	Description
--swap-used	Collects and sends SwapUsed metrics, reported in megabytes.
--disk-path=PATH	<p>Selects the disk on which to report.</p> <p>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.</p> <p>To select a disk for the filesystems mounted on / and /home, use the following parameters:</p> <pre>--disk-path=/ --disk-path=/home</pre>
--disk-space-util	<p>Collects and sends the DiskSpaceUtilization metric for the selected disks. The metric is reported in percentages.</p> <p>Note that the disk utilization metrics calculated by this script differ from the values calculated by the df -k -l command. If you find the values from df -k -l more useful, you can change the calculations in the script.</p>
--disk-space-used	<p>Collects and sends the DiskSpaceUsed 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 DiskSpaceAvailable 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 --aws-access-key-id and --aws-secret-key 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 --aws-secret-key option. Do not use this option with the --aws-credential-file 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 --aws-access-key-id option. Do not use this option with --aws-credential-file parameter.

Name	Description
--aws-iam-role=VALUE	<p>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.</p> <p>Do not use this option with the --aws-credential-file, --aws-access-key-id, or --aws-secret-key parameters.</p>
--aggregated[=only]	Adds aggregated metrics for instance type, AMI ID, and overall for the region. The value =only is optional; if specified, the script reports only aggregated metrics.
--auto-scaling[=only]	Adds aggregated metrics for the Auto Scaling group. The value =only 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.

To perform a simple test run without posting data to CloudWatch

```
./mon-put-instance-data.pl --mem-util --verify --verbose
```

To collect all available memory metrics and send 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
```

To set a cron schedule for metrics reported to CloudWatch

1. Start editing the crontab using the following command:

```
crontab -e
```

2. 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, the script will write the error message in the system log.

To collect aggregated metrics for an Auto Scaling group and send them to Amazon CloudWatch without reporting individual instance metrics

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail --auto-scaling=only
```

To collect aggregated metrics for instance type, AMI ID and region, and send them to Amazon CloudWatch without reporting individual instance metrics

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail --aggregated=only
```

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 that collects the metrics, prepares a complete HTTP request, but does not actually call CloudWatch

Name	Description
	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.
--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
```

Network and Security

Amazon EC2 provides the following network and security features.

Features

- [Amazon EC2 Key Pairs \(p. 521\)](#)
- [Amazon EC2 Security Groups for Linux Instances \(p. 531\)](#)
- [Controlling Access to Amazon EC2 Resources \(p. 545\)](#)
- [Amazon EC2 and Amazon Virtual Private Cloud \(p. 630\)](#)
- [Amazon EC2 Instance IP Addressing \(p. 658\)](#)
- [Elastic IP Addresses \(p. 674\)](#)
- [Elastic Network Interfaces \(p. 683\)](#)
- [Placement Groups \(p. 701\)](#)
- [Network Maximum Transmission Unit \(MTU\) for Your EC2 Instance \(p. 706\)](#)
- [Enhanced Networking on Linux \(p. 708\)](#)

If you access Amazon EC2 using the command line tools or an API, you'll need your access key ID and secret access key. For more information, see [How Do I Get Security Credentials?](#) in the *Amazon Web Services General Reference*.

You can launch an instance into one of two platforms: EC2-Classic or EC2-VPC. An instance that's launched into EC2-Classic or a default VPC is automatically assigned a public IP address. An instance that's launched into a nondefault VPC can be assigned a public IP address on launch. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms \(p. 637\)](#).

Instances can fail or terminate for reasons outside of your control. If an instance fails and you launch a replacement instance, the replacement has a different public IP address than the original. However, if your application needs a static IP address, you can use an *Elastic IP address*.

You can use *security groups* to control who can access your instances. These are analogous to an inbound network firewall that enables you to specify the protocols, ports, and source IP ranges that are allowed to reach your instances. You can create multiple security groups and assign different rules to each group. You can then assign each instance to one or more security groups, and we use the rules to determine which traffic is allowed to reach the instance. You can configure a security group so that only specific IP addresses or specific security groups have access to the instance.

Amazon EC2 Key Pairs

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. Public–key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a *key pair*.

To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance. On a Linux instance, the

public key content is placed in an entry within `~/.ssh/authorized_keys`. This is done at boot time and enables you to securely access your instance using the private key instead of a password.

Creating a Key Pair

You can use Amazon EC2 to create your key pair. For more information, see [Creating a Key Pair Using Amazon EC2 \(p. 523\)](#).

Alternatively, you could use a third-party tool and then import the public key to Amazon EC2. For more information, see [Importing Your Own Public Key to Amazon EC2 \(p. 523\)](#).

Each key pair requires a name. Be sure to choose a name that is easy to remember. Amazon EC2 associates the public key with the name that you specify as the key name.

Amazon EC2 stores the public key only, and you store the private key. Anyone who possesses your private key can decrypt your login information, so it's important that you store your private keys in a secure place.

The keys that Amazon EC2 uses are 2048-bit SSH-2 RSA keys. You can have up to five thousand key pairs per region.

Launching and Connecting to Your Instance

When you launch an instance, you should specify the name of the key pair you plan to use to connect to the instance. If you don't specify the name of an existing key pair when you launch an instance, you won't be able to connect to the instance. When you connect to the instance, you must specify the private key that corresponds to the key pair you specified when you launched the instance.

Note

Amazon EC2 doesn't keep a copy of your private key; therefore, if you lose a private key, there is no way to recover it. If you lose the private key for an instance store-backed instance, you can't access the instance; you should terminate the instance and launch another instance using a new key pair. If you lose the private key for an EBS-backed Linux instance, you can regain access to your instance. For more information, see [Connecting to Your Linux Instance if You Lose Your Private Key \(p. 528\)](#).

Key Pairs for Multiple Users

If you have several users that require access to a single instance, you can add user accounts to your instance. For more information, see [Managing User Accounts on Your Linux Instance \(p. 432\)](#). You can create a key pair for each user, and add the public key information from each key pair to the `.ssh/authorized_keys` file for each user on your instance. You can then distribute the private key files to your users. That way, you do not have to distribute the same private key file that's used for the root account to multiple users.

Contents

- [Creating a Key Pair Using Amazon EC2 \(p. 523\)](#)
- [Importing Your Own Public Key to Amazon EC2 \(p. 523\)](#)
- [Retrieving the Public Key for Your Key Pair on Linux \(p. 525\)](#)
- [Retrieving the Public Key for Your Key Pair on Windows \(p. 525\)](#)
- [Retrieving the Public Key for Your Key Pair From Your Instance \(p. 525\)](#)
- [Verifying Your Key Pair's Fingerprint \(p. 526\)](#)
- [Deleting Your Key Pair \(p. 526\)](#)
- [Adding or Replacing a Key Pair for Your Instance \(p. 527\)](#)
- [Connecting to Your Linux Instance if You Lose Your Private Key \(p. 528\)](#)

Creating a Key Pair Using Amazon EC2

You can create a key pair using the Amazon EC2 console or the command line. After you create a key pair, you can specify it when you launch your instance. You can also add the key pair to a running instance to enable another user to connect to the instance. For more information, see [Adding or Replacing a Key Pair for Your Instance \(p. 527\)](#).

To create your key pair using the Amazon EC2 console

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. Enter a name for the new key pair in the **Key pair name** field of the **Create Key Pair** dialog box, and then choose **Create**.
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. 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.

6. If you will use an SSH client on a Mac 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 \(p. 914\)](#).

To create your key pair 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-key-pair \(AWS CLI\)](#)
- [New-EC2KeyPair \(AWS Tools for Windows PowerShell\)](#)

Importing 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.

Amazon EC2 accepts the following formats:

- OpenSSH public key format (the format in `~/.ssh/authorized_keys`)

- Base64 encoded DER format
- SSH public key file format as specified in [RFC4716](#)

Amazon EC2 does not accept DSA keys. Make sure your key generator is set up to create RSA keys.

Supported lengths: 1024, 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.

Use the following steps to import your key pair using the Amazon EC2 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**.

To import the public key using the command line

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

- [import-key-pair](#) (AWS CLI)
- [Import-EC2KeyPair](#) (AWS Tools for Windows PowerShell)

After the public key file is imported, you can verify that the key pair was imported successfully using the Amazon EC2 console as follows.

To verify that your key pair was imported

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region in which you created the key pair.
3. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
4. Verify that the key pair that you imported is in the displayed list of key pairs.

To view your key pair 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-key-pairs](#) (AWS CLI)

- [Get-EC2KeyPair](#) (AWS Tools for Windows PowerShell)

Retrieving the Public Key for Your Key Pair on Linux

On your local Linux or Mac computer, you can use the **ssh-keygen** command to retrieve the public key for your key pair.

To retrieve the public key from your computer

1. Use the **ssh-keygen** command on a computer to which you've downloaded your private key:

```
ssh-keygen -y
```

2. When prompted to enter the file in which the key is, specify the path to your .pem file; for example:

```
/path_to_key_pair/my-key-pair.pem
```

3. The command returns the public key:

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4xyyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUzofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb70z1PnWOyN0qFU0XA246RA8QFYiCNYW13f05p6KLxEXAMPLE
```

If this command fails, ensure that you've changed the permissions on your key pair file so that only you can view it by running the following command:

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

Retrieving the Public Key for Your Key Pair on Windows

On your local Windows computer, you can use PuTTYgen to get the public key for your key pair.

Start PuTTYgen, choose **Load**, and select the .ppk or .pem file. PuTTYgen displays the public key.

Retrieving the Public Key for Your Key Pair From Your Instance

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:

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4xyyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUzofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb70z1PnWOyN0qFU0XA246RA8QFYiCNYW13f05p6KLxEXAMPLE
```

```
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE 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; you'll continue to see the public key for the key pair you specified when you launched the instance in the instance metadata.

For more information, see [Retrieving Instance Metadata \(p. 454\)](#).

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. For example:

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4xyyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBITntckkiJ7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+OFzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcxkvJdWPkYQS3xqC0+FmUzofz221CBt5IMucxxPkJ4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

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 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 that's stored in AWS.

If you created your key pair using AWS, you can use the OpenSSL tools to generate a fingerprint from the private key file:

```
$ openssl pkcs8 -in path_to_private_key -inform PEM -outform DER -topk8 -nocrypt | openssl
sha1 -c
```

If you created your key pair using a third-party tool and uploaded the public key to AWS, you can use the OpenSSL tools to generate a fingerprint from the private key file on your local machine:

```
$ openssl rsa -in path_to_private_key -pubout -outform DER | openssl md5 -c
```

The output should match the fingerprint that's displayed in the console.

Deleting Your Key Pair

When you delete a key pair, you are only deleting Amazon EC2's 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 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.

Note

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 the Amazon EC2 console or the command line.

To delete your key pair using the console

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**.

To delete your key pair 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-key-pair \(AWS CLI\)](#)
- [Remove-EC2KeyPair \(AWS Tools for Windows PowerShell\)](#)

Note

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 Linux Instance \(p. 432\)](#).

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.

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 Linux Instance \(p. 432\)](#).

Before you begin, create a new key pair using [the Amazon EC2 console \(p. 523\)](#) or a [third-party tool \(p. 523\)](#).

To add or replace a key pair

1. Retrieve the public key from your new key pair. For more information, see [Retrieving the Public Key for Your Key Pair on Linux \(p. 525\)](#) or [Retrieving the Public Key for Your Key Pair on Windows \(p. 525\)](#).
2. Connect to your instance using your existing private key file.
3. 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.
4. Disconnect from your instance, and test that you can connect to your instance using the new private key file.

5. (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 (for example, in an Elastic Beanstalk environment), ensure that the key pair you're replacing 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.

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, 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 \(p. 350\)](#).

This procedure isn't supported for instance store-backed instances. 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 must have the private key in order to connect to the instance.

Prerequisites

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.

To connect to an EBS-backed instance with a different key pair

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 **Details** tab, save the following information that you'll 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`).
 - [EC2-Classic] If the original instance has an associated Elastic IP address, write down the Elastic IP address shown in the **Elastic IP** field in the details pane.
4. 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. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

5. 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 \(p. 101\)](#).
 - 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 instance you'd like to connect to. 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**.
6. 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**, and then select **Detach Volume**. Wait for the state of the volume to become available. (You might need to choose the **Refresh** icon.)
7. 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 **Yes, 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.

8. Connect to the temporary instance.
9. 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 may appear differently on your instance. For example, devices mounted as `/dev/sdf` may show up as `/dev/xvdf` on the instance. Some versions of Red Hat (or its variants, such as CentOS) may even increment the trailing letter by 4 characters, where `/dev/sdf` becomes `/dev/xvdk`.

- 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 above 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 you identified earlier.

```
[ec2-user ~]$ sudo mount /dev/xvdf1 /mnt/tempvol
```

10. 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 may 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'll need to update the file using **sudo** and then check the permissions on the file to verify that you'll be 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
```

- 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
```

- From the Amazon EC2 console, select the volume with the volume ID that you wrote down, choose **Actions**, and then select **Detach Volume**. Wait for the state of the volume to become available. (You might need to choose the **Refresh** icon.)
- With the volume still selected, choose **Actions**, **Attach Volume**. Select the instance ID of the original instance, specify the device name you noted earlier for the original root device attachment (/dev/sda1 or /dev/xvda), and then choose **Yes, 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.

- 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 to 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.

- [EC2-Classic] If the original instance had an associated Elastic IP address before you stopped it, you must re-associate it with the instance as follows:

- In the navigation pane, choose **Elastic IPs**.

- b. Select the Elastic IP address that you wrote down at the beginning of this procedure.
 - c. Choose **Actions**, and then select **Associate address**.
 - d. Select the ID of the original instance, and then choose **Associate**.
16. (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**.

Amazon EC2 Security Groups for Linux Instances

A *security group* acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You 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; the new rules are automatically applied to all instances that are associated with the security group after a short period. When we decide whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups that are associated with the instance.

If you need to allow traffic to a Windows instance, see [Amazon EC2 Security Groups for Windows Instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

Topics

- [Security Groups for EC2-Classic \(p. 531\)](#)
- [Security Groups for EC2-VPC \(p. 531\)](#)
- [Security Group Rules \(p. 532\)](#)
- [Default Security Groups \(p. 534\)](#)
- [Custom Security Groups \(p. 535\)](#)
- [Working with Security Groups \(p. 535\)](#)
- [Security Group Rules Reference \(p. 540\)](#)

If you have requirements that aren't met by security groups, you can maintain your own firewall on any of your instances in addition to using security groups.

Your account may support EC2-Classic in some regions, depending on when you created it. For more information, see [Supported Platforms \(p. 637\)](#). Security groups for EC2-Classic are separate to security groups for EC2-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.

In EC2-Classic, you can have up to 500 security groups in each region for each account. You can associate an instance with up to 500 security groups and add up to 100 rules to a security group.

Security Groups for EC2-VPC

When you launch an instance in a VPC, you must specify a security group that's created for the VPC. If your account supports EC2-Classic, you can't specify a security group that you created for EC2-Classic

when you launch an instance in a VPC. Security groups for EC2-VPC have additional capabilities that aren't supported by security groups for EC2-Classic. For more information, see [Differences Between Security Groups for EC2-Classic and EC2-VPC](#) in the *Amazon VPC User Guide*.

After you launch an instance in a VPC, 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 [Changing the Security Group \(p. 696\)](#).

Security groups for EC2-VPC have separate limits. For more information, see [Amazon VPC Limits](#) in the *Amazon VPC User Guide*. The security groups for EC2-Classic do not count against the security group limit for EC2-VPC.

Your VPC can be enabled for IPv6. For more information, see [IP addressing in Your VPC](#) in the *Amazon VPC User Guide*. You can add rules to your VPC security groups to enable inbound and outbound IPv6 traffic.

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 and 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.
- You can't change the outbound rules for an EC2-Classic security group.
- Security group rules are always permissive; you can't create rules that deny access.
- 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 \(p. 533\)](#).
- You can add and remove rules at any time. Your changes are automatically applied to the instances associated with the security group after a short period.

Note

The effect of some rule changes may depend on how the traffic is tracked. For more information, see [Connection Tracking \(p. 533\)](#).

- When you associate multiple security groups with an instance, the rules from each security group are effectively aggregated to create one set of rules. We use this set of rules to determine whether to allow access.

Note

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:

- **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.

- (VPC only) 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.
- (VPC only) A range of IPv6 addresses, in CIDR block notation, for example, 2001:db8:1234:1a00::/64.
- (VPC only) The prefix list ID for the AWS service; for example, pl-1a2b3c4d. For more information, see [Gateway VPC Endpoints](#) in the *Amazon VPC User Guide*.
- Another security group. This allows instances associated with the specified security group to access instances associated with this security group. This 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.
 - EC2-Classic: A different security group for EC2-Classic in the same region.
 - EC2-Classic: A security group for another AWS account in the same region (add the AWS account ID as a prefix; for example, 111122223333/sg-edcd9784).
 - EC2-VPC: A different security group for the same VPC or a peer VPC in a VPC peering connection.
- **(Optional) Description:** You can add a description for the rule; for example, to 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 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 \(p. 658\)](#). 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, we apply 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 allows security groups to be stateful — 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) and there is a corresponding rule in the other direction that permits all response traffic (0.0.0.0/0) 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 an outbound rule that allows all outbound 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
ICMP	All	0.0.0.0/0
Outbound rules		
Protocol type	Port number	Destination IP
All	All	0.0.0.0/0

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). ICMP traffic is always tracked, regardless of rules. If you remove the outbound rule from the security group, then all traffic to and from the instance is tracked, including traffic on port 80 (HTTP).

An existing flow of traffic that is tracked may not be interrupted when you remove the security group rule that enables that flow. Instead, the flow is interrupted when it's stopped by you or the other host for at least a few minutes (or up to 5 days for established TCP connections). For UDP, this may require terminating actions on the remote side of the flow. An untracked flow of traffic is immediately interrupted if the rule that enables the flow is removed or modified. For example, if you remove a rule that allows all inbound SSH traffic to the instance, then your existing SSH connections to the instance are immediately dropped.

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, because it's regarded as response traffic.

For VPC security groups, 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* per VPC and per region for EC2-Classic. If you don't specify a security group when you launch an instance, the instance is automatically associated with the default security group.

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 the inbound rules for any EC2-Classic default security group. You can add or remove inbound and outbound rules for any VPC default security group.

You can't delete a default security group. If you try to delete the EC2-Classic default security group, you'll get the following error: `Client.InvalidGroup.Reserved: The security group 'default' is reserved.` If you try to delete a VPC default security group, you'll get 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:

- EC2-Classic: ASCII characters
- EC2-VPC: a-z, A-Z, 0-9, spaces, and ._-:/()#@[]+=&{}!\$*

A security group name cannot start with sg-. For EC2-Classic, the security group name must be unique within your account for the region. For EC2-VPC, the name must be unique within 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. In EC2-VPC, you can also change its outbound rules.

For more information about the types of rules you can add to security groups, see [Security Group Rules Reference \(p. 540\)](#).

Working with Security Groups

You can create, view, update, and delete security groups and security group rules using the Amazon EC2 console.

Contents

- [Creating a Security Group \(p. 535\)](#)
- [Describing Your Security Groups \(p. 536\)](#)
- [Adding Rules to a Security Group \(p. 537\)](#)
- [Updating Security Group Rules \(p. 538\)](#)
- [Deleting Rules from a Security Group \(p. 539\)](#)
- [Deleting a Security Group \(p. 539\)](#)

Creating a Security Group

You can create a custom security group using the Amazon EC2 console. For EC2-VPC, you must specify the VPC for which you're creating the security group.

To create a new security group using the console

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. (EC2-Classic only) To create a security group for use in EC2-Classic, choose **No VPC**.
(EC2-VPC) For **VPC**, choose a VPC ID to create a security group for that 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 \(p. 537\)](#).

To create a security group using the command line

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

The Amazon EC2 console enables you to copy the rules from an existing security group to a new security group.

To copy a security group using the console

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. In the **VPC** list, choose **No VPC** to create a security group for EC2-Classic, or choose a VPC ID to create a security group for that VPC. When you are done, choose **Create**.

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 in EC2-Classic, you can't change its security groups. After you launch an instance in a VPC, you can change its security groups. For more information, see [Changing an Instance's Security Groups](#) in the *Amazon VPC User Guide*.

[EC2-VPC] To modify the security groups for an instance using the command line

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Describing Your Security Groups

You can view information about your security groups using the Amazon EC2 console or the command line.

To describe your security groups for EC2-Classic using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select **Network Platforms** from the filter list, then choose **EC2-Classic**.

4. Select a security group. The **Description** tab displays general information. The **Inbound** tab displays the inbound rules.

To describe your security groups for EC2-VPC using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**.
3. Select **Network Platforms** from the filter list, then choose **EC2-VPC**.
4. Select a security group. We display general information in the **Description** tab, inbound rules on the **Inbound** tab, and outbound rules on the **Outbound** tab.

To describe one or more security groups using the command line

- [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 associated with the security group after a short period.

For more information about choosing security group rules for specific types of access, see [Security Group Rules Reference \(p. 540\)](#).

To add rules to a security group using the console

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 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.

Note

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) 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 \(p. 540\)](#).

5. Choose **Save**.

6. For a VPC security group, 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.

Note

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 IP address of your local computer.
- For **Description**, you can optionally specify a description for the rule.

7. Choose **Save**.

To add one or more ingress rules to a security group using the command line

- [authorize-security-group-ingress](#) (AWS CLI)
- [Grant-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

[EC2-VPC] To add one or more egress rules to a security group using the command line

- [authorize-security-group-egress](#) (AWS CLI)
- [Grant-EC2SecurityGroupEgress](#) (AWS Tools for Windows PowerShell)

Updating Security Group Rules

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 using the console

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 **Inbound Rules** to update a rule for inbound traffic or **Outbound Rules** to update a rule for outbound traffic.
4. Choose **Edit**. Modify the rule entry as required and choose **Save**.

To update the protocol, port range, or source or destination of an existing rule using the Amazon EC2 API or a command line tool, you cannot modify the rule. Instead, you must delete the existing rule and add a new rule. To update the rule description only, you can use the [update-security-group-rule-descriptions-ingress](#) and [update-security-group-rule-descriptions-egress](#) commands.

To update the description for an ingress security group rule using the command line

- [update-security-group-rule-descriptions-ingress](#) (AWS CLI)
- [Update-EC2SecurityGroupRuleIngressDescription](#) (AWS Tools for Windows PowerShell)

[EC2-VPC] To update the description for an egress security group rule using the command line

- [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 after a short period.

To delete a security group rule using the console

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**.

To remove one or more ingress rules from a security group using the command line

- [revoke-security-group-ingress](#) (AWS CLI)
- [Revoke-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

[EC2-VPC] To remove one or more egress rules from a security group using the command line

- [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.

To delete a security group using the console

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**.

To delete a security group using the command line

- [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, and 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.

Topics

- [Web server \(p. 540\)](#)
- [Database server \(p. 540\)](#)
- [Access from another instance in the same group \(p. 542\)](#)
- [Access from local computer \(p. 542\)](#)
- [Path MTU Discovery \(p. 543\)](#)
- [Ping your instance \(p. 543\)](#)
- [DNS server \(p. 544\)](#)
- [Amazon EFS file system \(p. 544\)](#)
- [Elastic Load Balancing \(p. 544\)](#)

Web server

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	(VPC only) Allows inbound HTTP access from any IPv6 address
TCP	6	443 (HTTPS)	::/0	(VPC only) Allows inbound HTTPS access from any IPv6 address

Database server

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 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
TCP	6	1521 (Oracle)	The default port to access an Oracle database, for example, on an Amazon RDS instance

(VPC only) You can optionally restrict outbound traffic from your database servers, for example, if you want 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

Protocol type	Protocol number	Port	Destination IP	Notes
				outbound HTTPS access to any IPv6 address

Access from another instance in the same 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 VPC 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

The following table describes inbound rules for an EC2-Classic security group that enable associated instances to communicate with each other. The rules allow all types of traffic.

Protocol type	Protocol number	Ports	Source IP
ICMP	1	-1 (All)	The ID of the security group
TCP	6	0 - 65535 (All)	The ID of the security group
UDP	17	0 - 65535 (All)	The ID of the security group

Access from local 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 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 range of IP addresses

Protocol type	Protocol number	Port	Source IP
			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.

Path MTU Discovery

The path MTU is the maximum packet size that's supported on the path 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 returns the following ICMP message:

```
Destination Unreachable: Fragmentation Needed and Don't Fragment was Set
```

To ensure that your instance can receive this message and the packet does not get dropped, you must add an ICMP rule to your inbound security group rules.

Protocol type	Protocol number	ICMP type	ICMP code	Source IP
ICMP	1	3 (Destination Unreachable)	4 (Fragmentation Needed and Don't Fragment was Set)	The IP addresses of the hosts that communicate with your instance

Ping your instance

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 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 your local network

DNS server

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:

- A specific IP address or range of IP addresses in a network
- A security group ID for a group 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 file system

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 for your network.	Allows inbound SSH access from your local computer.

Elastic Load Balancing

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) For an internal load-balancer: the IPv4 CIDR block of the VPC	Allow inbound traffic on the load balancer listener port.
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*.

Controlling Access to Amazon EC2 Resources

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 \(p. 546\)](#)
- [Amazon EC2 Permission Attributes \(p. 546\)](#)
- [IAM and Amazon EC2 \(p. 546\)](#)
- [IAM Policies for Amazon EC2 \(p. 548\)](#)
- [IAM Roles for Amazon EC2 \(p. 620\)](#)
- [Authorizing Inbound Traffic for Your Linux Instances \(p. 628\)](#)

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 \(p. 628\)](#).

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 \(p. 91\)](#).

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 \(p. 802\)](#).

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**, type 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**, type 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 type 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 \(p. 548\)](#)
- [IAM Roles for Amazon EC2 \(p. 620\)](#)
- [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. 549)
Define actions in your policy	Actions for Amazon EC2 (p. 549)
Define specific resources in your policy	Amazon Resource Names for Amazon EC2 (p. 550)
Apply conditions to the use of the resources	Condition Keys for Amazon EC2 (p. 553)
Work with the available resource-level permissions for Amazon EC2	Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 557)
Test your policy	Checking That Users Have the Required Permissions (p. 556)
Example policies for a CLI or SDK	Example Policies for Working with the AWS CLI or an AWS SDK (p. 586)
Example policies for the Amazon EC2 console	Example Policies for Working in the Amazon EC2 Console (p. 612)

Policy Structure

The following topics explain the structure of an IAM policy.

Topics

- [Policy Syntax \(p. 549\)](#)
- [Actions for Amazon EC2 \(p. 549\)](#)
- [Amazon Resource Names for Amazon EC2 \(p. 550\)](#)
- [Condition Keys for Amazon EC2 \(p. 553\)](#)
- [Checking That Users Have the Required Permissions \(p. 556\)](#)

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 \(p. 549\)](#).
- **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. To specify a resource in the statement, you need to use its Amazon Resource Name (ARN). For more information about specifying the ARN value, see [Amazon Resource Names for Amazon EC2 \(p. 550\)](#). For more information about which API actions support which ARNs, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 557\)](#). If the API action does not support ARNs, use the * wildcard to specify that all resources can be affected by the action.
- **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 \(p. 553\)](#).

For more information about example IAM policy statements for Amazon EC2, see [Example Policies for Working with the AWS CLI or an AWS SDK \(p. 586\)](#).

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*.

Amazon Resource Names for Amazon EC2

Each IAM policy statement applies to the resources that you specify using their ARNs.

Important

Currently, not all API actions support individual ARNs. We'll add support for additional API actions and ARNs for additional Amazon EC2 resources later. For information about which ARNs you can use with which Amazon EC2 API actions, as well as supported condition keys for each ARN, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 557\)](#).

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 also specify all instances that belong to a specific account by using the * wildcard as follows:

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

To specify all resources, or if a specific API action does not support ARNs, use the * wildcard in the `Resource` element as follows:

```
"Resource": "*"
```

The following table describes the ARNs for each type of resource used by the Amazon EC2 API actions.

Resource Type	ARN
All Amazon EC2 resources	arn:aws:ec2:*
All Amazon EC2 resources owned by the specified account in the specified region	arn:aws:ec2: <i>region</i> :account:*
Customer gateway	arn:aws:ec2: <i>region</i> :account:customer-gateway/ <i>cgw-id</i> Where <i>cgw-id</i> is cgw-xxxxxxxx
DHCP options set	arn:aws:ec2: <i>region</i> :account:dhcp-options/ <i>dhcp-options-id</i> Where <i>dhcp-options-id</i> is dopt-xxxxxxxx
Elastic GPU	arn:aws:ec2: <i>region</i> :account:elastic-gpu/*
Image	arn:aws:ec2: <i>region</i> ::image/ <i>image-id</i> Where <i>image-id</i> is the ID of the AMI, AKI, or ARI, and <i>account</i> isn't used
Instance	arn:aws:ec2: <i>region</i> :account:instance/ <i>instance-id</i> Where <i>instance-id</i> is i-xxxxxxxx or i-xxxxxxxxxxxxxxxxxxxx
Instance profile	arn:aws:iam::account:instance-profile/ <i>instance-profile-name</i> Where <i>instance-profile-name</i> is the name of the instance profile, and <i>region</i> isn't used
Internet gateway	arn:aws:ec2: <i>region</i> :account:internet-gateway/ <i>igw-id</i> Where <i>igw-id</i> is igw-xxxxxxxx
Key pair	arn:aws:ec2: <i>region</i> :account:key-pair/ <i>key-pair-name</i> Where <i>key-pair-name</i> is the key pair name (for example, gsg-keypair)
Launch template	arn:aws:ec2: <i>region</i> :account:launch-template/ <i>launch-template-id</i> Where <i>launch-template-id</i> is lt-xxxxxxxxxxxxxxxxxx
NAT gateway	arn:aws:ec2: <i>region</i> :account:natgateway/ <i>natgateway-id</i> Where <i>natgateway-id</i> is nat-xxxxxxxxxxxxxxxxxx
Network ACL	arn:aws:ec2: <i>region</i> :account:network-acl/ <i>nacl-id</i> Where <i>nacl-id</i> is acl-xxxxxxxx
Network interface	arn:aws:ec2: <i>region</i> :account:network-interface/ <i>eni-id</i> Where <i>eni-id</i> is eni-xxxxxxxx

Resource Type	ARN
Placement group	arn:aws:ec2:region:account:placement-group/ <i>placement-group-name</i> Where <i>placement-group-name</i> is the placement group name (for example, <code>my-cluster</code>)
Reserved Instance	arn:aws:ec2:region:account:reserved-instances/ <i>reservation-id</i> Where <i>reservation-id</i> is <code>xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx</code>
Route table	arn:aws:ec2:region:account:route-table/ <i>route-table-id</i> Where <i>route-table-id</i> is <code>rtb-xxxxxxxx</code>
Security group	arn:aws:ec2:region:account:security-group/ <i>security-group-id</i> Where <i>security-group-id</i> is <code>sg-xxxxxxxx</code>
Snapshot	arn:aws:ec2:region::snapshot/ <i>snapshot-id</i> Where <i>snapshot-id</i> is <code>snap-xxxxxxxx</code> or <code>snap-xxxxxxxxxxxxxxxxxx</code> , and <i>account</i> isn't used
Spot Instance request	arn:aws:ec2:region:account:spot-instances-request/ <i>spot-instance-request-id</i> Where <i>spot-instance-request-id</i> is <code>sir-xxxxxxxx</code>
Subnet	arn:aws:ec2:region:account:subnet/ <i>subnet-id</i> Where <i>subnet-id</i> is <code>subnet-xxxxxxxx</code>
Volume	arn:aws:ec2:region:account:volume/ <i>volume-id</i> Where <i>volume-id</i> is <code>vol-xxxxxxxx</code> or <code>vol-xxxxxxxxxxxxxxxxxx</code>
VPC	arn:aws:ec2:region:account:vpc/ <i>vpc-id</i> Where <i>vpc-id</i> is <code>vpc-xxxxxxxx</code>
VPC peering connection	arn:aws:ec2:region:account:vpc-peering-connection/ <i>vpc-peering-connection-id</i> Where <i>vpc-peering connection-id</i> is <code>pcx-xxxxxxxx</code>
VPN connection	arn:aws:ec2:region:account:vpn-connection/ <i>vpn-connection-id</i> Where <i>vpn-connection-id</i> is <code>vpn-xxxxxxxx</code>
VPN gateway	arn:aws:ec2:region:account:vpn-gateway/ <i>vpn-gateway-id</i> Where <i>vpn-gateway-id</i> is <code>vgw-xxxxxxxx</code>

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 more general information about ARNs, see [Amazon Resource Names \(ARN\) and AWS Service Namespaces](#) in the *Amazon Web Services General Reference*. For more information about the resources that are created or modified by the Amazon EC2 actions, and the ARNs that you can use in your IAM policy statements, see [Granting IAM Users Required Permissions for Amazon EC2 Resources](#) in the *Amazon EC2 API Reference*.

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.

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*.

Amazon EC2 implements the following service-specific condition keys. For information about which condition keys you can use with which Amazon EC2 resources, on an action-by-action basis, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 557\)](#).

Condition Key	Key-Value Pair	Evaluation Types
<code>ec2:AccepterVpc</code>	<code>"ec2:AccepterVpc":"vpc-arn"</code> Where <code>vpc-arn</code> is the VPC ARN for the accepter VPC in a VPC peering connection	ARN, Null
<code>ec2:AuthorizedService</code>	<code>"ec2:AuthorizedService":"service-principal"</code> Where <code>service-principal</code> is the service principal (for example, <code>ecs.amazonaws.com</code>)	String, Null
<code>ec2:AuthorizedUser</code>	<code>"ec2:AuthorizedUser":"principal-arn"</code> Where <code>principal-arn</code> is the ARN for the principal (for example, <code>arn:aws:iam::123456789012:root</code>)	ARN, Null
<code>ec2:AvailabilityZone</code>	<code>"ec2:AvailabilityZone":"az-api-name"</code> Where <code>az-api-name</code> is the name of the Availability Zone (for example, <code>us-east-2a</code>) To list your Availability Zones, use describe-availability-zones	String, Null

Condition Key	Key-Value Pair	Evaluation Types
ec2:CreateAction	"ec2:CreateAction":" <i>api-name</i> " Where <i>api-name</i> is the name of the resource-creating action (for example, RunInstances)	String, Null
ec2:EbsOptimized	"ec2:EbsOptimized":" <i>optimized-flag</i> " Where <i>optimized-flag</i> is true false (for an instance)	Boolean, Null
ec2:ElasticGpuType	"ec2:ElasticGpuType":" <i>elastic-gpu-type</i> " Where <i>elastic-gpu-type</i> is the name of the elastic GPU type	String, Null
ec2:Encrypted	"ec2:Encrypted":" <i>encrypted-flag</i> " Where <i>encrypted-flag</i> is true false (for an EBS volume)	Boolean, Null
ec2:ImageType	"ec2:ImageType":" <i>image-type-api-name</i> " Where <i>image-type-api-name</i> is ami aki ari	String, Null
ec2:InstanceMarket	"ec2:InstanceMarketType":" <i>market-type</i> " Where <i>market-type</i> is spot on-demand	String, Null
ec2:InstanceProfile	"ec2:InstanceProfile":" <i>instance-profile-arn</i> " Where <i>instance-profile-arn</i> is the instance profile ARN	ARN, Null
ec2:InstanceType	"ec2:InstanceType":" <i>instance-type-api-name</i> " Where <i>instance-type-api-name</i> is the name of the instance type.	String, Null
ec2:IsLaunchTemplateResource	"ec2:LaunchTemplateResource":" <i>launch-template-resource-flag</i> " Where <i>launch-template-resource-flag</i> is true false	Boolean, Null
ec2:LaunchTemplate	"ec2:LaunchTemplate":" <i>launch-template-arn</i> " Where <i>launch-template-arn</i> is the launch template ARN	ARN, Null
ec2:Owner	"ec2:Owner":" <i>account-id</i> " Where <i>account-id</i> is amazon aws-marketplace aws-account-id	String, Null
ec2:ParentSnapshot	"ec2:ParentSnapshot":" <i>snapshot-arn</i> " Where <i>snapshot-arn</i> is the snapshot ARN	ARN, Null
ec2:ParentVolume	"ec2:ParentVolume":" <i>volume-arn</i> " Where <i>volume-arn</i> is the volume ARN	ARN, Null
ec2:Permission	"ec2:Permission":" <i>permission</i> " Where <i>permission</i> is INSTANCE-ATTACH EIP-ASSOCIATE	String, Null

Condition Key	Key-Value Pair	Evaluation Types
ec2:PlacementGroup	"ec2:PlacementGroup":"placement-group-arn" Where <i>placement-group-arn</i> is the placement group ARN	ARN, Null
ec2:PlacementGroupStrategy	"ec2:PlacementGroupStrategy":"placement-group-strategy" Where <i>placement-group-strategy</i> is cluster spread	String, Null
ec2:ProductCode	"ec2:ProductCode":"product-code" Where <i>product-code</i> is the product code	String, Null
ec2:Public	"ec2:Public":"public-flag" Where <i>public-flag</i> is true false (for an AMI)	Boolean, Null
ec2:Region	"ec2:Region":"region-name" Where <i>region-name</i> is the name of the region (for example, us-east-2). To list your regions, use describe-regions . This condition key can be used with all Amazon EC2 actions.	String, Null
ec2:RequesterVpc	"ec2:RequesterVpc":"vpc-arn" Where <i>vpc-arn</i> is the VPC ARN for the requester VPC in a VPC peering connection	ARN, Null
ec2:ReservedInstancesOfferingType	"ec2:ReservedInstancesOfferingType":"offering-type" Where <i>offering-type</i> is No Upfront Partial Upfront All Upfront	String, Null
ec2:ResourceTag/ <i>tag-key</i>	"ec2:ResourceTag/tag-key":"tag-value" Where <i>tag-key</i> and <i>tag-value</i> are the tag-key pair	String, Null
ec2:RootDeviceType	"ec2:RootDeviceType":"root-device-type-name" Where <i>root-device-type-name</i> is ebs instance-store	String, Null
ec2:SnapshotTime	"ec2:SnapshotTime":"time" Where <i>time</i> is the snapshot creation time (for example, 2013-06-01T00:00:00Z)	Date, Null
ec2:Subnet	"ec2:Subnet":"subnet-arn" Where <i>subnet-arn</i> is the subnet ARN	ARN, Null
ec2:Tenancy	"ec2:Tenancy":"tenancy-attribute" Where <i>tenancy-attribute</i> is default dedicated host	String, Null
ec2:VolumeIops	"ec2:VolumeIops":"volume-iops" Where <i>volume-iops</i> is the input/output operations per second (IOPS). For more information, see Amazon EBS Volume Types (p. 735) .	Numeric, Null

Condition Key	Key-Value Pair	Evaluation Types
ec2:VolumeSize	"ec2:VolumeSize":" <i>volume-size</i> " Where <i>volume-size</i> is the size of the volume, in GiB	Numeric, Null
ec2:VolumeType	"ec2:VolumeType":" <i>volume-type-name</i> " Where <i>volume-type-name</i> is gp2 for General Purpose SSD volumes, io1 for Provisioned IOPS SSD volumes, st1 for Throughput Optimized HDD volumes, sc1 for Cold HDD volumes, or standard for Magnetic volumes.	String, Null
ec2:Vpc	"ec2:Vpc":" <i>vpc-arn</i> " Where <i>vpc-arn</i> is the VPC ARN	ARN, Null

Amazon EC2 also implements the AWS-wide condition keys. For more information, see [Information Available in All Requests](#) in the *IAM User Guide*.

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 [12: Allowing a Specific Instance to View Resources in Other AWS Services \(p. 611\)](#). The `ec2:SourceInstanceARN` key cannot be used as a variable to populate the ARN for the `Resource` element in a statement.

The following AWS condition keys were introduced for Amazon EC2 and are supported by a limited number of additional services.

Condition Key	Key/Value Pair	Evaluation Types
aws:RequestTag/ <i>tag-key</i>	"aws:Request/tag-key":" <i>tag-value</i> " Where <i>tag-key</i> and <i>tag-value</i> are the tag key-value pair	String, Null
aws:TagKeys	"aws:TagKeys":" <i>tag-key</i> " Where <i>tag-key</i> is a list of tag keys (for example, ["A","B"])	String, Null

For example policy statements for Amazon EC2, see [Example Policies for Working with the AWS CLI or an AWS SDK \(p. 586\)](#).

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*.

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.

The following table describes the Amazon EC2 API actions that currently support resource-level permissions, as well as the supported resources (and their ARNs) and condition keys for each action. When specifying an ARN, you can use the * wildcard in your paths; for example, when you cannot or do not want to specify exact resource IDs. For examples of using wildcards, see [Example Policies for Working with the AWS CLI or an AWS SDK \(p. 586\)](#).

Important

If an Amazon EC2 API action is not listed in this table, then it does not support resource-level permissions. If an Amazon EC2 API action does not support resource-level permissions, you can grant users permissions to use the action, but you have to specify a * for the resource element of your policy statement. For an example, see [1: Read-Only Access \(p. 586\)](#). For a list of Amazon EC2 API actions that currently do not support resource-level permissions, see [Unsupported Resource-Level Permissions in the Amazon EC2 API Reference](#).

All Amazon EC2 actions support the `ec2:Region` condition key. For an example, see [2: Restricting Access to a Specific Region \(p. 587\)](#).

API Action	Resources	Condition Keys
AcceptVpcPeeringConnection	VPC peering connection arn:aws:ec2:region:account:vpc-peering-connection/* arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id	ec2:AcceptorVpc ec2:Region ec2:ResourceTag/tag-key ec2:RequesterVpc
	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id Where <i>vpc-id</i> is a VPC owned by the accepter.	ec2:ResourceTag/tag-key ec2:Region ec2:Tenancy

API Action	Resources	Condition Keys
AssociateIamInstanceProfile	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
AttachClassicLinkVpc	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
	Security group arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/ <i>security-group-id</i> Where the security group is the security group for the VPC.	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/ <i>vpc-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Tenancy

API Action	Resources	Condition Keys
AttachVolume	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
	Volume arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/ <i>volume-id</i>	ec2:AvailabilityZone ec2:Encrypted ec2:ParentSnapshot ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:VolumeLops ec2:VolumeSize ec2:VolumeType
AuthorizeSecurityGroupEgress	Security group arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/ <i>security-group-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
AuthorizeSecurityGroupIngress	Security group arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/ <i>security-group-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
CreateLaunchTemplateVersion	Launch template arn:aws:ec2:region:account:launch-template/* arn:aws:ec2:region:account:launch-template/ <i>launch-template-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i>

API Action	Resources	Condition Keys
CreateNetworkInterfacePermission	Network interface arn:aws:ec2:region:account:network-interface/* arn:aws:ec2:region:account:network-interface/ <i>eni-id</i>	ec2:AuthorizedUser ec2:AvailabilityZone ec2:Permission ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Subnet ec2:Vpc
CreateRoute	Route table arn:aws:ec2:region:account:route-table/* arn:aws:ec2:region:account:route-table/ <i>route-table-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
CreateSnapshot	Snapshot arn:aws:ec2:region::snapshot/*	ec2:ParentVolume ec2:Region aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Volume arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/ <i>volume-id</i>	ec2:Encrypted ec2:Region ec2:VolumeLops ec2:VolumeSize ec2:VolumeType ec2:ResourceTag/ <i>tag-key</i>
CreateTags	Amazon FPGA image (AFI) arn:aws:ec2:region:account:fpga-image/* arn:aws:ec2:region:account:fpga-image/ <i>afi-id</i>	ec2:CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	DHCP options set arn:aws:ec2:region:account:dhcp-options/* arn:aws:ec2:region:account:dhcp-options/ <i>dhcp-options-id</i>	ec2:CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys

API Action	Resources	Condition Keys
	Image arn:aws:ec2:region::image/* arn:aws:ec2:region::image/ <i>image-id</i>	ec2:CreateAction ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType
		aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:AvailabilityZone ec2:CreateAction ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
		aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Internet gateway arn:aws:ec2:region:account:internet-gateway/* arn:aws:ec2:region:account:internet-gateway/ <i>igw-id</i>	ec2:CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i>
		aws:RequestTag/ <i>tag-key</i> aws:TagKeys

API Action	Resources	Condition Keys
	Launch template arn:aws:ec2:region:account:launch-template/* arn:aws:ec2:region:account:launch-template/ <i>launch-template-id</i>	ec2:CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	NAT gateway arn:aws:ec2:region:account:natgateway/* arn:aws:ec2:region:account:natgateway/ <i>natgateway-id</i>	ec2:CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Network ACL arn:aws:ec2:region:account:network-acl/* arn:aws:ec2:region:account:network-acl/ <i>nacl-id</i>	ec2:CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Network interface arn:aws:ec2:region:account:network-interface/* arn:aws:ec2:region:account:network-interface/ <i>eni-id</i>	ec2:AvailabilityZone ec2:CreateAction ec2:Region ec2:Subnet ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc aws:RequestTag/ <i>tag-key</i> aws:TagKeys

API Action	Resources	Condition Keys
	Reserved Instance <code>arn:aws:ec2:region:account:reserved-instances/*</code> <code>arn:aws:ec2:region:account:reserved-instances/<i>reservation-id</i></code>	ec2:AvailabilityZone ec2>CreateAction ec2:InstanceType ec2:ReservedInstancesOfferingType ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Tenancy
		aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Route table <code>arn:aws:ec2:region:account:route-table/*</code> <code>arn:aws:ec2:region:account:route-table/<i>route-table-id</i></code>	ec2>CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
		aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Security group <code>arn:aws:ec2:region:account:security-group/*</code> <code>arn:aws:ec2:region:account:security-group/<i>security-group-id</i></code>	ec2>CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
		aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Snapshot <code>arn:aws:ec2:region::snapshot/*</code> <code>arn:aws:ec2:region::snapshot/<i>snapshot-id</i></code>	ec2>CreateAction ec2:Owner ec2:ParentVolume ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:SnapshotTime ec2:VolumeSize
		aws:RequestTag/ <i>tag-key</i> aws:TagKeys

API Action	Resources	Condition Keys
	Spot Instance request arn:aws:ec2:region:account:spot-instances-request/* arn:aws:ec2:region:account:spot-instances-request/ <i>spot-instance-request-id</i>	ec2:CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Subnet arn:aws:ec2:region:account:subnet/* arn:aws:ec2:region:account:subnet/ <i>subnet-id</i>	ec2:AvailabilityZone ec2:CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Volume arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/ <i>volume-id</i>	ec2:AvailabilityZone ec2:CreateAction ec2:Encrypted ec2:ParentSnapshot ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Volumelops ec2:VolumeSize ec2:VolumeType aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc- <i>id</i>	ec2:CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Tenancy aws:RequestTag/ <i>tag-key</i> aws:TagKeys

API Action	Resources	Condition Keys
	VPN connection arn:aws:ec2:region:account:vpn-connection/* arn:aws:ec2:region:account:vpn-connection/vpn-connection-id	ec2:CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	VPN gateway arn:aws:ec2:region:account:vpn-gateway/* arn:aws:ec2:region:account:vpn-gateway/vpn-gateway-id	ec2:CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
CreateVolume	Volume arn:aws:ec2:region:account:volume/*	ec2:AvailabilityZone ec2:Encrypted ec2:ParentSnapshot ec2:Region ec2:VolumeLops ec2:VolumeSize ec2:VolumeType aws:RequestTag/ <i>tag-key</i> aws:TagKeys
CreateVpcPeeringConnection	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id Where <i>vpc-id</i> is a requester VPC.	ec2:ResourceTag/ <i>tag-key</i> ec2:Region ec2:Tenancy
	VPC peering connection arn:aws:ec2:region:account:vpc-peering-connection/*	ec2:AcceptorVpc ec2:Region ec2:RequesterVpc
DeleteCustomerGateway	Customer gateway arn:aws:ec2:region:account:customer-gateway/* arn:aws:ec2:region:account:customer-gateway/cgw-id	ec2:Region ec2:ResourceTag/ <i>tag-key</i>

API Action	Resources	Condition Keys
DeleteDhcpOptions	DHCP options set arn:aws:ec2:region:account:dhcp-options/* arn:aws:ec2:region:account:dhcp-options/dhcp-options-id	ec2:Region ec2:ResourceTag/ <i>tag-key</i>
DeleteInternetGateway	Internet gateway arn:aws:ec2:region:account:internet-gateway/* arn:aws:ec2:region:account:internet-gateway/igw-id	ec2:Region ec2:ResourceTag/ <i>tag-key</i>
DeleteLaunchTemplate	Launch template arn:aws:ec2:region:account:launch-template/* arn:aws:ec2:region:account:launch-template/launch-template-id	ec2:Region ec2:ResourceTag/ <i>tag-key</i>
DeleteLaunchTemplateVersion	Launch template arn:aws:ec2:region:account:launch-template/* arn:aws:ec2:region:account:launch-template/launch-template-id	ec2:Region ec2:ResourceTag/ <i>tag-key</i>
DeleteNetworkAcl	Network ACL arn:aws:ec2:region:account:network-acl/* arn:aws:ec2:region:account:network-acl-nacl-id	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
DeleteNetworkAclEntry	Network ACL arn:aws:ec2:region:account:network-acl/* arn:aws:ec2:region:account:network-acl-nacl-id	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
DeleteRoute	Route table arn:aws:ec2:region:account:route-table/* arn:aws:ec2:region:account:route-table/route-table-id	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
DeleteRouteTable	Route table arn:aws:ec2:region:account:route-table/* arn:aws:ec2:region:account:route-table/route-table-id	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc

API Action	Resources	Condition Keys
DeleteSecurityGroup	Security group arn:aws:ec2:region:account:security-group/ <i>security-group-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
DeleteSnapshot	Snapshot arn:aws:ec2:region::snapshot/* arn:aws:ec2:region::snapshot/ <i>snapshot-id</i>	ec2:Owner ec2:ParentVolume ec2:Region ec2:SnapshotTime ec2:VolumeSize ec2:ResourceTag/ <i>tag-key</i>
DeleteTags	Amazon FPGA image (AFI) arn:aws:ec2:region:account:fpga-image/* arn:aws:ec2:region:account:fpga-image/ <i>afi-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	DHCP options set arn:aws:ec2:region:account:dhcp-options/* arn:aws:ec2:region:account:dhcp-options/ <i>dhcp-options-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Image arn:aws:ec2:region::image/* arn:aws:ec2:region::image/ <i>image-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Internet gateway arn:aws:ec2:region:account:internet-gateway/* arn:aws:ec2:region:account:internet-gateway/ <i>igw-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys

API Action	Resources	Condition Keys
	Launch template arn:aws:ec2:region:account:launch-template/* arn:aws:ec2:region:account:launch-template/ <i>launch-template-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	NAT gateway arn:aws:ec2:region:account:natgateway/* arn:aws:ec2:region:account:natgateway/ <i>natgateway-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Network ACL arn:aws:ec2:region:account:network-acl/* arn:aws:ec2:region:account:network-acl/ <i>nacl-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Network interface arn:aws:ec2:region:account:network-interface/* arn:aws:ec2:region:account:network-interface/ <i>eni-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Reserved Instance arn:aws:ec2:region:account:reserved-instances/* arn:aws:ec2:region:account:reserved-instances/ <i>reservation-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Route table arn:aws:ec2:region:account:route-table/* arn:aws:ec2:region:account:route-table/ <i>route-table-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Security group arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/ <i>security-group-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys

API Action	Resources	Condition Keys
	Snapshot arn:aws:ec2:region::snapshot/* arn:aws:ec2:region::snapshot/ <i>snapshot-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Spot Instance request arn:aws:ec2:region:account:spot-instances-request/* arn:aws:ec2:region:account:spot-instances-request/ <i>spot-instance-request-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Subnet arn:aws:ec2:region:account:subnet/* arn:aws:ec2:region:account:subnet/ <i>subnet-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Volume arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/ <i>volume-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/ <i>vpc-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	VPN connection arn:aws:ec2:region:account:vpn-connection/* arn:aws:ec2:region:account:vpn-connection/ <i>vpn-connection-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	VPN gateway arn:aws:ec2:region:account:vpn-gateway/* arn:aws:ec2:region:account:vpn-gateway/ <i>vpn-gateway-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> aws:RequestTag/ <i>tag-key</i> aws:TagKeys

API Action	Resources	Condition Keys
DeleteVolume	Volume arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/ <i>volume-id</i>	ec2:AvailabilityZone ec2:Encrypted ec2:ParentSnapshot ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Volumelops ec2:VolumeSize ec2:VolumeType
DeleteVpcPeeringConnection	VPC peering connection arn:aws:ec2:region:account:vpc-peering-connection/* arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection- <i>id</i>	ec2:AcceptorVpc ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RequesterVpc
DetachClassicLinkVpc	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc- <i>id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Tenancy

API Action	Resources	Condition Keys
DetachVolume	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
	Volume arn:aws:ec2:region:account:volume/* arn:aws:ec2:region:account:volume/ <i>volume-id</i>	ec2:AvailabilityZone ec2:Encrypted ec2:ParentSnapshot ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:VolumeLops ec2:VolumeSize ec2:VolumeType
DisableVpcClassicLink	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc- <i>id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Tenancy
DisassociateIamInstanceProfile	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy

API Action	Resources	Condition Keys
EnableVpcClassicLink	VPC arn:aws:ec2:region:account:vpc/* arn:aws:ec2:region:account:vpc/vpc-id	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Tenancy
GetConsoleScreenshot	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
ModifyLaunchTemplate	Launch template arn:aws:ec2:region:account:launch-template/* arn:aws:ec2:region:account:launch-template/ <i>launch-template-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i>
ModifySnapshotAttribute	Snapshot arn:aws:ec2:region::snapshot/* arn:aws:ec2:region::snapshot/ <i>snapshot-id</i>	ec2:Owner ec2:ParentVolume ec2:Region ec2:SnapshotTime ec2:VolumeSize ec2:ResourceTag/ <i>tag-key</i>

API Action	Resources	Condition Keys
RebootInstances	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
RejectVpcPeeringConnection	VPC peering connection arn:aws:ec2:region:account:vpc-peering-connection/* arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection- <i>id</i>	ec2:AcceptorVpc ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RequesterVpc
ReplaceIamInstanceProfile	Assumption arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
ReplaceRoute	Route table arn:aws:ec2:region:account:route-table/* arn:aws:ec2:region:account:route-table/route-table- <i>id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
RevokeSecurityGroupEgress	Security group arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/security-group- <i>id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc

API Action	Resources	Condition Keys
RevokeSecurityGroupIngress	Security group arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/ <i>security-group-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
RunInstances	Elastic GPU arn:aws:ec2:region:account:elastic-gpu/*	ec2:ElasticGpuType ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Region
	Image arn:aws:ec2:region::image/* arn:aws:ec2:region::image/ <i>image-id</i>	ec2:ImageType ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/ <i>tag-key</i>
	Instance arn:aws:ec2:region:account:instance/*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceMarketType ec2:InstanceProfile ec2:InstanceType ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy aws:RequestTag/ <i>tag-key</i> aws:TagKeys

API Action	Resources	Condition Keys
	Key pair arn:aws:ec2:region:account:key-pair/* arn:aws:ec2:region:account:key-pair/ <i>key-pair-name</i>	ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Region
	Launch template arn:aws:ec2:region:account:launch-template/* arn:aws:ec2:region:account:launch-template/ <i>launch-template-id</i>	ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Region
	Network interface arn:aws:ec2:region:account:network-interface/* arn:aws:ec2:region:account:network-interface/ <i>eni-id</i>	ec2:AvailabilityZone ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Region ec2:Subnet ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
	Placement group arn:aws:ec2:region:account:placement-group/* arn:aws:ec2:region:account:placement-group/ <i>placement-group-name</i>	ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Region ec2:PlacementGroupStrategy
	Security group arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/ <i>security-group-id</i>	ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc

API Action	Resources	Condition Keys
	Snapshot <code>arn:aws:ec2:<region>::snapshot/*</code> <code>arn:aws:ec2:<region>::snapshot/<snapshot-id></code>	<code>ec2:IsLaunchTemplateResource</code> <code>ec2:LaunchTemplate</code> <code>ec2:Owner</code> <code>ec2:ParentVolume</code> <code>ec2:Region</code> <code>ec2:SnapshotTime</code> <code>ec2:ResourceTag/<i>tag-key</i></code> <code>ec2:VolumeSize</code>
	Subnet <code>arn:aws:ec2:<region>:<account>:subnet/*</code> <code>arn:aws:ec2:<region>:<account>:subnet/<subnet-id></code>	<code>ec2:AvailabilityZone</code> <code>ec2:IsLaunchTemplateResource</code> <code>ec2:LaunchTemplate</code> <code>ec2:Region</code> <code>ec2:ResourceTag/<i>tag-key</i></code> <code>ec2:Vpc</code>
	Volume <code>arn:aws:ec2:<region>:<account>:volume/*</code>	<code>ec2:AvailabilityZone</code> <code>ec2:Encrypted</code> <code>ec2:IsLaunchTemplateResource</code> <code>ec2:LaunchTemplate</code> <code>ec2:ParentSnapshot</code> <code>ec2:Region</code> <code>ec2:Volumelops</code> <code>ec2:VolumeSize</code> <code>ec2:VolumeType</code> <code>aws:RequestTag/<i>tag-key</i></code> <code>aws:TagKeys</code>

API Action	Resources	Condition Keys
StartInstances	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
StopInstances	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
TerminateInstances	Instance arn:aws:ec2:region:account:instance/* arn:aws:ec2:region:account:instance/ <i>instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy

API Action	Resources	Condition Keys
UpdateSecurityGroupRule	arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/ <i>security-group-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
UpdateSecurityGroupRule	arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/ <i>security-group-id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc

Resource-Level Permissions for RunInstances

The [RunInstances](#) API action launches one or more instances, and creates and uses a number of Amazon EC2 resources. The action requires an AMI and creates an instance; and the instance must be associated with a security group. Launching into a VPC requires a subnet, and creates a network interface. Launching from an Amazon EBS-backed AMI creates a volume. The user must have permissions to use these resources, so they must be specified in the `Resource` element of any policy that uses resource-level permissions for the `ec2:RunInstances` action. If you don't intend to use resource-level permissions with the `ec2:RunInstances` action, you can specify the `*` wildcard in the `Resource` element of your statement instead of individual ARNs.

If you are using resource-level permissions, the following table describes the minimum resources required to use the `ec2:RunInstances` action.

Type of launch	Resources required	Condition keys
Launching into EC2-Classic using an instance store-backed AMI	arn:aws:ec2:region:account:instance/*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceMarketType ec2:InstanceProfile ec2:InstanceType ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	arn:aws:ec2:region::image/* (or a specific AMI ID)	ec2:ImageType ec2:IsLaunchTemplateResource

Type of launch	Resources required	Condition keys
		ec2:LaunchTemplate ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/ <i>tag-key</i>
	arn:aws:ec2: <i>region</i> :account:securitygroup/* (or a specific security group ID)	ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
Launching into EC2-Classic using an Amazon EBS-backed AMI	arn:aws:ec2: <i>region</i> :account:instance/*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceMarketType ec2:InstanceProfile ec2:InstanceType ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	arn:aws:ec2: <i>region</i> ::image/* (or a specific AMI ID)	ec2:ImageType ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/ <i>tag-key</i>

Type of launch	Resources required	Condition keys
	arn:aws:ec2:region:account:securitygroup/* (or a specific security group ID)	ec2:LaunchTemplate ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
	arn:aws:ec2:region:account:volume/*	ec2:AvailabilityZone ec2:Encrypted ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:ParentSnapshot ec2:Region ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
Launching into a VPC using an instance store-backed AMI	arn:aws:ec2:region:account:instance/*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceMarketType ec2:InstanceProfile ec2:InstanceType ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy

Type of launch	Resources required	Condition keys
	arn:aws:ec2:region::image/* (or a specific AMI ID)	ec2:ImageType ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/ <i>tag-key</i>
	arn:aws:ec2:region:account:securitygroup/* (or a specific security group ID)	ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
	arn:aws:ec2:region:account:networkinterface/* (or a specific network interface ID)	ec2:AvailabilityZone ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Region ec2:Subnet ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
	arn:aws:ec2:region:account:subnet/* (or a specific subnet ID)	ec2:AvailabilityZone ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc

Type of launch	Resources required	Condition keys
Launching into a VPC using an Amazon EBS-backed AMI	arn:aws:ec2:region:account:instance/*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceMarketType ec2:InstanceProfile ec2:InstanceType ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	arn:aws:ec2:region::image/* (or a specific AMI ID)	ec2:ImageType ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/ <i>tag-key</i>
	arn:aws:ec2:region:account:securitygroup/* (or a specific security group ID)	ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc

Type of launch	Resources required	Condition keys
	arn:aws:ec2:region:account:networkinterface/* (or a specific network interface ID)	ec2:AvailabilityZone ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Region ec2:Subnet ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
	arn:aws:ec2:region:account:volume/*	ec2:AvailabilityZone ec2:Encrypted ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:ParentSnapshot ec2:Region ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
	arn:aws:ec2:region:account:subnet/* (or a specific subnet ID)	ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc

We recommend that you also specify the key pair resource in your policy — even though it's not required to launch an instance, you can't connect to your instance without a key pair. For examples of using resource-level permissions with the `ec2:RunInstances` action, see [6: Launching Instances \(RunInstances\) \(p. 594\)](#).

For additional information about resource-level permissions in Amazon EC2, see the following AWS Security Blog post: [Demystifying EC2 Resource-Level Permissions](#).

Resource-Level Permissions for RunInstances and Launch Templates

You can create a [launch template \(p. 361\)](#) that contains the parameters to launch an instance. When users use the `ec2:RunInstances` action, they can specify the launch template to use to launch instances. You can apply resource-level permissions for the launch template resource for the `ec2:RunInstances` action. For example, you can specify that users can only launch instances using a

launch template, and that they must use a specific launch template. You can also control the parameters that users can or cannot override in the launch template. This enables you to manage the parameters for launching an instance in a launch template rather than an IAM policy. For example policies, see [Launch Templates \(p. 603\)](#).

Resource-Level Permissions for Tagging

Some resource-creating Amazon EC2 API actions enable you to specify tags when you create the resource. For more information, see [Tagging Your Resources \(p. 889\)](#).

To enable users to tag resources on creation, they must have permissions to use the action that creates the resource (for example, `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.

For the `ec2:CreateTags` action, you can use the `ec2:CreateAction` condition key to restrict tagging permissions to the resource-creating actions only. For example, the following policy 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"  
                }  
            }  
        }  
    ]  
}
```

```
        "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 and the launch template is specified in the `ec2:RunInstances` action. For an example policy, see [Applying Tags in a Launch Template \(p. 602\)](#).

You can control the tag keys and values that are applied to resources by using the following condition keys:

- `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 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 more information about multi-value conditions, see [Creating a Condition That Tests Multiple Key Values](#) in the *IAM User Guide*. For example IAM policies, see [Example Policies for Working with the AWS CLI or an AWS SDK \(p. 586\)](#).

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 \(p. 612\)](#). For examples of IAM policies specific to Amazon VPC, see [Controlling Access to Amazon VPC Resources](#).

Contents

- [1: Read-Only Access \(p. 586\)](#)
- [2: Restricting Access to a Specific Region \(p. 587\)](#)
- [3: Working with Instances \(p. 587\)](#)
- [4. Working with Volumes \(p. 588\)](#)
- [5. Working with Snapshots \(p. 591\)](#)
- [6: Launching Instances \(RunInstances\) \(p. 594\)](#)
- [7. Working with ClassicLink \(p. 605\)](#)
- [8. Working with Reserved Instances \(p. 607\)](#)
- [9. Tagging Resources \(p. 608\)](#)
- [10: Working with IAM Roles \(p. 610\)](#)
- [11: Working with Route Tables \(p. 611\)](#)
- [12: Allowing a Specific Instance to View Resources in Other AWS Services \(p. 611\)](#)
- [13. Working with Launch Templates \(p. 612\)](#)

1: 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 [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 557\)](#).

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": "*"  
     ]  
}
```

2: Restricting Access to a Specific Region

The following policy grants users permissions to use all Amazon EC2 API actions in the EU (Frankfurt) region only. Users cannot view, create, modify, or delete resources in any other region.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:*",  
            "Resource": "*",  
            "Condition": {"StringEquals": {"aws:RequestedRegion": [  
                "eu-central-1"  
            ]}}  
        }  
    ]  
}
```

3: Working with Instances

Topics

- [Describe, Launch, Stop, Start, and Terminate All Instances \(p. 587\)](#)
- [Describe All Instances, and Stop, Start, and Terminate Only Particular Instances \(p. 587\)](#)

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 [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 557\)](#).

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": "*"  
        }  
    ]  
}
```

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 [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 557\)](#).

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": {  
                    "ec2:ResourceTag/purpose": "test"  
                }  
            }  
        }  
    ]  
}
```

4. Working with Volumes

Topics

- [Attaching and Detaching Volumes \(p. 588\)](#)
- [Creating a Volume \(p. 589\)](#)
- [Creating a Volume with Tags \(p. 590\)](#)

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}"  
                }  
            }  
        }  
    ]  
}
```

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"  
                }  
            }  
        }  
    ]  
}
```

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 [Resource-Level Permissions for Tagging \(p. 584\)](#).

```
{  
    "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"  
            ],  
            "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"
    }
}
]
```

5. Working with Snapshots

Topics

- [Creating a Snapshot \(p. 591\)](#)
- [Creating a Snapshot with Tags \(p. 591\)](#)
- [Modifying Permission Settings for Snapshots \(p. 594\)](#)

Creating a Snapshot

The following policy allows customers to use the [CreateSnapshot](#) API action. The customer may create a snapshot 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"
                }
            }
        }
    ]
}
```

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` may 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 [Resource-Level Permissions for Tagging](#).

```
{  
    "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"  
                }  
            }  
        }  
    ]  
}
```

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` 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:RequestTag": "  
                }  
            }  
        }  
    ]  
}
```

```
        "aws:TagKeys":"purpose"
    }
}
]
```

The following policy allows a snapshot 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 may 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 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"  
                }  
            }  
        }  
    ]  
}
```

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}"  
                }  
            }  
        }  
    ]  
}
```

6: Launching Instances (RunInstances)

The `RunInstances` API action launches one or more instances. `RunInstances` requires an AMI and creates an instance; and users can specify a key pair and security group in the request. Launching into EC2-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 [Resource-Level Permissions for RunInstances \(p. 578\)](#).

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 [3: Working with Instances \(p. 587\)](#).

Topics

- [AMI \(p. 595\)](#)
- [Instance Type \(p. 596\)](#)
- [Subnet \(p. 597\)](#)
- [EBS Volumes \(p. 598\)](#)
- [Applying Tags \(p. 599\)](#)
- [Applying Tags in a Launch Template \(p. 602\)](#)
- [Attaching an Elastic GPU \(p. 602\)](#)
- [Launch Templates \(p. 603\)](#)

AMI

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/*"
    ]
}
]
}

```

[EC2-Classic only] The following policy allows users to launch instances using only the AMIs that have the specified tag, "department=dev", associated with them. The users can't launch instances using other AMIs because the Condition element of the first statement requires that users specify an AMI that has this tag. Users can only launch into EC2-Classic, as the policy does not grant permissions for the subnet and network interface resources. The second statement uses a wildcard to enable users to create instance resources, and requires users to specify the key pair `project_keypair` and the security group `sg-1a2b3c4d`. Users are still able to launch instances without a key pair.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region::image/ami-*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/department": "dev"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region:account:instance/*",
                "arn:aws:ec2:region:account:volume/*",
                "arn:aws:ec2:region:account:key-pair/project_keypair",
                "arn:aws:ec2:region:account:security-group/sg-1a2b3c4d"
            ]
        }
    ]
}

```

Instance Type

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/*"
            ]
        }
    ]
}
```

Subnet

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). Users are still able to launch instances into EC2-Classic.

```
{
    "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:account: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. Users are still able to launch instances into EC2-Classic.

```
{
    "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/*"
            ]
        }
    ]
}
```

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/*"
        ]
    }
]
```

Applying Tags

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 [Resource-Level Permissions for Tagging \(p. 584\)](#).

```
{
    "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"
                }
            }
        }
    ]
}
```

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"
                }
            }
        }
    ]
}
```

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"
        }
    }
}
]
```

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": {
                    "ec2:CreateAction" : "RunInstances"
                }
            }
        }
    ]
}
```

```
        "StringEquals": {
            "aws:RequestTag/purpose": "test",
            "ec2:CreateAction" : "RunInstances"
        },
        "ForAllValues:StringEquals": {
            "aws:TagKeys": "purpose"
        }
    }
}
```

Applying 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"
                }
            }
        }
    ]
}
```

Attaching an Elastic GPU

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 of the launch template resources in the `RunInstances` request.

```
{
    "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"
            }
        }
    }
]
```

7. Working with 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.

Topics

- [Full Permissions to Work with ClassicLink \(p. 605\)](#)
- [Enable and Disable a VPC for ClassicLink \(p. 606\)](#)
- [Link Instances \(p. 606\)](#)
- [Unlink Instances \(p. 607\)](#)

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/*"
            ]
        }
    ]
}
```

8. 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": "*"
        }
    ]
}
```

9. 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 `environment` 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/environment": ["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",
                    "ec2:ResourceTag/owner": "${aws:username}"
                },
                "ForAllValues:StringEquals": {

```

```
        "aws:TagKeys": [ "environment" ]
    }
}
]
```

10: 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 \(p. 621\)](#).

```
{
    "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-*"
        }
    ]
}
```

11: 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"
                }
            }
        }
    ]
}
```

12: 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",
                "lambda:InvokeFunction"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:SourceInstanceARN": "arn:aws:ec2:region:account:instance/i-093452212644b0dd6"
                }
            }
        }
    ]
}
```

```
        "rds:DescribeDBInstances"
    ],
    "Resource": [
        "*"
    ],
    "Condition": {
        "ArnEquals": {
            "ec2:SourceInstanceARN": "arn:aws:ec2:region:account:instance/i-093452212644b0dd6"
        }
    }
}
```

13. 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"
                }
            }
        }
    ]
}
```

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.

Topics

- [1: Read-Only Access \(p. 613\)](#)
- [2: Using the EC2 Launch Wizard \(p. 614\)](#)
- [3: Working with Volumes \(p. 616\)](#)
- [4: Working with Security Groups \(p. 617\)](#)
- [5: Working with Elastic IP Addresses \(p. 619\)](#)
- [6: Working with Reserved Instances \(p. 619\)](#)

Note

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.

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](#).

1: 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: [1: Read-Only Access \(p. 586\)](#). Users cannot perform any actions on those resources or create new resources, unless another statement grants them permission to do so.

a. 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

Currently, 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 [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 557\)](#).

b. 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": "*"  
    }]  
}
```

2: 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.

a. 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, which are EC2-Classic and a list of VPCs. This is required even if you are not launching into a VPC.
- `ec2:DescribeSubnets`: If launching into a VPC, to view all available subnets for the chosen VPC.
- `ec2:DescribeSecurityGroups`: To view the security groups page in the wizard. Users can select an existing security group.
- `ec2:DescribeKeyPairs` or `ec2>CreateKeyPair`: To select an existing key pair, or create a new one.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": [  
            "ec2:DescribeInstances", "ec2:DescribeImages",  
            "ec2:DescribeKeyPairs", "ec2:DescribeVpcs", "ec2:DescribeSubnets",  
            "ec2:DescribeSecurityGroups"  
        ],  
        "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**: If launching into EC2-Classic, to view and select a specific Availability Zone.
- **ec2:DescribeNetworkInterfaces**: If launching into a VPC, to view and select existing network interfaces for the selected subnet.
- **ec2:CreateSecurityGroup**: To create a new security group; for example, to create the wizard's suggested launch-wizard-x security group. However, this action alone only creates the security group; it does not add or modify any rules. To add inbound rules, users must be granted permission to use the **ec2:AuthorizeSecurityGroupIngress** API action. 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 [Resource-Level Permissions for Tagging \(p. 584\)](#). 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. This limits your ability to use the **ec2:ResourceTag** condition key to restrict the use of other resources; users can change a resource's tag in order to bypass those restrictions.

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.

b. Restrict access to specific instance type, subnet, and region

The following policy allows users to launch **m1.small** 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, as demonstrated 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 [6: Launching Instances \(RunInstances\) \(p. 594\)](#). The third and fourth statements grant users permission to use the instance and AMI resources respectively, but only if the instance is an **m1.small** 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:DescribeVpcs", "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups"
            ],

```

```

    "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": "m1.small"
        }
    }
},
{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
        "arn:aws:ec2:sa-east-1::image/ami-*"
    ],
    "Condition": {
        "StringEquals": {
            "ec2:Owner": "amazon"
        }
    }
}
]
}

```

3: 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/*"
}
]
```

4: Working with Security Groups

a. View security groups and add and remove rules

The following policy grants users permission to view security groups in the Amazon EC2 console, and to add and remove inbound and outbound rules for existing security groups that have the tag Department=Test.

Note

You can't modify outbound rules for EC2-Classic security groups. For more information about security groups, see [Amazon EC2 Security Groups for Linux Instances \(p. 531\)](#).

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"
            ],
            "Resource": [
                "arn:aws:ec2:region:111122223333:security-group/*"
            ],
            "Condition": {
                "StringEquals": {

```

```
        "ec2:ResourceTag/Department": "Test"
    }
}
]
```

b. 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. This action is not required for creating security groups in EC2-Classic.

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.

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": [
                "arn:aws:ec2:vpc-1a2b3c4d/*"
            ]
        }
    ]
}
```

```
        "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"
        }
    }
}
```

5: 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 address for use in VPC or EC2-Classic.
- `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. For an EC2-Classic instance, users only need permission to use `ec2:DescribeInstances`.
- `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": "*"
        }
    ]
}
```

6: 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: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 a running or stopped 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.

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. 557) and the following example: [10: Working with IAM Roles \(p. 610\)](#).

Topics

- [Instance Profiles \(p. 621\)](#)
- [Retrieving Security Credentials from Instance Metadata \(p. 621\)](#)
- [Granting an IAM User Permission to Pass an IAM Role to an Instance \(p. 622\)](#)
- [Working with IAM Roles \(p. 622\)](#)

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`.

```
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 \(p. 454\)](#).

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 \(p. 622\)](#)
- [Launching an Instance with an IAM Role \(p. 624\)](#)
- [Attaching an IAM Role to an Instance \(p. 625\)](#)
- [Detaching an IAM Role \(p. 626\)](#)
- [Replacing an IAM Role \(p. 627\)](#)

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, type a name for the role and choose **Create role**.

Alternatively, you can use the AWS CLI to create an IAM role.

To create an IAM role and instance profile using the AWS CLI

- Create an IAM role with a policy that allows the role to use an Amazon S3 bucket.

- a. 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"  
        }  
    ]  
}
```

- b. Create the `s3access` role and specify the trust policy that you created.

```
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": "AROAIIZKPBK52LEXAMPLE",  
        "CreateDate": "2013-12-12T23:46:37.247Z",  
        "RoleName": "s3access",  
        "Path": "/",  
        "Arn": "arn:aws:iam::123456789012:role/s3access"  
    }  
}
```

- c. 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": ["*"]
    }
]
```

- d. Attach the access policy to the role.

```
aws iam put-role-policy --role-name s3access --policy-name S3-Permissions --policy-document file://ec2-role-access-policy.json
```

- e. Create an instance profile named s3access-profile.

```
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"
    }
}
```

- f. 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
```

For more information about these commands, see [create-role](#), [put-role-policy](#), and [create-instance-profile](#) in the *AWS CLI Command Reference*.

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 using the 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.

```
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 using the 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

After you've created an IAM role, you can attach it to a running or stopped instance.

To attach an IAM role 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, 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 using the 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"

{
    "IamInstanceProfileAssociation": {
        "InstanceId": "i-1234567890abcdef0",
        "State": "associating",
        "AssociationId": "iip-assoc-0dbd8529a48294120",
        "IamInstanceProfile": {
            "Id": "AIPAJLNLDX3AMYZNWWYAY",
            "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-1"
        }
    }
}
```

Alternatively, use the following Tools for Windows PowerShell commands:

- [Get-EC2Instance](#)
- [Register-EC2IamInstanceProfile](#)

Detaching an IAM Role

You can detach an IAM role from a running or stopped instance.

To detach an IAM role 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, 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 using the 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](#)

Replacing an IAM Role

You can replace an IAM role for a running instance. You can do this if you want to change the IAM role for an instance without detaching the existing one first; for example, to ensure that API actions performed by applications running on the instance are not interrupted.

To replace an IAM role 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, 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 using the 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](#)

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 \(p. 534\)](#) and [Custom Security Groups \(p. 535\)](#). 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.

For more information about security groups, see [Amazon EC2 Security Groups for Linux Instances \(p. 531\)](#).

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 using the 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 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 \(p. 628\)](#).

5. Choose **Save**.

(VPC only) 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 using the 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 in EC2-Classic, you can't change its security groups. After you launch an instance in a VPC, you can change its security groups. For more information, see [Changing an Instance's Security Groups](#) in the *Amazon VPC User Guide*.

Amazon EC2 and Amazon Virtual Private Cloud

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 AWS resources, such as instances, into your VPC. Your VPC closely resembles a traditional network that you might operate in your own data center, with the benefits of using AWS's scalable infrastructure. 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. You can connect your VPC to your own corporate data center, making the AWS cloud an extension of your data center. To protect the resources in each subnet, you can use multiple layers of security, including security groups and network access control lists. For more information, see the [Amazon VPC User Guide](#).

Your account may support both the EC2-VPC and EC2-Classic platforms, on a region-by-region basis. If you created your account after 2013-12-04, it supports EC2-VPC only. To find out which platforms your account supports, see [Supported Platforms \(p. 637\)](#). If your account supports EC2-VPC only, we create a *default VPC* for you. 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. For more information, see [Your Default VPC and Subnets](#) in the *Amazon VPC User Guide*. If your account supports EC2-Classic and EC2-VPC, you can launch instances into either platform. Regardless of which platforms your account supports, you can create your own *nondefault VPC*, and configure it as you need.

Contents

- [Benefits of Using a VPC \(p. 631\)](#)
- [Differences Between EC2-Classic and EC2-VPC \(p. 631\)](#)
- [Sharing and Accessing Resources Between EC2-Classic and EC2-VPC \(p. 634\)](#)

- [Instance Types Available Only in a VPC \(p. 636\)](#)
- [Amazon VPC Documentation \(p. 636\)](#)
- [Supported Platforms \(p. 637\)](#)
- [ClassicLink \(p. 638\)](#)
- [Migrating from a Linux Instance in EC2-Classic to a Linux Instance in a VPC \(p. 648\)](#)

Benefits of Using a VPC

By launching your instances into a VPC instead of EC2-Classic, you gain the ability to:

- Assign static private IPv4 addresses to your instances that persist across starts and stops
- Assign multiple IPv4 addresses to your instances
- Define network interfaces, and attach one or more network interfaces to your instances
- Change security group membership for your instances while they're running
- Control the outbound traffic from your instances (egress filtering) in addition to controlling the inbound traffic to them (ingress filtering)
- Add an additional layer of access control to your instances in the form of network access control lists (ACL)
- Run your instances on single-tenant hardware
- Assign IPv6 addresses to your instances

Differences Between EC2-Classic and EC2-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 Amazon's public IP address pool)	Your instance receives a public IPv4 address.	Your instance launched in a default subnet receives a public IPv4 address by default, unless you specify otherwise during launch, or you modify the subnet's public IPv4 address attribute.	Your instance doesn't receive a public IPv4 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.

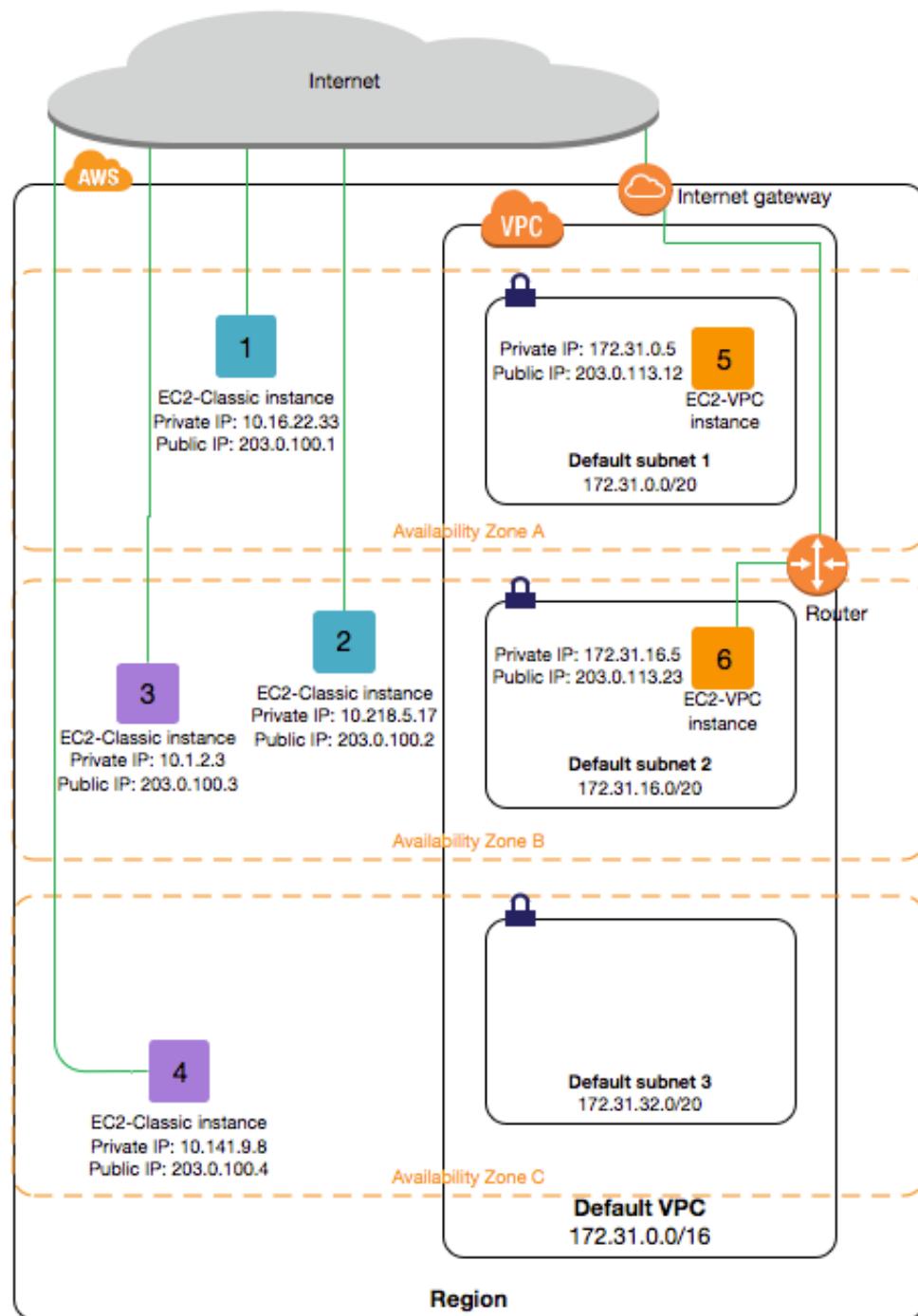
Amazon Elastic Compute Cloud
User Guide for Linux Instances
Differences Between EC2-Classic and EC2-VPC

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
DNS hostnames	DNS hostnames are enabled by default.	DNS hostnames are enabled by default.	DNS hostnames are disabled by default.
Security group	<p>A security group can reference security groups that belong to other AWS accounts.</p> <p>You can create up to 500 security groups in each region.</p>	<p>A security group can reference security groups for your VPC only.</p> <p>You can create up to 500 security groups per VPC.</p>	<p>A security group can reference security groups for your VPC only.</p> <p>You can create up to 500 security groups per VPC.</p>
Security group association	<p>You can assign an unlimited number of security groups to an instance when you launch it.</p> <p>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).</p>	<p>You can assign up to 5 security groups to an instance.</p> <p>You can assign security groups to your instance when you launch it and while it's running.</p>	<p>You can assign up to 5 security groups to an instance.</p> <p>You can assign security groups to your instance when you launch it and while it's running.</p>
Security group rules	<p>You can add rules for inbound traffic only.</p> <p>You can add up to 100 rules to a security group.</p>	<p>You can add rules for inbound and outbound traffic.</p> <p>You can add up to 50 rules to a security group.</p>	<p>You can add rules for inbound and outbound traffic.</p> <p>You can add up to 50 rules to a security group.</p>
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.

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
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.

The following diagram shows instances in each platform. Note the following:

- Instances 1, 2, 3, and 4 are in the EC2-Classic platform. 1 and 2 were launched by one account, and 3 and 4 were launched by a different account. These instances can communicate with each other, can access the Internet directly.
- Instances 5 and 6 are in different subnets in the same VPC in the EC2-VPC platform. They were launched by the account that owns the VPC; no other account can launch instances in this VPC. These instances can communicate with each other and can access instances in EC2-Classic and the Internet through the Internet gateway.



Sharing and Accessing Resources Between EC2-Classic and EC2-VPC

Some resources and features in your AWS account can be shared or accessed between the EC2-Classic and EC2-VPC platforms, for example, through ClassicLink. For more information about ClassicLink, see [ClassicLink \(p. 638\)](#).

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 a Linux Instance in EC2-Classic to a Linux Instance in a VPC \(p. 648\)](#).

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 EC2-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 to EC2-VPC (p. 677) .
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 a Linux Instance in EC2-Classic to a Linux Instance in a VPC (p. 648) .
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 EC2-VPC. For more information, see Modifying Reserved Instances (p. 264) .
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 in EC2-Classic to a security group

Resource	Notes
	in a VPC. For more information, see Creating a Security Group (p. 535) .
Snapshot	

The following resources can't be shared or moved between EC2-Classic and a VPC:

- Spot instances

Instance Types Available Only in a VPC

Instances of the following instance types are not supported in EC2-Classic and must be launched in a VPC:

- General purpose: M4, M5, M5d, T2
- Compute optimized: C4, C5, C5d
- Memory optimized: R4, R5, R5d, X1, z1d
- Storage optimized: H1, I3
- Accelerated computing: F1, G3, P2, P3

If your account supports EC2-Classic but you have not created a nondefault VPC, you can do one of the following to launch a VPC-only instance:

- 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. For more information, see [Create a Virtual Private Cloud \(VPC\) \(p. 23\)](#).
- 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 in Your VPC](#) in the *Amazon VPC User Guide*.

If you have other resources in EC2-Classic, you can take steps to migrate them to EC2-VPC. For more information, see [Migrating from a Linux Instance in EC2-Classic to a Linux Instance in a VPC \(p. 648\)](#).

Amazon VPC Documentation

For more information about Amazon VPC, see the following documentation.

Guide	Description
Amazon VPC Getting Started Guide	Provides a hands-on introduction to Amazon VPC.

Guide	Description
Amazon VPC User Guide	Provides detailed information about how to use Amazon VPC.
Amazon VPC Network Administrator Guide	Helps network administrators configure your customer gateway.

Supported Platforms

Amazon EC2 supports the following platforms. Your AWS account is capable of launching instances either into both platforms or only into EC2-VPC, on a region basis.

Platform	Introduced In	Description
EC2-Classic	The original release of Amazon EC2	Your instances run in a single, flat network that you share with other customers.
EC2-VPC	The original release of Amazon VPC	Your instances run in a virtual private cloud (VPC) that's logically isolated to your AWS account.

For more information about the availability of either platform in your account, see [Availability](#) in the *Amazon VPC User Guide*. For more information about the differences between EC2-Classic and EC2-VPC, see [Differences Between EC2-Classic and EC2-VPC \(p. 631\)](#).

Supported Platforms in the Amazon EC2 Console

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**. If there are two values, **EC2** and **VPC**, you can launch instances into either platform. If there is one value, **VPC**, you can launch instances only into EC2-VPC.

If you can launch instances only into EC2-VPC, we create a default VPC for you. Then, 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.

EC2-VPC

The dashboard displays the following under **Account Attributes** to indicate that the account supports only the EC2-VPC platform, and has a default VPC with the identifier `vpc-1a2b3c4d`.

Supported Platforms

VPC

Default VPC

`vpc-1a2b3c4d`

If your account supports only EC2-VPC, you can select a VPC from the **Network** list, and a subnet from the **Subnet** list when you launch an instance using the launch wizard.

Network	<input type="button" value="i"/> <code>vpc-1a2b3c4d (172.31.0.0/16) (default)</code>	<input type="button" value="C"/> Create new VPC
Subnet	<input type="button" value="i"/> No preference (default subnet in any Availability Zone)	<input type="button" value="Create new subnet"/>

EC2-Classic, EC2-VPC

The dashboard displays the following under **Account Attributes** to indicate that the account supports both the EC2-Classic and EC2-VPC platforms.

Supported Platforms

EC2

VPC

If your account supports EC2-Classic and EC2-VPC, you can launch into EC2-Classic using the launch wizard by selecting **Launch into EC2-Classic** from the **Network** list. To launch into a VPC, you can select a VPC from the **Network** list, and a subnet from the **Subnet** list.

Related Topic

For more information about how you can tell which platforms you can launch instances into, see [Detecting Your Supported Platforms](#) in the *Amazon VPC User Guide*.

ClassicLink

ClassicLink allows you to link your EC2-Classic instance to a VPC in your account, within the same region. This allows you to associate the VPC security groups with the EC2-Classic instance, enabling 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. For more information about private and public IPv4 addresses, see [IP Addressing in Your VPC](#).

ClassicLink is available to all users with accounts that support the EC2-Classic platform, and can be used with any EC2-Classic instance. To find out which platform your account supports, see [Supported Platforms \(p. 637\)](#). For more information about the benefits of using a VPC, see [Amazon EC2 and Amazon Virtual Private Cloud \(p. 630\)](#). For more information about migrating your resources to a VPC, see [Migrating from a Linux Instance in EC2-Classic to a Linux Instance in a VPC \(p. 648\)](#).

There is no additional charge for using ClassicLink. Standard charges for data transfer and instance usage apply.

Note

EC2-Classic instances cannot be enabled 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.

Contents

- [ClassicLink Basics \(p. 638\)](#)
- [ClassicLink Limitations \(p. 641\)](#)
- [Working with ClassicLink \(p. 641\)](#)
- [API and CLI Overview \(p. 645\)](#)
- [Example: ClassicLink Security Group Configuration for a Three-Tier Web Application \(p. 646\)](#)

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.

Note

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 EC2-Classic Instances \(p. 643\)](#).

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 \(p. 644\)](#).

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 \(p. 548\)](#).

For more information about policies for working with ClassicLink, see the following example: [7. Working with ClassicLink \(p. 605\)](#).

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 \(p. 646\)](#).

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 `local1`. 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 `local1`. 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 use ClassicLink to enable communication between an EC2-Classic instance and a C5, C5d, i3.meta1, M5, M5d, R5, R5d, and z1d instance.
- You cannot associate a VPC Elastic IP address with a linked EC2-Classic instance.
- 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 \(p. 640\)](#).
- 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 endpoint, 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 \(p. 642\)](#)
- [Linking an Instance to a VPC \(p. 642\)](#)
- [Creating a VPC with ClassicLink Enabled \(p. 642\)](#)
- [Linking an EC2-Classic Instance to a VPC at Launch \(p. 643\)](#)
- [Viewing Your ClassicLink-Enabled VPCs and Linked EC2-Classic Instances \(p. 643\)](#)
- [Enabling ClassicLink DNS Support \(p. 644\)](#)
- [Disabling ClassicLink DNS Support \(p. 644\)](#)
- [Unlinking a EC2-Classic Instance from a VPC \(p. 644\)](#)
- [Disabling ClassicLink for a VPC \(p. 644\)](#)

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 \(p. 640\)](#).

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. Choose a VPC, and then choose **Actions, Enable ClassicLink**.
4. In the confirmation dialog box, choose **Yes, Enable**.

Linking an Instance to a VPC

After you've enabled a VPC for ClassicLink, you can link an EC2-Classic instance to it.

Note

You can only link a running EC2-Classic instance to a VPC. You cannot link an instance that's in the stopped 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 the running EC2-Classic instance, choose **Actions, ClassicLink, Link to VPC**. You can select more than one instance to link to the same VPC.
4. In the dialog box that displays, select a VPC from the list. Only VPCs that have been enabled for ClassicLink are displayed.
5. Select one or more of the VPC security groups to associate with your instance. When you are done, choose **Link to VPC**.

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 **Start 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*.

Linking an EC2-Classic 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. On the **Configure Instance Details** page, ensure that you select **Launch into EC2-Classic** from the **Network** list.

Note

Some instance types, such as T2 instance types, can only be launched into a VPC. Ensure that you select an instance type that can be launched into EC2-Classic.

4. In the **Link to VPC (ClassicLink)** section, select a VPC from **Link to VPC**. Only ClassicLink-enabled VPCs are displayed. Select the security groups from the VPC to associate with the instance. Complete the other configuration options on the page, and then complete the rest of the steps in the wizard to launch your instance. For more information about using the launch wizard, see [Launching Your Instance from an AMI \(p. 355\)](#).

Viewing Your ClassicLink-Enabled VPCs and Linked EC2-Classic 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 a VPC, and in the **Summary** tab, look for the **ClassicLink** field. A value of **Enabled** indicates that the VPC is enabled for ClassicLink.
4. Alternatively, look for the **ClassicLink** column, and view the value that's displayed for each VPC (**Enabled** or **Disabled**). If the column is not visible, choose **Edit Table Columns** (the gear-shaped icon), select the **ClassicLink** attribute, and then choose **Close**.

To view your linked EC2-Classic instances

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select an EC2-Classic instance, and in the **Description** tab, look for the **ClassicLink** field. If the instance is linked to a VPC, the field displays the ID of the VPC to which the instance is linked. If the instance is not linked to any VPC, the field displays **Unlinked**.
4. Alternatively, you can filter your instances to display only linked EC2-Classic instances for a specific VPC or security group. In the search bar, start typing **ClassicLink**, select the relevant ClassicLink resource attribute, and then select the security group ID or the VPC ID.

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 your VPC, and choose **Actions, Edit ClassicLink DNS Support**.
4. Choose **Yes** to enable ClassicLink DNS support, and choose **Save**.

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 your VPC, and choose **Actions, Edit ClassicLink DNS Support**.
4. Choose **No** to disable ClassicLink DNS support, and choose **Save**.

Unlinking a EC2-Classic 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.

Note

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**, and select your instance.
3. In the **Actions** list, select **ClassicLink, Unlink Instance**. You can select more than one instance to unlink from the same VPC.
4. Choose **Yes** in the confirmation dialog box.

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, then choose **Actions, Disable ClassicLink**.
4. In the confirmation dialog box, choose **Yes, Disable**.

API and CLI Overview

You can perform the tasks described on this page using the command line or the Query API. For more information about the command line interfaces and a list of available API actions, see [Accessing Amazon EC2 \(p. 3\)](#).

Enable a VPC for ClassicLink

- [enable-vpc-classic-link](#) (AWS CLI)
- [Enable-EC2VpcClassicLink](#) (AWS Tools for Windows PowerShell)
- [EnableVpcClassicLink](#) (Amazon EC2 Query API)

Link (attach) an EC2-Classic instance to a VPC

- [attach-classic-link-vpc](#) (AWS CLI)
- [Add-EC2ClassicLinkVpc](#) (AWS Tools for Windows PowerShell)
- [AttachClassicLinkVpc](#) (Amazon EC2 Query API)

Unlink (detach) an EC2-Classic instance from a VPC

- [detach-classic-link-vpc](#) (AWS CLI)
- [Dismount-EC2ClassicLinkVpc](#) (AWS Tools for Windows PowerShell)
- [DetachClassicLinkVpc](#) (Amazon EC2 Query API)

Disable ClassicLink for a VPC

- [disable-vpc-classic-link](#) (AWS CLI)
- [Disable-EC2VpcClassicLink](#) (AWS Tools for Windows PowerShell)
- [DisableVpcClassicLink](#) (Amazon EC2 Query API)

Describe the ClassicLink status of VPCs

- [describe-vpc-classic-link](#) (AWS CLI)
- [Get-EC2VpcClassicLink](#) (AWS Tools for Windows PowerShell)
- [DescribeVpcClassicLink](#) (Amazon EC2 Query API)

Describe linked EC2-Classic instances

- [describe-classic-link-instances](#) (AWS CLI)
- [Get-EC2ClassicLinkInstances](#) (AWS Tools for Windows PowerShell)
- [DescribeClassicLinkInstances](#) (Amazon EC2 Query API)

Enable a VPC peering connection for ClassicLink

- [modify-vpc-peering-connection-options](#) (AWS CLI)
- [Edit-EC2VpcPeeringConnectionOption](#) (AWS Tools for Windows PowerShell)
- [ModifyVpcPeeringConnectionOptions](#) (Amazon EC2 Query API)

Enable a VPC for ClassicLink DNS support

- [enable-vpc-classic-link-dns-support](#) (AWS CLI)
- [Enable-EC2VpcClassicLinkDnsSupport](#) (AWS Tools for Windows PowerShell)
- [EnableVpcClassicLinkDnsSupport](#) (Amazon EC2 Query API)

Disable a VPC for ClassicLink DNS support

- [disable-vpc-classic-link-dns-support](#) (AWS CLI)
- [Disable-EC2VpcClassicLinkDnsSupport](#) (AWS Tools for Windows PowerShell)
- [DisableVpcClassicLinkDnsSupport](#) (Amazon EC2 Query API)

Describe ClassicLink DNS support for VPCs

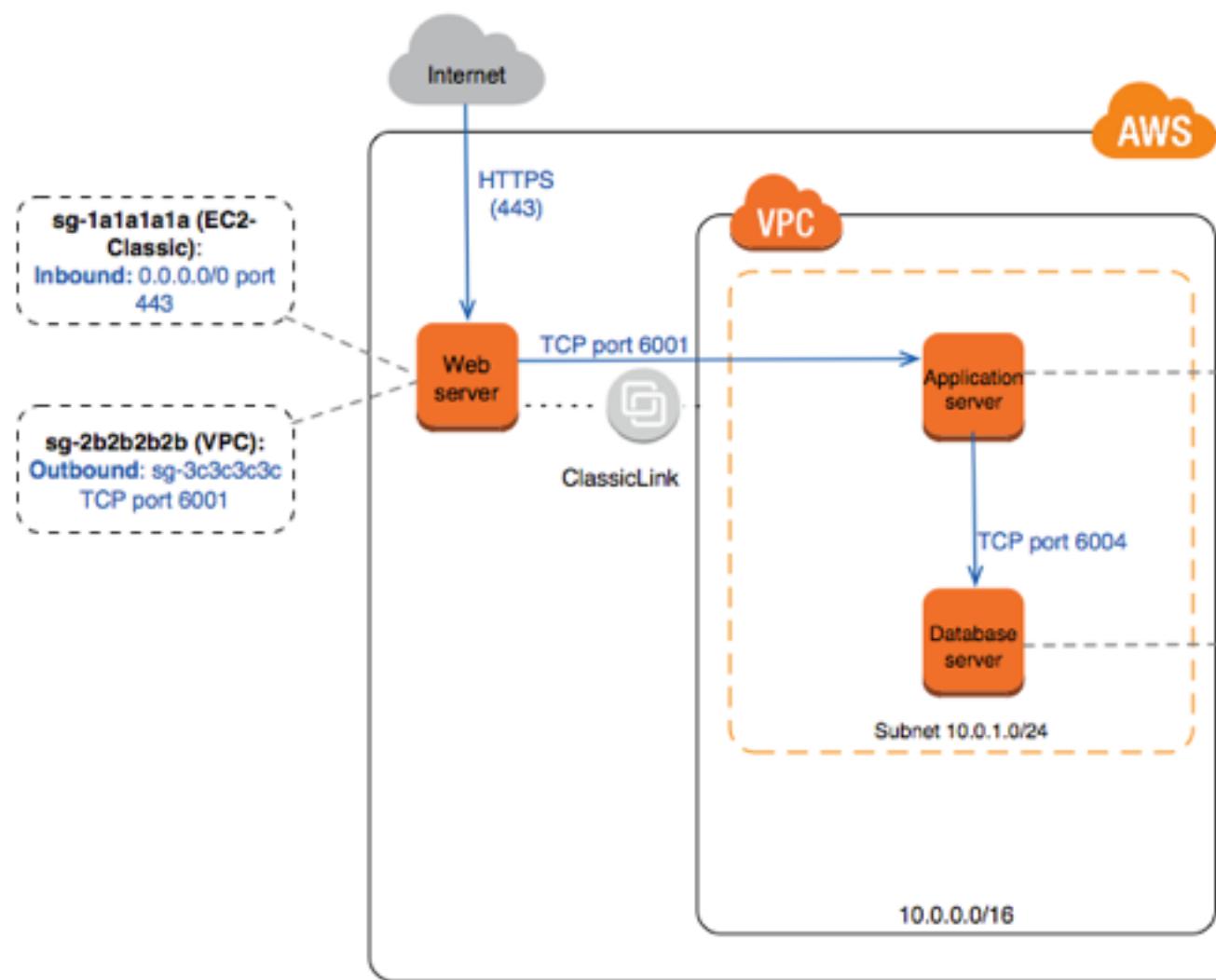
- [describe-vpc-classic-link-dns-support](#) (AWS CLI)
- [Get-EC2VpcClassicLinkDnsSupport](#) (AWS Tools for Windows PowerShell)
- [DescribeVpcClassicLinkDnsSupport](#) (Amazon EC2 Query API)

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 a Linux Instance in EC2-Classic to a Linux Instance in a VPC

Your AWS account might support both EC2-Classic and EC2-VPC, depending on when you created your account and which regions you've used. For more information, and to find out which platform your account supports, see [Supported Platforms \(p. 637\)](#). For more information about the benefits of using a VPC, and the differences between EC2-Classic and EC2-VPC, see [Amazon EC2 and Amazon Virtual Private Cloud \(p. 630\)](#).

You create and use resources in your AWS account. Some resources and features, such as enhanced networking and certain instance types, can be used only in a 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 EC2-VPC \(p. 634\)](#).

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.

There are two ways of migrating to a VPC. You can do a full migration, or you can do an incremental migration over time. The method you choose depends on the size and complexity of your application in EC2-Classic. For example, if your application consists of one or two instances running a static website, and you can afford a short period of downtime, you can do a full migration. If you have a multi-tier application with processes that cannot be interrupted, you can do an incremental migration using ClassicLink. This allows you to transfer functionality one component at a time until your application is running fully in your VPC.

If you need to migrate a Windows instance, see [Migrating a Windows Instance from EC2-Classic to a VPC](#) in the *Amazon EC2 User Guide for Windows Instances*.

Contents

- [Full Migration to a VPC \(p. 649\)](#)
- [Incremental Migration to a VPC Using ClassicLink \(p. 655\)](#)

Full Migration to a VPC

Complete the following tasks to fully migrate your application from EC2-Classic to a VPC.

Tasks

- [Step 1: Create a VPC \(p. 649\)](#)
- [Step 2: Configure Your Security Group \(p. 650\)](#)
- [Step 3: Create an AMI from Your EC2-Classic Instance \(p. 650\)](#)
- [Step 4: Launch an Instance Into Your VPC \(p. 651\)](#)
- [Example: Migrating a Simple Web Application \(p. 652\)](#)

Step 1: Create a VPC

To start using a VPC, ensure that you have one in your account. You can create one using one of these methods:

- Use a new, EC2-VPC-only AWS account. Your EC2-VPC-only account comes with a default VPC in each region, which is ready for you to use. Instances that you launch are by default launched into this VPC, unless you specify otherwise. For more information about your default VPC, see [Your Default VPC and Subnets](#). Use this option if you'd prefer not to set up a VPC yourself, or if you do not need specific requirements for your VPC configuration.
- In your existing AWS account, open the Amazon VPC console and use the VPC wizard to create a new VPC. For more information, see [Scenarios for Amazon VPC](#). Use this option if you want to set up a VPC quickly in your existing EC2-Classic account, using one of the available configuration sets in the wizard. You'll specify this VPC each time you launch an instance.
- In your existing AWS account, open the Amazon VPC console and set up the components of a VPC according to your requirements. For more information, see [Your VPC and Subnets](#). Use this option if you have specific requirements for your VPC, such as a particular number of subnets. You'll specify this VPC each time you launch an instance.

Step 2: Configure Your Security Group

You cannot use the same security groups between EC2-Classic and a VPC. However, 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.

Important

You can only copy security group rules to a new security group in the same AWS account in the same region. If you've created a new AWS account, you cannot use this method to copy your existing security group rules to your new account. You'll have to create a new security group, and add the rules yourself. For more information about creating a new security group, see [Amazon EC2 Security Groups for Linux Instances \(p. 531\)](#).

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**.
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 \(p. 531\)](#).

Note

If you've defined a rule in your EC2-Classic security group that references another security group, you will not be able to use the same rule in your VPC security group. Modify the rule to reference a security group in the same VPC.

6. Choose **Create**.

Step 3: Create an AMI from Your EC2-Classic Instance

An AMI is a template for launching your instance. You can create your own AMI based on an existing EC2-Classic instance, then use that AMI to launch instances into your VPC.

The method you use to create your 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 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, check 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 may have to reinstall your software on a base HVM AMI. For more information about PV and HVM virtualization, see [Linux AMI Virtualization Types \(p. 86\)](#).

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 (p. 101) .
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 (p. 105) .
Instance store	Convert your instance store-backed instance to an EBS-backed instances. For more information, see Converting your Instance Store-Backed AMI to an Amazon EBS-Backed AMI (p. 116) .

(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 \(p. 733\)](#)
- [Creating an Amazon EBS Volume \(p. 747\)](#)
- [Attaching an Amazon EBS Volume to an Instance \(p. 750\)](#)

To back up the data on your Amazon EBS volume, you can take periodic snapshots of your volume. If you need to, you can restore an Amazon EBS volume from your snapshot. For more information about Amazon EBS snapshots, see the following topics:

- [Amazon EBS Snapshots \(p. 792\)](#)
- [Creating an Amazon EBS Snapshot \(p. 795\)](#)
- [Restoring an Amazon EBS Volume from a Snapshot \(p. 748\)](#)

Step 4: Launch an Instance Into Your VPC

After you've created an AMI, you can launch an instance into your VPC. The instance will have the same data and configurations as your existing EC2-Classic instance.

You can either launch your instance into a VPC that you've created in your existing account, or into a new, VPC-only AWS account.

Using Your Existing EC2-Classic Account

You can use the Amazon EC2 launch wizard to launch an instance into your VPC.

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.

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 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 you created earlier. 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 you can configure in each step of the wizard, see [Launching an Instance Using the Launch Instance Wizard \(p. 355\)](#).

Using Your New, VPC-Only Account

To launch an instance in your new AWS account, you'll first have to share the AMI you created with your new account. You can then use the Amazon EC2 launch wizard to launch an instance into your default VPC.

To share an AMI with your new AWS account

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Switch to the account in which you created your AMI.
3. In the navigation pane, choose **AMIs**.
4. In the **Filter** list, ensure **Owned by me** is selected, then select your AMI.
5. In the **Permissions** tab, choose **Edit**. Enter the account number of your new AWS account, choose **Add Permission**, and then choose **Save**.

To launch an instance into your default VPC

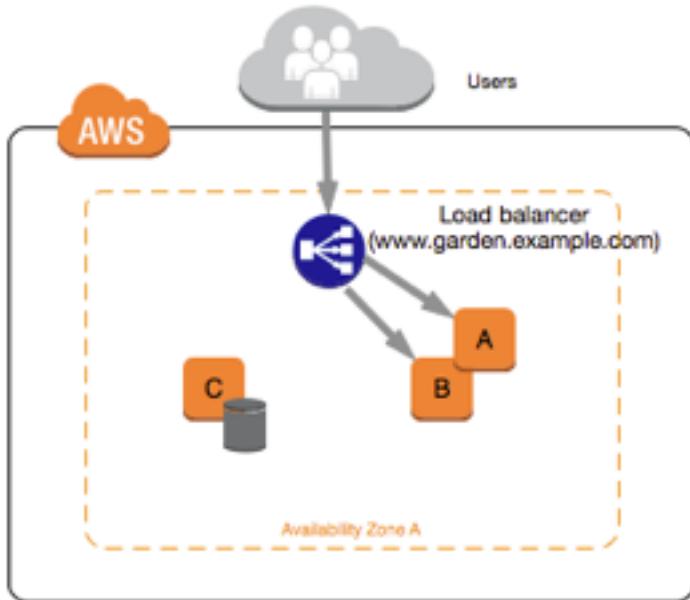
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Switch to your new AWS account.
3. In the navigation pane, choose **AMIs**.
4. In the **Filter** list, select **Private images**. Select the AMI that you shared from your EC2-Classic account, then choose **Launch**.
5. On the **Choose an Instance Type** page, select the type of instance, and choose **Next: Configure Instance Details**.
6. On the **Configure Instance Details** page, your default VPC should be selected in the **Network** list. Configure any other details you require, then go through the next pages of the wizard until you reach the **Configure Security Group** page.
7. Select **Select an existing group**, and select the security group you created earlier. Choose **Review and Launch**.
8. Review your instance details, then choose **Launch** to specify a key pair and launch your instance.

For more information about the parameters you can configure in each step of the wizard, see [Launching an Instance Using the Launch Instance Wizard \(p. 355\)](#).

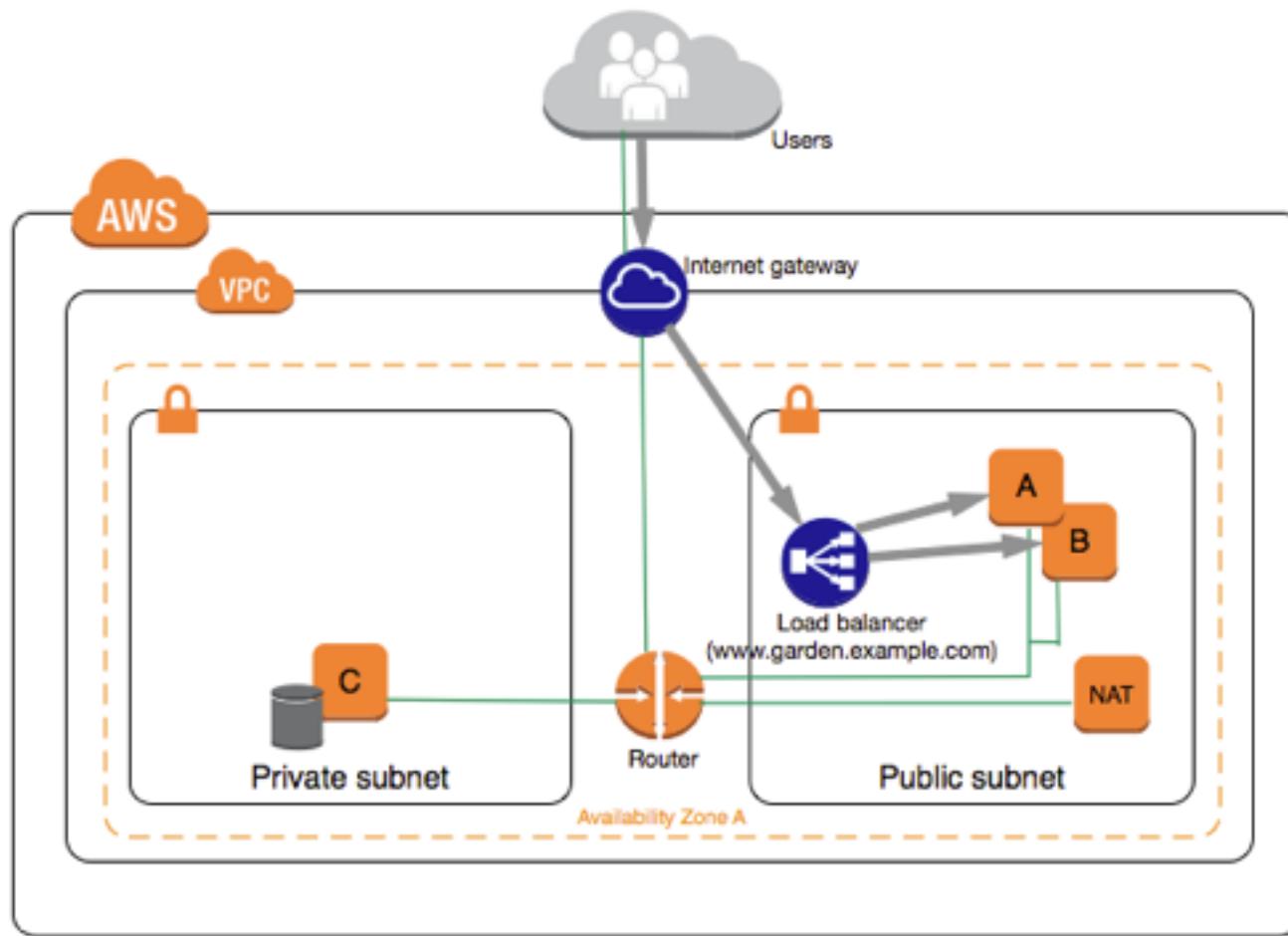
Example: Migrating a Simple Web Application

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 will suit 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 may 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 EC2-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 [Scenario 2: VPC with Public and Private Subnets](#) in the *Amazon VPC User Guide*.
- **Create AMIs from your instances:** Create an AMI from one of your web servers, and a second AMI from your database server. For more information, see [Step 3: Create an AMI from Your EC2-Classic Instance \(p. 650\)](#).
- **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 [Step 2: Configure Your Security Group \(p. 650\)](#).

Tip

Create the security groups that are referenced by other security groups first.

- **Launch an instance into your new VPC:** Launch replacement web servers into your public subnet, and launch your replacement database server into your private subnet. For more information, see [Step 4: Launch an Instance Into Your VPC \(p. 651\)](#).
- **Configure your NAT device:** If you are using a NAT instance, you must create 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 the configuration information that was stored in that instance was copied to the AMI. You may 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'll have 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 may 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 EC2-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 to EC2-VPC \(p. 677\)](#).
- **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 you can configure your load balancer in VPC. For more information, see [Elastic Load Balancing in Amazon VPC](#).
- **Update your DNS records:** After you've set up your load balancer in your public subnet, ensure that your `www.garden.example.com` domain points to your new load balancer. To do this, you'll need to update your DNS records and update 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. Terminate your EC2-Classic instances, and release your EC2-Classic Elastic IP addresses.

Incremental Migration to a VPC Using ClassicLink

The ClassicLink feature makes it easier to manage an incremental migration to a VPC. ClassicLink allows 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 to the VPC one step at a time. This topic provides some basic steps for managing an incremental migration from EC2-Classic to a VPC.

For more information about ClassicLink, see [ClassicLink \(p. 638\)](#).

Topics

- [Step 1: Prepare Your Migration Sequence \(p. 656\)](#)
- [Step 2: Create a VPC \(p. 656\)](#)
- [Step 3: Enable Your VPC for ClassicLink \(p. 656\)](#)
- [Step 4: Create an AMI from Your EC2-Classic Instance \(p. 656\)](#)
- [Step 5: Launch an Instance Into Your VPC \(p. 657\)](#)
- [Step 6: Link Your EC2-Classic Instances to Your VPC \(p. 658\)](#)
- [Step 7: Complete the VPC Migration \(p. 658\)](#)

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.

Step 2: Create a VPC

To start using a VPC, ensure that you have one in your account. You can create one using one of these methods:

- In your existing AWS account, open the Amazon VPC console and use the VPC wizard to create a new VPC. For more information, see [Scenarios for Amazon VPC](#). Use this option if you want to set up a VPC quickly in your existing EC2-Classic account, using one of the available configuration sets in the wizard. You'll specify this VPC each time you launch an instance.
- In your existing AWS account, open the Amazon VPC console and set up the components of a VPC according to your requirements. For more information, see [Your VPC and Subnets](#). Use this option if you have specific requirements for your VPC, such as a particular number of subnets. You'll specify this VPC each time you launch an instance.

Step 3: Enable Your VPC for ClassicLink

After you've created a VPC, you can enable it for ClassicLink. For more information about ClassicLink, see [ClassicLink \(p. 638\)](#).

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 your VPC, and then select **Enable ClassicLink** from the **Actions** list.
4. In the confirmation dialog box, choose **Yes, Enable**.

Step 4: Create an AMI from Your EC2-Classic Instance

An AMI is a template for launching your instance. You can create your own AMI based on an existing EC2-Classic instance, then use that AMI to launch instances into your VPC.

The method you use to create your 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 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, check 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 may have to reinstall your software on a base HVM AMI. For more information about PV and HVM virtualization, see [Linux AMI Virtualization Types \(p. 86\)](#).

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 (p. 101) .
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 (p. 105) .
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 (p. 116) .

(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 \(p. 733\)](#)
- [Creating an Amazon EBS Volume \(p. 747\)](#)
- [Attaching an Amazon EBS Volume to an Instance \(p. 750\)](#)

To back up the data on your Amazon EBS volume, you can take periodic snapshots of your volume. If you need to, you can restore an Amazon EBS volume from your snapshot. For more information about Amazon EBS snapshots, see the following topics:

- [Amazon EBS Snapshots \(p. 792\)](#)
- [Creating an Amazon EBS Snapshot \(p. 795\)](#)
- [Restoring an Amazon EBS Volume from a Snapshot \(p. 748\)](#)

Step 5: Launch an Instance Into Your VPC

The next step in the migration process is to launch instances into your VPC so that you can start transferring functionality to them. You can use the AMIs that you created in the previous step to launch instances into your VPC. The instances will have the same data and configurations as your existing EC2-Classic instances.

To launch an instance into your VPC using your custom AMI

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.
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 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 you created earlier. 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 you can configure in each step of the wizard, see [Launching an Instance Using the Launch Instance Wizard \(p. 355\)](#).

After you've launched your instance and it's in the `running` state, you can connect to it and configure it as required.

Step 6: Link Your EC2-Classic Instances to Your VPC

After you've configured your 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.

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 your EC2-Classic instance, then choose **Actions, ClassicLink, and Link to VPC**.

Note

Ensure that your instance is in the `running` state.

4. In the dialog box, select your ClassicLink-enabled VPC (only VPCs that are enabled for ClassicLink are displayed).
5. Select one or more of the VPC security groups to associate with your instance. When you are done, choose **Link to VPC**.

Step 7: Complete the VPC Migration

Depending on the size of your application and the functionality that must be migrated, repeat steps 4 to 6 until you've moved all 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. You can migrate your Elastic IP addresses that you are currently using in the EC2-Classic platform to the EC2-VPC platform. For more information, see [Migrating an Elastic IP Address from EC2-Classic to EC2-VPC \(p. 677\)](#).

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 may 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.

Amazon EC2 Instance IP Addressing

We provide your instances with IP addresses and IPv4 DNS hostnames. These can vary depending on whether you launched the instance in the EC2-Classic platform or in a virtual private cloud (VPC). For information about the EC2-Classic and EC2-VPC platforms, see [Supported Platforms \(p. 637\)](#).

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*.

IPv6 is not supported for the EC2-Classic platform.

Contents

- [Private IPv4 Addresses and Internal DNS Hostnames \(p. 659\)](#)
- [Public IPv4 Addresses and External DNS Hostnames \(p. 660\)](#)
- [Elastic IP Addresses \(IPv4\) \(p. 661\)](#)
- [Amazon DNS Server \(p. 661\)](#)
- [IPv6 Addresses \(p. 661\)](#)
- [IP Address Differences Between EC2-Classic and EC2-VPC \(p. 662\)](#)
- [Working with IP Addresses for Your Instance \(p. 663\)](#)
- [Multiple IP Addresses \(p. 667\)](#)

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 network (EC2-Classic or a 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 network, but we can't resolve the DNS hostname outside the network that the instance is in.

An instance launched in a VPC receives a primary private IP address from the IPv4 address range of the subnet. For more information, see [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 in a VPC 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 \(p. 667\)](#).

For instances launched in EC2-Classic, we release the private IPv4 address when the instance is stopped or terminated. If you restart your stopped instance, it receives a new private IPv4 address.

For instances launched in a VPC, a private IPv4 address remains associated with the network interface when the instance is stopped and restarted, and is released when the instance is terminated.

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
```

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 outside the network of the instance, and to the private IPv4 address of the instance from within the network of the instance. The public IP address is mapped to the primary private IP address through network address translation (NAT). For more information about NAT, see [RFC 1631: The IP Network Address Translator \(NAT\)](#).

When you launch an instance in EC2-Classic, we automatically assign a public IP address to the instance from the EC2-Classic public IPv4 address pool. You cannot modify this behavior. When you launch an instance into a VPC, your subnet has an attribute that determines whether instances launched into that subnet receive a public IP address from the EC2-VPC public IPv4 address pool. By default, we assign a public IP address to instances launched in a default VPC, and we don't assign a public IP address to instances launched in a nondefault subnet.

You can control whether your instance in a VPC receives a public IP address by doing the following:

- 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 \(p. 664\)](#).

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 the public IP address for your instance when it's stopped or terminated. Your stopped instance receives a new public IP address when it's restarted.
- We release the public IP address for your instance when you associate an Elastic IP address with your instance, or when you associate an Elastic IP address with the primary network interface (`eth0`) of your instance in a VPC. 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 you require a persistent public IP address that can be associated to and from instances as you require, use an Elastic IP address instead. For example, 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 \(p. 674\)](#).

If your instance is in a VPC and you assign it an Elastic IP address, 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 from instances as you require, and 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 \(p. 674\)](#).

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. In EC2-Classic, the Amazon DNS server is located at 172.16.0.23. In EC2-VPC, 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 in a VPC 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 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 \(p. 684\)](#).

IP Address Differences Between EC2-Classic and EC2-VPC

The following table summarizes the differences between IP addresses for instances launched in EC2-Classic, instances launched in a default subnet, and instances launched in a nondefault subnet.

Characteristic	EC2-Classic	Default Subnet	Nondefault Subnet
Public IP address (from Amazon's public IPv4 address pool)	Your instance receives a public IP address.	Your instance receives a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.	Your instance doesn't receive a public IP address by default, unless you specify otherwise during launch, or you modify the subnet's public IP address attribute.
Private IPv4 address	Your instance receives a private IP address from the EC2-Classic range each time it's started.	Your instance receives a static private IP address from the IPv4 address range of your default subnet.	Your instance receives a static private IP address from the IPv4 address range of your subnet.
Multiple IPv4 addresses	We select a single private IP address for your instance; multiple IP addresses are not supported.	You can assign multiple private IP addresses to your instance.	You can assign multiple private IP addresses to your instance.
Network interfaces	IP addresses are associated with the instance; network interfaces aren't supported.	IP addresses are associated with a network interface. Each instance has one or more network interfaces.	IP addresses are associated with a network interface. Each instance has one or more network interfaces.
Elastic IP address (IPv4)	An Elastic IP address is disassociated from your instance when you stop it.	An Elastic IP address remains associated with your instance when you stop it.	An Elastic IP address remains associated with your instance when you stop it.
DNS hostnames (IPv4)	DNS hostnames are enabled by default.	DNS hostnames are enabled by default.	DNS hostnames are disabled by default, except if you've created your VPC using the VPC wizard in the Amazon VPC console.
IPv6 address	Not supported. Your instance cannot receive an IPv6 address.	Your instance does not receive an IPv6 address by default unless you've associated an IPv6 CIDR block with your VPC and subnet, and either specified an IPv6 address during launch, or modified your subnet's IPv6 addressing attribute.	Your instance does not receive an IPv6 address by default unless you've associated an IPv6 CIDR block with your VPC and subnet, and either specified an IPv6 address during launch, or modified your subnet's IPv6 addressing attribute.

Working with IP Addresses for Your Instance

You can view the IP addresses assigned to your instance, assign a public IPv4 address to your instance during launch, or assign an IPv6 address to your instance during launch.

Contents

- [Determining Your Public, Private, and Elastic IP Addresses \(p. 663\)](#)
- [Determining Your IPv6 Addresses \(p. 664\)](#)
- [Assigning a Public IPv4 Address During Instance Launch \(p. 664\)](#)
- [Assigning an IPv6 Address to an Instance \(p. 665\)](#)
- [Unassigning an IPv6 Address From an Instance \(p. 666\)](#)

Determining Your Public, Private, and Elastic IP Addresses

You can use the Amazon EC2 console to determine 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 \(p. 454\)](#).

To determine your instance's private IPv4 addresses 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. In the details pane, get the private IPv4 address from the **Private IPs** field, and get the internal DNS hostname from the **Private DNS** field.
4. (VPC only) If you have one or more secondary private IPv4 addresses assigned to network interfaces that are attached to your instance, get those IP addresses from the **Secondary private IPs** field.
5. (VPC only) Alternatively, in the navigation pane, choose **Network Interfaces**, and then select the network interface that's associated with your instance.
6. Get the primary private IP address from the **Primary private IPv4 IP** field, and the internal DNS hostname from the **Private DNS (IPv4)** field.
7. If you've assigned secondary private IP addresses to the network interface, get those IP addresses from the **Secondary private IPv4 IPs** field.

To determine your instance's public IPv4 addresses 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. In the details pane, get the public IP address from the **IPv4 Public IP** field, and get the external DNS hostname from the **Public DNS (IPv4)** field.
4. If one or more Elastic IP addresses have been associated with the instance, get the Elastic IP addresses from the **Elastic IPs** field.

Note

If your instance does not have a public IPv4 address, but you've associated an Elastic IP address with a network interface for the instance, the **IPv4 Public IP** field displays the Elastic IP address.

5. (VPC only) Alternatively, in the navigation pane, choose **Network Interfaces**, and then select a network interface that's associated with your instance.

6. Get the public IP address from the **IPv4 Public IP** field. An asterisk (*) indicates the public IPv4 address or Elastic IP address that's mapped to the primary private IPv4 address.

Note

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 within the instance, you can use instance metadata.

To determine your instance's IPv4 addresses using instance metadata

1. Connect to your instance.
2. Use the following command to access the private IP address:

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4
```

3. Use the following command to access the public IP address:

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-ipv4
```

Note that if an Elastic IP address is associated with the instance, the value returned is that of the Elastic IP address.

Determining Your IPv6 Addresses

(VPC only) You can use the Amazon EC2 console to determine the IPv6 addresses of your instances.

To determine your instance's IPv6 addresses 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. In the details pane, get the IPv6 addresses from **IPv6 IPs**.

To determine your instance's IPv6 addresses using instance metadata

1. Connect to your instance.
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/`):

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

Assigning a Public IPv4 Address During Instance Launch

If you launch an instance in EC2-Classic, it is assigned a public IPv4 address by default. You can't modify this behavior.

In a VPC, all subnets have 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.

Important

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 \(p. 660\)](#). 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 \(p. 674\)](#).

To access the public IP addressing feature 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 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.

Important

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.

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 \(p. 355\)](#). 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.

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 \(p. 674\)](#). 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 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)

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, an instance type, and choose **Next: Configure Instance Details**.

Note

Ensure that you select an instance type that supports IPv6 addresses. For more information, see [Instance Types \(p. 163\)](#).

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.

Alternatively, you can assign an IPv6 address to your instance after launch.

To assign an IPv6 address to your 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, choose **Actions, Networking, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Assign new IP**. 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 **Save**.

Note

If you launched your instance using Amazon Linux 2016.09.0 or later, or Windows Server 2008 R2 or later, your instance is configured for IPv6, and no additional steps are needed to ensure that the IPv6 address is recognized on the instance. If you launched your instance from an older AMI, you may have to configure your instance manually. For more information, see [Configure IPv6 on Your Instances](#) in the *Amazon VPC User Guide*.

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 using the Amazon EC2 console.

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 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

In EC2-VPC, 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 \(p. 684\)](#).

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 \(p. 667\)](#)
- [Working with Multiple IPv4 Addresses \(p. 668\)](#)
- [Working with Multiple IPv6 Addresses \(p. 671\)](#)

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 can be attached to or detached from 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 the secondary IPv4 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.
- Security groups apply to network interfaces, not to IP addresses. Therefore, IP addresses are subject to the security group of the network interface in which they're specified.
- 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 the network interface when it is detached from an instance or attached to another instance.
- Although you can't move the primary network interface from an instance, you can reassign the secondary private IPv4 address of the primary network interface to another network interface.

- You can move any additional network interface from one instance to another.

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 \(p. 668\)](#)
- [Configuring the Operating System on Your Instance to Recognize the Secondary Private IPv4 Address \(p. 670\)](#)
- [Associating an Elastic IP Address with the Secondary Private IPv4 Address \(p. 670\)](#)
- [Viewing Your Secondary Private IPv4 Addresses \(p. 670\)](#)
- [Unassigning a Secondary Private IPv4 Address \(p. 671\)](#)

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 in EC2-VPC \(p. 668\)](#)
- [To assign a secondary IPv4 address during launch using the command line \(p. 669\)](#)
- [To assign a secondary private IPv4 address to a network interface \(p. 669\)](#)
- [To assign a secondary private IPv4 to an existing instance using the command line \(p. 669\)](#)

To assign a secondary private IPv4 address when launching an instance in EC2-VPC

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 \(p. 684\)](#).

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 \(p. 677\)](#).

- 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.
- 6. 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**.
- 7. On the **Add Tags** page, specify tags for the instance, such as a user-friendly name, and then choose **Next: Configure Security Group**.
- 8. On the **Configure Security Group** page, select an existing security group or create a new one. Choose **Review and Launch**.
- 9. 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 the Secondary Private IPv4 Address \(p. 670\)](#).

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 the Secondary Private IPv4 Address

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 li`. 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 \(p. 691\)](#).
- 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 in EC2-VPC

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 in EC2-VPC

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 \(p. 672\)](#)

- [Viewing Your IPv6 Addresses \(p. 673\)](#)
- [Unassigning an IPv6 Address \(p. 674\)](#)

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 \(p. 163\)](#).
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 \(p. 684\)](#).
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 \(p. 540\)](#). 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).

Elastic IP Addresses

An *Elastic IP address* is a static IPv4 address designed for dynamic cloud computing. An Elastic IP address is associated with your AWS account. With 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 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, to connect to your instance from your local computer.

We currently do not support Elastic IP addresses for IPv6.

Contents

- [Elastic IP Address Basics \(p. 675\)](#)
- [Elastic IP Address Differences for EC2-Classic and EC2-VPC \(p. 675\)](#)
- [Working with Elastic IP Addresses \(p. 677\)](#)
- [Using Reverse DNS for Email Applications \(p. 682\)](#)

- [Elastic IP Address Limit \(p. 683\)](#)

Elastic IP Address Basics

The following are the basic characteristics of an Elastic IP address:

- 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 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. For more information, see [Public IPv4 Addresses and External DNS Hostnames \(p. 660\)](#).
- You can disassociate an Elastic IP address from a resource, and reassociate it with a different resource. Any open connections to an instance continue to work for a time even after you disassociate its Elastic IP address and reassociate it with another instance. We recommend that you reopen these connections using the reassigned Elastic IP address.
- 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 [Amazon EC2 Pricing](#).
- An Elastic IP address is for use in a specific region only.
- When you associate an Elastic IP address with an instance that previously had a public IPv4 address, the public DNS hostname of the instance changes to match the Elastic IP address.
- We resolve a public DNS hostname 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.

If your account supports EC2-Classic, the use and behavior of Elastic IP addresses for EC2-Classic and EC2-VPC may differ. For more information, see [Elastic IP Address Differences for EC2-Classic and EC2-VPC \(p. 675\)](#).

Elastic IP Address Differences for EC2-Classic and EC2-VPC

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 the EC2-VPC platform. 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 to the EC2-VPC platform. You cannot migrate an Elastic IP address to another region. For more information about EC2-Classic and EC2-VPC, see [Supported Platforms \(p. 637\)](#).

When you associate an Elastic IP address with an instance in EC2-Classic, a default VPC, or an instance in a nondefault VPC in which you assigned a public IPv4 to the eth0 network interface during launch, the instance's current public IPv4 address is released back into the public IP address pool. If you disassociate an Elastic IP address from the instance, the instance is automatically assigned a new public IPv4 address within a few minutes. However, if you have attached a second network interface to an instance in a VPC, the instance is not automatically assigned a new public IPv4 address. For more information about public IPv4 addresses, see [Public IPv4 Addresses and External DNS Hostnames \(p. 660\)](#).

For information about using an Elastic IP address with an instance in a VPC, see [Elastic IP Addresses](#) in the [Amazon VPC User Guide](#).

The following table lists the differences between Elastic IP addresses on EC2-Classic and EC2-VPC. For more information about the differences between private and public IP addresses, see [IP Address Differences Between EC2-Classic and EC2-VPC \(p. 662\)](#).

Characteristic	EC2-Classic	EC2-VPC
Allocating an Elastic IP address	When you allocate an Elastic IP address, it's for use in EC2-Classic; however, you can migrate an Elastic IP address to the EC2-VPC platform. For more information, see Migrating an Elastic IP Address from EC2-Classic to EC2-VPC (p. 677) .	When you allocate an Elastic IP address, it's for use only in a VPC.
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 can associate an Elastic IP address with an instance by updating the network interface attached to the instance. For more information, see Elastic Network Interfaces (p. 683) .
Reassociating an Elastic IP address	If you try to associate an Elastic IP address that's already associated with another instance, the address is automatically associated with the new instance.	If your account supports EC2-VPC only, and you try to associate an Elastic IP address that's already associated with another instance, the address is automatically associated with the new instance. If you're using a VPC in an EC2-Classic account, and you try to associate an Elastic IP address that's already associated with another instance, it succeeds only if you allowed reassociation.
Associating an Elastic IP address with a target that has an existing Elastic IP address	The existing Elastic IP address is disassociated from the instance, but remains allocated to your account.	If your account supports EC2-VPC only, the existing Elastic IP address is disassociated from the instance, but remains allocated to your account. If you're using a VPC in an EC2-Classic account, you cannot associate an Elastic IP address with a network interface or instance that has an existing Elastic IP address.
Stopping an instance	If you stop an instance, its Elastic IP address is disassociated, and you must reassociate the Elastic IP address when you restart the instance.	If you stop an instance, its Elastic IP address remains associated.
Assigning multiple IP addresses	Instances support only a single private IPv4 address and a corresponding Elastic IP address.	Instances support multiple IPv4 addresses, and each one can have a corresponding Elastic IP address. For more information, see Multiple IP Addresses (p. 667) .
Tagging Elastic IP addresses	Does not support Elastic IP address tagging.	Supports Elastic IP address tagging. This allows you to assign your own metadata to each Elastic IP address.

Migrating an Elastic IP Address from EC2-Classic to EC2-VPC

If your account supports EC2-Classic, you can migrate Elastic IP addresses that you've allocated for use in the EC2-Classic platform to the EC2-VPC platform, 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 EC2-VPC, you cannot use it in the EC2-Classic platform; however, if required, you can restore it to EC2-Classic. After you've restored an Elastic IP address to EC2-Classic, you cannot use it in EC2-VPC until you migrate it again. You can only migrate an Elastic IP address from EC2-Classic to EC2-VPC. You cannot migrate an Elastic IP address that was originally allocated for use in EC2-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 and Reassociating with a Different Instance \(p. 679\)](#).

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 to EC2-VPC, it counts against your Elastic IP address limit for EC2-VPC. 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 \(p. 683\)](#).

You cannot migrate an Elastic IP address that has been allocated to your account for less than 24 hours.

For more information, see [Moving an Elastic IP Address \(p. 680\)](#).

Working with Elastic IP Addresses

The following sections describe how you can work with Elastic IP addresses.

Tasks

- [Allocating an Elastic IP Address \(p. 677\)](#)
- [Describing Your Elastic IP Addresses \(p. 678\)](#)
- [Tagging an Elastic IP Address \(p. 678\)](#)
- [Associating an Elastic IP Address with a Running Instance \(p. 679\)](#)
- [Disassociating an Elastic IP Address and Reassociating with a Different Instance \(p. 679\)](#)
- [Moving an Elastic IP Address \(p. 680\)](#)
- [Releasing an Elastic IP Address \(p. 681\)](#)
- [Recovering an Elastic IP Address \(p. 682\)](#)

Allocating an Elastic IP Address

You can allocate an Elastic IP address using the Amazon EC2 console or the command line. If your account supports EC2-Classic, you can allocate an address for use in EC2-Classic or in EC2-VPC.

To allocate an Elastic IP address for use in EC2-VPC 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. (EC2-Classic accounts) Choose **VPC**, and then choose **Allocate**. Close the confirmation screen.
5. (VPC-only accounts) Choose **Allocate**, and close the confirmation screen.

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.

To allocate 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 \(p. 3\)](#).

- [allocate-address](#) (AWS CLI)
- [New-EC2Address](#) (AWS Tools for Windows PowerShell)

Describing Your Elastic IP Addresses

You can describe an Elastic IP address using the Amazon EC2 or the command line.

To describe your Elastic IP addresses 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. Select a filter from the Resource Attribute list to begin searching. You can use multiple filters in a single search.

To describe your Elastic IP 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-addresses](#) (AWS CLI)
- [Get-EC2Address](#) (AWS Tools for Windows PowerShell)

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 you've assigned it.

Note

Cost allocation tracking using Elastic IP address tags is not supported.

You can tag an Elastic IP address using the Amazon EC2 console or the command line tools.

To tag an Elastic IP address 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. 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**.

To tag an Elastic IP address using the command line

Use one of the following commands:

- [create-tags](#) (AWS CLI)

```
aws ec2 create-tags --resources eipalloc-12345678 --tags Key=Owner,Value=TeamA
```

- [New-EC2Tag](#) (AWS Tools for Windows PowerShell)

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

You can associate an Elastic IP address to an instance using the Amazon EC2 console or the command line.

(VPC only) 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*.

To associate an Elastic IP address with 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 IPs**.
3. Select an Elastic IP address and choose **Actions, Associate address**.
4. Select the instance from **Instance** and then choose **Associate**.

To associate 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 \(p. 3\)](#).

- [associate-address](#) (AWS CLI)
- [Register-EC2Address](#) (AWS Tools for Windows PowerShell)

Disassociating an Elastic IP Address and Reassociating with a Different Instance

You can disassociate an Elastic IP address and then reassociate it using the Amazon EC2 console or the command line.

To disassociate and reassociate an Elastic IP address 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. Select the Elastic IP address, choose **Actions**, and then select **Disassociate address**.
4. Choose **Disassociate address**.
5. Select the address that you disassociated in the previous step. For **Actions**, choose **Associate address**.
6. Select the new instance from **Instance**, and then choose **Associate**.

To disassociate 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 \(p. 3\)](#).

- [disassociate-address](#) (AWS CLI)
- [Unregister-EC2Address](#) (AWS Tools for Windows PowerShell)

To associate 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 \(p. 3\)](#).

- [associate-address](#) (AWS CLI)
- [Register-EC2Address](#) (AWS Tools for Windows PowerShell)

Moving an Elastic IP Address

You can move an Elastic IP address from EC2-Classic to EC2-VPC 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 to EC2-VPC 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**.

Note

You can tag an Elastic IP address after you have moved it from EC2-Classic to EC2-VPC.

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**.

Note

If you choose to restore a previously migrated Elastic IP address to EC2-Classic, the tags assigned to the Elastic IP address after migration are lost.

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 to EC2-VPC, 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 Amazon EC2 Query API or AWS CLI

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)
- [MoveAddressToVpc](#) (Amazon EC2 Query API)
- [Move-EC2AddressToVpc](#) (AWS Tools for Windows PowerShell)

To restore an Elastic IP address to EC2-Classic using the Amazon EC2 Query API or AWS CLI

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)
- [RestoreAddressToClassic](#) (Amazon EC2 Query API)
- [Restore-EC2AddressToClassic](#) (AWS Tools for Windows PowerShell)

To describe the status of your moving addresses using the Amazon EC2 Query API or AWS CLI

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)
- [DescribeMovingAddresses](#) (Amazon EC2 Query API)
- [Get-EC2Address](#) (AWS Tools for Windows PowerShell)

To retrieve the allocation ID for your migrated Elastic IP address in EC2-VPC

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

- [describe-addresses](#) (AWS CLI)
- [DescribeAddresses](#) (Amazon EC2 Query API)
- [Get-EC2Address](#) (AWS Tools for Windows PowerShell)

Releasing an Elastic IP Address

If you no longer need an Elastic IP address, we recommend that you release it (the address must not be associated with an instance). You incur charges for any Elastic IP address that's allocated for use with EC2-Classic but not associated with an instance.

You can release an Elastic IP address using the Amazon EC2 console or the command line.

To release an Elastic IP address 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. Select the Elastic IP address, choose **Actions**, and then select **Release addresses**. Choose **Release** when prompted.

To release 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 \(p. 3\)](#).

- [release-address](#) (AWS CLI)
- [Remove-EC2Address](#) (AWS Tools for Windows PowerShell)

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 can only recover an Elastic IP address that was originally allocated for use in EC2-VPC, or that was moved from EC2-Classic to EC2-VPC.
- You cannot recover an Elastic IP address if it has been allocated to another AWS account, or if it will result in your exceeding your Elastic IP address limit.
- You cannot recover tags associated with an Elastic IP address.

Currently, you can recover an Elastic IP address using the Amazon EC2 API or a command line tool only.

To recover an Elastic IP address using the command line

1. (AWS CLI) Use the [allocate-address](#) command and specify the IP address using the `--address` parameter.

```
aws ec2 allocate-address --domain vpc --address 203.0.113.3
```

2. (AWS Tools for Windows PowerShell) Use the [New-EC2Address](#) command and specify the IP address using the `-Address` parameter.

```
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 you provision one or more Elastic IP addresses and provide them to us. 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 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.

If you feel your architecture warrants additional Elastic IP addresses, complete the [Amazon EC2 Elastic IP Address Request Form](#). Describe your use case so that we can understand your need for additional addresses.

Elastic Network Interfaces

An elastic network interface (referred to as a *network interface* in this documentation) is a logical networking component in a VPC that represents a virtual network card.

A network interface 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 \(p. 700\)](#).

All network interfaces have the `eni-xxxxxxxx` resource identifier.

Important

The term 'elastic network interface' is sometimes shortened to 'ENI'. This is not the same as the Elastic Network Adapter (ENA), which is a custom interface that optimizes network performance on some instance types. For more information, see [Enhanced Networking on Linux \(p. 708\)](#).

Contents

- [Network Interface Basics \(p. 684\)](#)
- [IP Addresses Per Network Interface Per Instance Type \(p. 684\)](#)
- [Scenarios for Network Interfaces \(p. 689\)](#)
- [Best Practices for Configuring Network Interfaces \(p. 691\)](#)
- [Working with Network Interfaces \(p. 692\)](#)

- [Requester-Managed Network Interfaces \(p. 700\)](#)

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.

You can also modify the attributes of your network interface, including changing its security groups and managing its IP addresses.

Every instance in a VPC has a default network interface, called the *primary network interface* (eth0). 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 \(p. 684\)](#).

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 (eth0) 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 for eth0, the public IPv4 addressing attribute is determined by the network interface.

For more information, see [Public IPv4 Addresses and External DNS Hostnames \(p. 660\)](#).

IPv6 addresses for network interfaces

You can associate an IPv6 CIDR block with your VPC and subnet, and assign one or more IPv6 addresses from the subnet range to a 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 (eth0) that's created.

For more information, see [IPv6 Addresses \(p. 661\)](#).

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. For more information, see [Multiple IP](#)

[Addresses \(p. 667\)](#). For more information about IPv6 in VPC, see [IP Addressing in Your VPC](#) in the [Amazon VPC User Guide](#).

Instance Type	Maximum Network Interfaces	IPv4 Addresses per Interface	IPv6 Addresses per Interface
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
c5.9xlarge	8	30	30
c5.18xlarge	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.18xlarge	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

Amazon Elastic Compute Cloud
 User Guide for Linux Instances
 IP Addresses Per Network Interface Per Instance Type

Instance Type	Maximum Network Interfaces	IPv4 Addresses per Interface	IPv6 Addresses per Interface
d2.8xlarge	8	30	30
f1.2xlarge	4	15	15
f1.16xlarge	8	50	50
g2.2xlarge	4	15	IPv6 not supported
g2.8xlarge	8	30	IPv6 not supported
g3.4xlarge	8	30	30
g3.8xlarge	8	30	30
g3.16xlarge	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
i3.8xlarge	8	30	30
i3.16xlarge	15	50	50
i3.metal	15	50	50
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

Amazon Elastic Compute Cloud
User Guide for Linux Instances
IP Addresses Per Network Interface Per Instance Type

Instance Type	Maximum Network Interfaces	IPv4 Addresses per Interface	IPv6 Addresses per Interface
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
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
m5.12xlarge	8	30	30
m5.24xlarge	15	50	50
m5d.large	3	10	10
m5d.xlarge	4	15	15
m5d.2xlarge	4	15	15
m5d.4xlarge	8	30	30
m5d.12xlarge	8	30	30
m5d.24xlarge	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
r3.large	3	10	10

Amazon Elastic Compute Cloud
 User Guide for Linux Instances
 IP Addresses Per Network Interface Per Instance Type

Instance Type	Maximum Network Interfaces	IPv4 Addresses per Interface	IPv6 Addresses per Interface
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
r5.2xlarge	4	15	15
r5.4xlarge	8	30	30
r5.12xlarge	8	30	30
r5.24xlarge	15	50	50
r5d.large	3	10	10
r5d.xlarge	4	15	15
r5d.2xlarge	4	15	15
r5d.4xlarge	8	30	30
r5d.12xlarge	8	30	30
r5d.24xlarge	15	50	50
t1.micro	2	2	IPv6 not supported
t2.nano	2	2	2
t2.micro	2	2	2
t2.small	2	4	4
t2.medium	3	6	6
t2.large	3	12	12
t2.xlarge	3	15	15
t2.2xlarge	3	15	15

Instance Type	Maximum Network Interfaces	IPv4 Addresses per Interface	IPv6 Addresses per Interface
x1.16xlarge	8	30	30
x1.32xlarge	8	30	30
x1e.xlarge	3	10	10
x1e.2xlarge	4	15	15
x1e.4xlarge	4	15	15
x1e.8xlarge	4	15	15
x1e.16xlarge	8	30	30
x1e.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

Scenarios for Network Interfaces

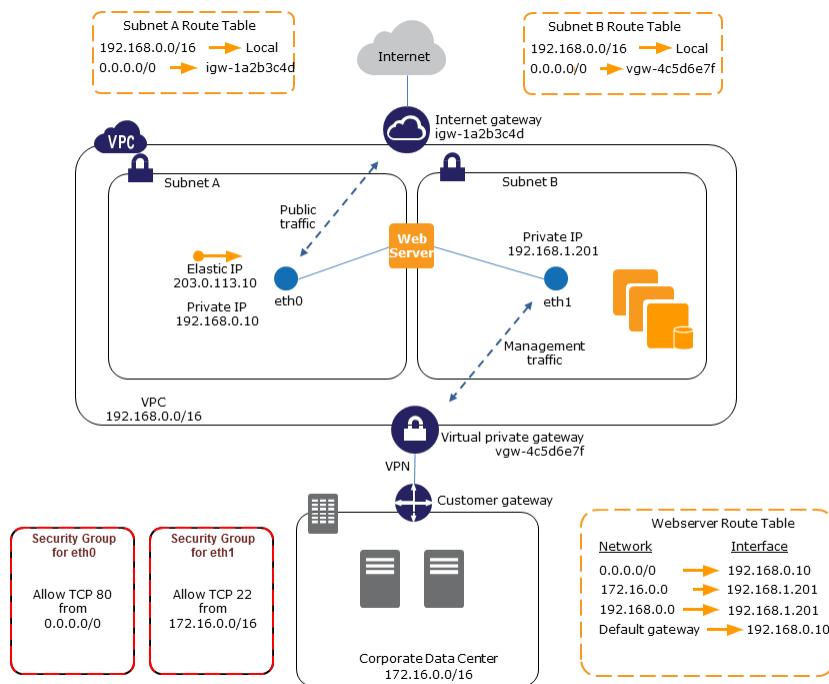
Attaching multiple network interfaces to an instance is useful when you want to:

- 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, the primary network interface (eth0) on the instance handles public traffic and 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) while 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 (ethN) network interfaces when the instance is running or stopped. However, you can't detach the primary (eth0) interface.
- You can attach a network interface in one subnet to an instance in another subnet in the same VPC; however, both the network interface and the instance must reside in the same Availability Zone.
- When launching an instance from the CLI or API, you can specify the network interfaces to attach to the instance for both the primary (eth0) 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 may encounter networking issues such as asymmetric routing. If possible, use a secondary private IPv4 address on the primary network interface instead. For more information, see [Assigning a Secondary Private IPv4 Address \(p. 668\)](#).

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
```

Working with Network Interfaces

You can work with network interfaces using the Amazon EC2 console or the command line.

Contents

- [Creating a Network Interface \(p. 693\)](#)
- [Deleting a Network Interface \(p. 693\)](#)
- [Viewing Details about a Network Interface \(p. 694\)](#)
- [Attaching a Network Interface When Launching an Instance \(p. 694\)](#)
- [Attaching a Network Interface to a Stopped or Running Instance \(p. 695\)](#)
- [Detaching a Network Interface from an Instance \(p. 696\)](#)
- [Changing the Security Group \(p. 696\)](#)
- [Changing the Source or Destination Checking \(p. 697\)](#)
- [Associating an Elastic IP Address \(IPv4\) \(p. 697\)](#)
- [Disassociating an Elastic IP Address \(IPv4\) \(p. 698\)](#)
- [Assigning an IPv6 Address \(p. 698\)](#)
- [Unassigning an IPv6 Address \(p. 698\)](#)
- [Changing Termination Behavior \(p. 699\)](#)
- [Adding or Editing a Description \(p. 699\)](#)

- [Adding or Editing Tags \(p. 700\)](#)

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. For **Security groups**, select one or more security groups.
9. 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 \(AWS Tools for Windows PowerShell\)](#)

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 \(AWS Tools for Windows PowerShell\)](#)

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 When Launching an Instance

You can specify an existing network interface or attach an additional network interface when you launch an instance.

Note

If an error occurs when attaching a network interface to your instance, this causes the instance launch to fail.

To attach a network interface 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. Select an AMI and instance type and choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, select a VPC for **Network**, and a subnet for **Subnet**.
5. In the **Network Interfaces** section, the console enables you to specify up to two network interfaces (new, existing, or a combination) when you launch an instance. You can also enter a primary IPv4 address and one or more secondary IPv4 addresses for any new interface.

You can add additional network interfaces to the instance after you launch it. 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 \(p. 684\)](#).

Note

If you specify more than one network interface, you cannot auto-assign a public IPv4 address to your instance.

6. (IPv6 only) If you're launching an instance into a subnet that has an associated IPv6 CIDR block, you can specify IPv6 addresses for any network interfaces that you attach. 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.
7. Choose **Next: Add Storage**.
8. On the **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**.
9. On the **Add Tags** page, specify tags for the instance, such as a user-friendly name, and then choose **Next: Configure Security Group**.
10. On the **Configure Security Group** page, you can select a security group or create a new one. Choose **Review and Launch**.

Note

If you specified an existing network interface in step 5, the instance is associated with the security group for that network interface, regardless of any option that you select in this step.

11. On the **Review Instance Launch** page, details about the primary and additional network interface are displayed. Review the 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.

To attach a network interface when launching 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 \(p. 3\)](#).

- [run-instances](#) (AWS CLI)
- [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

Attaching a Network Interface to a Stopped or Running Instance

You can attach a network interface to any of your stopped or running instances in your VPC, using either the **Instances** or **Network Interfaces** pages of the Amazon EC2 console.

Note

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 \(p. 660\)](#).

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. Choose **Actions, Networking, Attach Network Interface**.
4. In the **Attach Network Interface** dialog box, 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. In the **Attach Network Interface** dialog box, 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 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. Choose **Actions, Networking, Detach Network Interface**.
4. In the **Detach Network Interface** dialog box, select the network interface and choose **Detach**.

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 choose **Detach**.
4. In the **Detach Network Interface** dialog box, choose **Yes, Detach**. If the network interface fails to detach from the instance, choose **Force detachment**, and then try again.

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)

Changing the Security Group

You can change the security groups that are associated with a network interface. When you create the security group, be sure to specify the same VPC as the subnet for the network interface.

Note

To change security group membership for interfaces owned by other services, such as Elastic Load Balancing, use the console or command line interface for that service.

To change the security group 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. In the **Change Security Groups** dialog box, select the security groups to use, and choose **Save**.

To change the security group of 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 \(p. 3\)](#).

- [modify-network-interface-attribute](#) (AWS CLI)
- [Edit-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

Changing the Source or Destination Checking

The Source/Destination Check attribute controls whether source/destination checking is enabled on the instance. Disabling this attribute 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 set this value to disabled. The default value is enabled.

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 change source/destination checking 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 \(p. 3\)](#).

- [modify-network-interface-attribute](#) (AWS CLI)
- [Edit-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

Associating an Elastic IP Address (IPv4)

If you have an Elastic IP address (IPv4), 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.

You can associate an Elastic IP address using the Amazon EC2 console or the command line.

To associate an Elastic IP address 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, Associate Address**.
4. In the **Associate Elastic IP Address** dialog box, select the Elastic IP address from the **Address** list.
5. For **Associate to private IP address**, select the private IPv4 address to associate with the Elastic IP address.
6. 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**.

To associate 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 \(p. 3\)](#).

- [associate-address](#) (AWS CLI)
- [Register-EC2Address](#) (AWS Tools for Windows PowerShell)

Disassociating an Elastic IP Address (IPv4)

If the network interface has an Elastic IP address (IPv4) associated with it, you can disassociate the address, and then either associate it with another network interface or 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 using a network interface, as network interfaces are specific to a particular subnet.

You can disassociate an Elastic IP address using the Amazon EC2 console or the command line.

To disassociate an Elastic IP address 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, Disassociate Address**.
4. In the **Disassociate IP Address** dialog box, choose **Yes, Disassociate**.

To disassociate 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 \(p. 3\)](#).

- [disassociate-address](#) (AWS CLI)
- [Unregister-EC2Address](#) (AWS Tools for Windows PowerShell)

Assigning an IPv6 Address

You can assign one or more IPv6 addresses to a network interface. The network interface must be 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.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces** and select the network interface.
3. Choose **Actions, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Assign new IP**. Specify an IPv6 address from the range of the subnet. To let AWS choose an address for you, leave the **Auto-assign** value.
5. Choose **Yes, Update**.

To assign an IPv6 address to 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 \(p. 3\)](#).
 - [assign-ipv6-addresses](#) (AWS CLI)
 - [Register-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell)

Unassigning an IPv6 Address

You can unassign an IPv6 address from a network interface using the Amazon EC2 console.

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces** and select the network interface.
3. Choose **Actions, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Unassign** for the IPv6 address to remove.
5. Choose **Yes, Update**.

To unassign an IPv6 address from 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 \(p. 3\)](#).
 - `unassign-ipv6-addresses` (AWS CLI)
 - `Unregister-EC2Ipv6AddressList` (AWS Tools for Windows PowerShell)

Changing 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.

You can change the terminating behavior for a network interface using the Amazon EC2 console or the command line.

To change the termination behavior 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 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 the termination behavior 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 \(p. 3\)](#).

- `modify-network-interface-attribute` (AWS CLI)
- `Edit-EC2NetworkInterfaceAttribute` (AWS Tools for Windows PowerShell)

Adding or Editing a Description

You can change the description for a network interface using the Amazon EC2 console or the command line.

To change the description 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 Description**.
4. In the **Change Description** dialog box, enter a description for the network interface, and then choose **Save**.

To change the description 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 \(p. 3\)](#).

- [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 \(p. 887\)](#).

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)

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 \(p. 700\)](#).

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-c8fbc27e",  
    "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.

Placement Groups

You can launch or start instances in a *placement group*, which determines how instances are placed on underlying hardware. When you create a placement group, you specify one of the following strategies for the group:

- *Cluster*—clusters instances into a low-latency group in a single Availability Zone
- *Spread*—spreads instances across underlying hardware

There is no charge for creating a placement group.

Contents

- [Cluster Placement Groups \(p. 702\)](#)
- [Spread Placement Groups \(p. 702\)](#)
- [Placement Group Rules and Limitations \(p. 702\)](#)
- [Creating a Placement Group \(p. 703\)](#)
- [Launching Instances in a Placement Group \(p. 704\)](#)
- [Changing the Placement Group for an Instance \(p. 704\)](#)

- [Deleting a Placement Group \(p. 705\)](#)

Cluster Placement Groups

A cluster placement group is a logical grouping of instances within a single Availability Zone.

Cluster placement groups are recommended for applications that benefit from low network latency, high network throughput, or both, and if 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 \(p. 708\)](#).

We recommend that you launch the number of instances that you need in the placement group in a single launch request and that you 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. Restarting the instances may migrate them to hardware that has capacity for all the requested instances.

Spread Placement Groups

A spread placement group is a group of instances that are each placed on distinct underlying hardware.

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 underlying hardware. Spread placement groups provide access to distinct hardware, and are therefore suitable for mixing instance types or launching instances over time.

A spread placement group can span multiple Availability Zones, and 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

Before you use placement groups, be aware of the following rules:

- The name 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.
- Reserved Instances 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.
- Instances with a tenancy of host cannot be launched in placement groups.

The following rules apply to cluster placement groups:

- The following are the only instance types that you can use when you launch an instance into a cluster placement group:
 - General purpose: M4, M5, M5d
 - Compute optimized: C3, C4, C5, C5d, cc2.8xlarge
 - Memory optimized: cr1.8xlarge, R3, R4, R5, R5d, X1, X1e, z1d
 - Storage optimized: D2, H1, hs1.8xlarge, I2, I3, i3.metal
 - Accelerated computing: F1, G2, G3, P2, P3
- 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 10–Gbps or 25–Gbps network connectivity.
- For instances that are enabled for enhanced networking, the following rules apply:
 - Traffic between instances within the same region that is addressed using private IPv4 or IPv6 addresses can use 5 Gbps for single-flow traffic and up to 25 Gbps for multi-flow traffic. A flow represents a single, point-to-point network connection.
 - Instances within a cluster placement group can use up to 10 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.

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 zone and in the same spread placement group, the instance will not launch. If you need to have more than seven instances in an AZ, then the recommendation is to use multiple spread placement groups. This does not provide guarantees about the spread of instances between groups, but does ensure the spread for each group to limit impact from certain classes of failures.
- Spread placement groups are not supported for Dedicated Instances or Dedicated Hosts.

Creating a Placement Group

You can create a placement group using the Amazon EC2 console or the command line.

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 and choose the strategy.
4. Choose **Create**.

To create a placement group using the command line

- [create-placement-group](#) (AWS CLI)
- [New-EC2PlacementGroup](#) (AWS Tools for Windows PowerShell)

Launching Instances in a Placement Group

You can create an AMI specifically for the instances to be launched in a placement group. To do this, launch an instance and install the required software and applications on the instance. Then, create an AMI from the instance. For more information, see [Creating an Amazon EBS-Backed Linux AMI \(p. 101\)](#).

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 Amazon Machine Image (AMI)** page, select an AMI. To select an AMI you created, choose **My AMIs**.
 - 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, enter the total number of instances that you need in this placement group, as you might not be able to add instances to the placement group later.
 - On the **Configure Instance Details** page, select the placement group that you created from **Placement group**. If you do not see the **Placement group** list on this page, verify that you have selected an instance type that can be launched into a placement group, as this option is not available otherwise.

To launch instances into a placement group using the command line

1. Create an AMI for your instances using one of the following commands:
 - `create-image` (AWS CLI)
 - `New-EC2Image` (AWS Tools for Windows PowerShell)
2. Launch instances into your placement group using one of the following options:
 - `--placement` with `run-instances` (AWS CLI)
 - `-PlacementGroup` with `New-EC2Instance` (AWS Tools for Windows PowerShell)

Changing the Placement Group for an Instance

You can move an existing instance to a placement group, move an instance from one placement group to another, or remove an instance from a placement group. Before you begin, the instance must be in the stopped state.

You can change the placement group for an instance using the command line or an AWS SDK.

To move an instance to a placement group using the command line

1. Stop the instance using one of the following commands:
 - `stop-instances` (AWS CLI)
 - `Stop-EC2Instance` (AWS Tools for Windows PowerShell)
2. Use the `modify-instance-placement` command (AWS CLI) and specify the name of the placement group to which to move the instance.

```
aws ec2 modify-instance-placement --instance-id i-0aa51192b00939a40 --group-name MySpreadGroup
```

Alternatively, use the [Edit-EC2InstancePlacement](#) command (AWS Tools for Windows PowerShell).

3. Restart the instance using one of the following commands:
 - [start-instances](#) (AWS CLI)
 - [Start-EC2Instance](#) (AWS Tools for Windows PowerShell)

To remove an instance from a placement group using the command line

1. Stop the instance using one of the following commands:
 - [stop-instances](#) (AWS CLI)
 - [Stop-EC2Instance](#) (AWS Tools for Windows PowerShell)
2. Use the [modify-instance-placement](#) command (AWS CLI) and specify an empty string for the group name.

```
aws ec2 modify-instance-placement --instance-id i-0aa51192b00939a40 --group-name ""
```

Alternatively, use the [Edit-EC2InstancePlacement](#) command (AWS Tools for Windows PowerShell).

3. Restart the instance using one of the following commands:
 - [start-instances](#) (AWS CLI)
 - [Start-EC2Instance](#) (AWS Tools for Windows PowerShell)

Deleting a Placement Group

If you need to replace a placement group or no longer need one, you can delete it. Before you can delete your placement group, you must terminate all instances that you launched into the placement group, or move them to another placement group.

To terminate or move instances and 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 **Instances**.
3. Select and terminate all instances in the placement group. You can verify that the instance is in a placement group before you terminate it by checking the value of **Placement Group** in the details pane.

Alternatively, follow the steps in [Changing the Placement Group for an Instance \(p. 704\)](#) to move the instances to a different placement group.

4. In the navigation pane, choose **Placement Groups**.
5. Select the placement group and choose **Delete Placement Group**.
6. When prompted for confirmation, choose **Delete**.

To terminate instances and delete a placement group using the command line

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

- [terminate-instances](#) and [delete-placement-group](#) (AWS CLI)
- [Remove-EC2Instance](#) and [Remove-EC2PlacementGroup](#) (AWS Tools for Windows PowerShell)

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.

Contents

- [Jumbo Frames \(9001 MTU\) \(p. 706\)](#)
- [Path MTU Discovery \(p. 706\)](#)
- [Check the Path MTU Between Two Hosts \(p. 707\)](#)
- [Check and Set the MTU on Your Linux Instance \(p. 707\)](#)
- [Troubleshooting \(p. 708\)](#)

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 \(p. 701\)](#).

The following instances support jumbo frames:

- General purpose: M3, M4, M5, M5d, T2
- Compute optimized: C3, C4, C5, C5d, CC2
- Accelerated computing: F1, G2, G3, P2, P3
- Memory optimized: CR1, R3, R4, R5, R5d, X1, z1d
- Storage optimized: D2, H1, HS1, I2, I3

Path MTU Discovery

Path MTU Discovery is used to determine the path MTU between two devices. The path MTU is the maximum packet size that's supported on the path 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. To ensure that your instance can receive this message and the packet does not get dropped, you must add a **Custom ICMP Rule** with the **Destination Unreachable** protocol to the inbound security group rules for your instance. For more information, see [Path MTU Discovery \(p. 543\)](#).

Important

Modifying your instance's security group to allow 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
```

- For Amazon Linux, add the following lines to your `/etc/dhcp/dhclient-eth0.conf` file.

```
interface "eth0" {
    supersede interface-mtu 1500;
}
```

- For Ubuntu, add the following line to `/etc/network/interfaces.d/eth0.cfg`.

```
post-up /sbin/ifconfig eth0 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*

Enhanced Networking on Linux

Enhanced networking uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on [supported instance types \(p. 709\)](#). 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.

Contents

- [Enhanced Networking Types \(p. 709\)](#)
- [Enabling Enhanced Networking on Your Instance \(p. 709\)](#)
- [Enabling Enhanced Networking with the Intel 82599 VF Interface on Linux Instances \(p. 709\)](#)
- [Enabling Enhanced Networking with the Elastic Network Adapter \(ENA\) on Linux Instances \(p. 716\)](#)
- [Troubleshooting the Elastic Network Adapter \(ENA\) \(p. 723\)](#)

Enhanced Networking Types

Depending on your instance type, enhanced networking can be enabled using one of the following mechanisms:

Intel 82599 Virtual Function (VF) interface

The Intel 82599 Virtual Function interface supports network speeds of up to 10 Gbps for supported instance types.

C3, C4, D2, I2, M4 (excluding m4.16xlarge), and R3 instances use the Intel 82599 VF interface for enhanced networking.

Elastic Network Adapter (ENA)

The Elastic Network Adapter (ENA) supports network speeds of up to 25 Gbps for supported instance types.

C5, C5d, F1, G3, H1, I3, m4.16xlarge, M5, M5d, P2, P3, R4, R5, R5d, X1, and z1d instances use the Elastic Network Adapter for enhanced networking.

To find out which instance types support 10 or 25 Gbps network speeds, see [Amazon EC2 Instance Types](#).

Enabling Enhanced Networking on Your Instance

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 \(p. 709\)](#).

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 \(p. 716\)](#).

Enabling Enhanced Networking with the Intel 82599 VF Interface on Linux Instances

Amazon EC2 provides enhanced networking capabilities to C3, C4, D2, I2, M4 (excluding m4.16xlarge), and R3 instances through the Intel 82599 VF interface, which uses the Intel ixgbevf driver.

Contents

- [Requirements \(p. 709\)](#)
- [Testing Whether Enhanced Networking is Enabled \(p. 710\)](#)
- [Enabling Enhanced Networking on Amazon Linux \(p. 711\)](#)
- [Enabling Enhanced Networking on Ubuntu \(p. 712\)](#)
- [Enabling Enhanced Networking on Other Linux Distributions \(p. 714\)](#)
- [Troubleshooting Connectivity Issues \(p. 716\)](#)

Requirements

To prepare for enhanced networking using the Intel 82599 VF interface, set up your instance as follows:

- 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.

- Launch the instance in a VPC. (You can't enable enhanced networking if the instance is in EC2-Classic.)
- 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; otherwise, it is set as follows:

```
"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-image-attribute](#) (AWS CLI)

```
aws ec2 describe-image-attribute --image-id ami_id --attribute sriovNetSupport
```

Note that this command only works for images that you own. You receive an `AuthFailure` error for images that do not belong to your account.

- [Get-EC2ImageAttribute](#) (AWS Tools for Windows PowerShell)

```
Get-EC2ImageAttribute -ImageId ami_id -Attribute sriovNetSupport
```

If the attribute isn't set, `SriovNetSupport` is empty; otherwise, it is set as follows:

```
"SriovNetSupport": {  
    "Value": "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 wish to check. If you are using a single interface (default), it will be `eth0`.

```
[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 the above case, the `ixgbevf` module is not loaded, because the listed driver is `vif`.

```
[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
```

In this case, the `ixgbevf` module is loaded. This instance has enhanced networking properly configured.

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 a 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 \(p. 710\)](#).

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 \(p. 710\)](#).
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 \(instance store-backed instances\) \(p. 712\)](#).

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 \(p. 101\)](#). 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 \(p. 710\)](#).

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 \(p. 105\)](#), 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 \(p. 710\)](#) on your instance.

The Quick Start Ubuntu HVM AMIs include the necessary drivers for enhanced networking. If you have version 2.16.4 or later of `ixgbevf`, you do not need to compile a different version of the `ixgbevf` module unless you want additional functionality provided in a newer version. Versions of `ixgbevf` earlier than 2.16.4, including version 2.14.2, do not build properly on certain versions of Ubuntu.

If you've launched an instance from a different Ubuntu AMI, contact the AMI provider to confirm if you need a newer version of the enhanced networking drivers.

The following procedure provides the general steps for compiling the `ixgbevf` module on an Ubuntu instance.

To enable enhanced networking on Ubuntu

1. Connect to your instance.
2. Update the package cache and packages.

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y
```

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. Download the source of the `ixgbevf` module appropriate for the kernel version of your instance from Sourceforge at <https://sourceforge.net/projects/e1000/files/ixgbevf%20stable/>; for example:

```
ubuntu:~$ wget "https://sourceforge.net/projects/e1000/files/ixgbevf_stable/4.3.3/ixgbevf-4.3.3.tar.gz"
```

Note

Use the `uname -r` command to get the kernel version for your instance.

4. Decompress and unarchive the `ixgbevf` package.

```
ubuntu:~$ tar -xzf ixgbevf-4.3.3.tar.gz
```

5. Add, build, and install the `ixgbevf` module on your instance.
6. Rebuild `initramfs` so the correct module is loaded at boot time.

```
ubuntu:~$ sudo update-initramfs -c -k all
```

7. Verify that the `ixgbevf` module is installed and at the minimum recommended version using the `modinfo ixgbevf` command.

```
ubuntu:~$ modinfo ixgbevf
```

8. [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 \(instance store-backed instances\) \(p. 714\)](#).

9. From your local computer, enable the enhanced networking `sriovNetSupport` attribute using one of the following commands. Note that there is no way to disable this attribute after you've enabled it.

- [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"
```

10. (Optional) Create an AMI from the instance, as described in [Creating an Amazon EBS-Backed Linux AMI \(p. 101\)](#). The AMI inherits the enhanced networking sriovNetSupport attribute from the instance. Therefore, you can use this AMI to launch another instance with enhanced networking enabled by default.
11. 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.
12. (Optional) Connect to your instance and verify that the module is installed.

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 \(p. 105\)](#), 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 Other Linux Distributions

Before you begin, [check if enhanced networking is already enabled \(p. 710\)](#) 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, which 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 \(instance store-backed instances\) \(p. 716\)](#).

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 \(p. 101\)](#). 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 \(p. 105\)](#), 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.

Enabling Enhanced Networking with the Elastic Network Adapter (ENA) on Linux Instances

Amazon EC2 provides enhanced networking capabilities to C5, C5d, F1, G3, H1, I3, `m4.16xlarge`, M5, M5d, P2, P3, R4, R5, R5d, X1, and z1d instances through the Elastic Network Adapter (ENA).

Contents

- [Requirements \(p. 717\)](#)
- [Testing Whether Enhanced Networking Is Enabled \(p. 717\)](#)
- [Enabling Enhanced Networking on the Amazon Linux AMI \(p. 719\)](#)
- [Enabling Enhanced Networking on Ubuntu \(p. 720\)](#)
- [Enabling Enhanced Networking on Linux \(p. 721\)](#)
- [Troubleshooting \(p. 723\)](#)

Requirements

To prepare for enhanced networking using the ENA, set up your instance as follows:

- Launch the instance from an HVM AMI using Linux kernel version of 3.2 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, ENA enhanced networking is already enabled for your instance.
- Launch the instance in a VPC. (You can't enable enhanced networking if the instance is in EC2-Classic.)
- 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.

Testing Whether Enhanced Networking Is Enabled

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 follows:

```
[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 \(p. 720\)](#).

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 wish to check. If you are using a single interface (default), it will be eth0.

```
[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 the above case, the ena module is not loaded, because the listed driver is vif.

```
[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
```

In this case, the ena module is loaded and at the minimum recommended version. This instance has enhanced networking properly configured.

Enabling Enhanced Networking on the Amazon Linux AMI

Amazon Linux 2 and the latest versions of the Amazon Linux AMI have the module required for enhanced networking installed and have the required `enaSupport` attribute set. 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 \(p. 717\)](#).

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 \(p. 717\)](#).
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 \(instance store-backed instances\) \(p. 720\)](#).

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 \(p. 101\)](#). 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 \(p. 717\)](#).

If you are unable to connect to your instance after enabling enhanced networking, see [Troubleshooting the Elastic Network Adapter \(ENA\) \(p. 723\)](#).

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 \(p. 105\)](#), 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 have the module required for enhanced networking with ENA installed and have the required `enaSupport` attribute set. 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 \(p. 717\)](#).

If you launched your instance using an older AMI and it does not have enhanced networking enabled already, the following are the general steps to enable enhanced networking on an Ubuntu instance.

To enable enhanced networking on Ubuntu

1. Connect to your instance.
2. Update the package cache and packages.

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y
```

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. Install the `build-essential` package to compile the kernel module.

```
ubuntu:~$ sudo apt-get install -y build-essential
```

4. Clone the source code 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
```

5. Add, build, and install the `ena` module on your instance.
6. Rebuild `initramfs` so the correct module is loaded at boot time.

```
ubuntu:~$ sudo update-initramfs -c -k all
```

7. Verify that the `ena` module is installed using the `modinfo ena` command from [Testing Whether Enhanced Networking Is Enabled \(p. 717\)](#).

```
ubuntu:~$ modinfo ena
```

8. [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 \(instance store-backed instances\) \(p. 721\)](#).

9. 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
```

10. (Optional) Create an AMI from the instance, as described in [Creating an Amazon EBS-Backed Linux AMI \(p. 101\)](#). 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.
11. 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.
12. (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 \(ENA\) \(p. 723\)](#).

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 \(p. 105\)](#), 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 following procedure provides the general steps for enabling enhanced networking on a Linux distribution other than Amazon Linux AMI or Ubuntu, such as SUSE Linux Enterprise Server, Red Hat Enterprise Linux, or CentOS. Before you begin, see [Testing Whether Enhanced Networking Is Enabled \(p. 717\)](#) to check if your instance is already enabled for enhanced networking. 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. Clone the source code for the ena module on your instance from GitHub at <https://github.com/amzn/amzn-drivers>.

```
git clone https://github.com/amzn/amzn-drivers
```

3. Compile and install the ena module on your instance.
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 \(instance store-backed instances\) \(p. 723\)](#).

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 \(p. 101\)](#). 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 \(ENA\) \(p. 723\)](#).

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 \(p. 105\)](#), 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 ...
```

Troubleshooting

For additional information about troubleshooting your ENA adapter, see [Troubleshooting the Elastic Network Adapter \(ENA\) \(p. 723\)](#).

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 \(p. 724\)](#) 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 \(p. 724\)](#)
- [Keep-Alive Mechanism \(p. 725\)](#)
- [Register Read Timeout \(p. 726\)](#)
- [Statistics \(p. 726\)](#)
- [Driver Error Logs in syslog \(p. 728\)](#)

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 \(instance store-backed instances\) \(p. 724\)](#).

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 \(p. 716\)](#).

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 \(p. 105\)](#). 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: codeN`

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_refill_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.

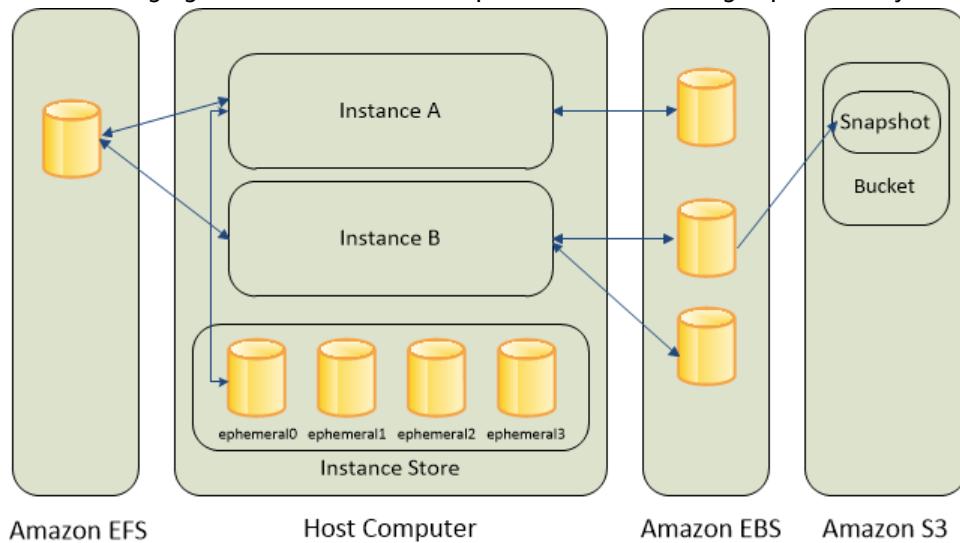
Storage

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 \(p. 731\)](#)
- [Amazon EC2 Instance Store \(p. 847\)](#)
- [Amazon Elastic File System \(Amazon EFS\) \(p. 858\)](#)
- [Amazon Simple Storage Service \(Amazon S3\) \(p. 862\)](#)

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 \(p. 819\)](#).

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 \(p. 731\)](#).

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 \(p. 847\)](#).

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 [Amazon Elastic File System \(Amazon EFS\) \(p. 858\)](#).

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 [Amazon Simple Storage Service \(Amazon S3\) \(p. 862\)](#).

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 \(p. 866\)](#).

You can also attach EBS volumes to a running instance. For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 750\)](#).

Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances. EBS volumes are highly available and reliable storage volumes that can be attached to any running instance that is in the same Availability Zone. EBS volumes that are attached to an EC2 instance are exposed as storage volumes that persist independently from the life of the instance. 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](#).

Amazon EBS is recommended when data 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.

For simplified data encryption, you can launch your EBS volumes as encrypted volumes. Amazon EBS encryption offers you a simple encryption solution for your EBS volumes without the need for you to build, manage, and secure your own key management infrastructure. 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 hosts EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage. For more information, see [Amazon EBS Encryption \(p. 819\)](#).

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 the AWS Key Management Service. 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](#).

You can attach multiple volumes to the same instance within the limits specified by your AWS account. 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 [Request to Increase the Amazon EBS Volume Limit](#).

Contents

- [Features of Amazon EBS \(p. 732\)](#)
- [Amazon EBS Volumes \(p. 733\)](#)
- [Amazon EBS Snapshots \(p. 792\)](#)
- [Amazon EBS–Optimized Instances \(p. 812\)](#)
- [Amazon EBS Encryption \(p. 819\)](#)
- [Amazon EBS and NVMe \(p. 823\)](#)
- [Amazon EBS Volume Performance on Linux Instances \(p. 824\)](#)
- [Amazon CloudWatch Events for Amazon EBS \(p. 840\)](#)

Features of Amazon EBS

- You can create EBS General Purpose SSD (gp2), Provisioned IOPS SSD (io1), Throughput Optimized HDD (st1), and Cold HDD (sc1) volumes up to 16 TiB in size. You can mount these volumes as devices on your Amazon EC2 instances. You can mount multiple volumes on the same instance, but each volume can be attached to only one instance at a time. You can dynamically change the configuration of a volume attached to an instance. For more information, see [Creating an Amazon EBS Volume \(p. 747\)](#).
- With General Purpose SSD (gp2) volumes, you can expect base performance of 3 IOPS/GiB, with the ability to burst to 3,000 IOPS for extended periods of time. Gp2 volumes are ideal for a broad range of use cases such as boot volumes, small and medium-size databases, and development and test environments. Gp2 volumes support up to 10,000 IOPS and 160 MB/s of throughput. For more information, see [General Purpose SSD \(gp2\) Volumes \(p. 737\)](#).
- With Provisioned IOPS SSD (io1) volumes, you can provision a specific level of I/O performance. Io1 volumes support up to 32,000 IOPS and 500 MB/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 \(io1\) Volumes \(p. 740\)](#).
- Throughput Optimized HDD (st1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With throughput of up to 500 MiB/s, this volume type is a good fit for large, sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing. For more information, see [Throughput Optimized HDD \(st1\) Volumes \(p. 741\)](#).
- Cold HDD (sc1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With throughput of up to 250 MiB/s, sc1 is a good fit ideal 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. For more information, see [Cold HDD \(sc1\) Volumes \(p. 743\)](#).
- EBS volumes behave like raw, unformatted block devices. You can create a file system on top of these volumes, or use them in any other way you would use a block device (like a hard drive). For more

information on creating file systems and mounting volumes, see [Making an Amazon EBS Volume Available for Use on Linux \(p. 751\)](#).

- You can use encrypted EBS volumes to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. For more information, see [Amazon EBS Encryption \(p. 819\)](#).
- 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 \(p. 792\)](#).
- 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. For more information, see [Creating an Amazon EBS Snapshot \(p. 795\)](#), [Restoring an Amazon EBS Volume from a Snapshot \(p. 748\)](#), and [Copying an Amazon EBS Snapshot \(p. 799\)](#).
- A large repository of public data set snapshots can be restored to EBS volumes and seamlessly integrated into AWS cloud-based applications. For more information, see [Using Public Data Sets \(p. 875\)](#).
- 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 \(p. 824\)](#).

Amazon EBS Volumes

An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance. 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. After a volume is attached to an instance, you can use it like any other physical hard drive. EBS volumes are flexible. For current-generation volumes attached to current-generation instance types, you can dynamically increase size, modify provisioned IOPS capacity, and change volume type on live production volumes. Amazon EBS provides the following volume types: General Purpose SSD (gp2), Provisioned IOPS SSD (io1), 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 \(p. 735\)](#).

Contents

- [Benefits of Using EBS Volumes \(p. 734\)](#)
- [Amazon EBS Volume Types \(p. 735\)](#)
- [Creating an Amazon EBS Volume \(p. 747\)](#)
- [Restoring an Amazon EBS Volume from a Snapshot \(p. 748\)](#)
- [Attaching an Amazon EBS Volume to an Instance \(p. 750\)](#)
- [Making an Amazon EBS Volume Available for Use on Linux \(p. 751\)](#)
- [Viewing Volume Information \(p. 754\)](#)
- [Monitoring the Status of Your Volumes \(p. 754\)](#)
- [Detaching an Amazon EBS Volume from an Instance \(p. 767\)](#)
- [Deleting an Amazon EBS Volume \(p. 769\)](#)
- [Modifying the Size, IOPS, or Type of an EBS Volume on Linux \(p. 769\)](#)

Benefits of Using EBS Volumes

EBS volumes provide several benefits that are not supported by instance store volumes.

- **Data availability**

When you create an EBS volume in an Availability Zone, it is automatically replicated within that zone to prevent data loss due to failure of any single hardware component. After you create a volume, you can attach it 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. The instance can format the EBS volume with a file system, such as ext3, and then install applications.

An EBS volume can be attached to only one instance at a time, but multiple volumes can be attached to a single instance. 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.

An EBS volume and the instance to which it attaches must be in the same Availability Zone.

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 [Monitoring Volumes with CloudWatch \(p. 754\)](#). For information about tracking the status of your volumes, see [Amazon CloudWatch Events for Amazon EBS](#).

- **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.

By default, EBS volumes that are attached to a running instance automatically detach from the instance with their data intact when that instance is terminated. The volume can then be reattached to a new instance, enabling quick recovery. 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 \(p. 819\)](#).

By default, EBS volumes that are created and attached to an instance at launch are 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.

- **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 \(p. 819\)](#).

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 restored 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 in order to restore the volume.

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 \(p. 887\)](#).

- **Flexibility**

EBS volumes support live configuration changes while in production. You can modify volume type, volume size, and IOPS capacity without service interruptions.

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

The following table describes the use cases and performance characteristics for each volume type:

	Solid-State Drives (SSD)		Hard disk Drives (HDD)	
Volume Type	General Purpose SSD (gp2)*	Provisioned IOPS SSD (io1)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance	Highest-performance SSD volume for mission-critical low-latency workloads	Low cost HDD volume designed for high-throughput workloads	Lowest cost HDD volume designed for less frequently accessed data

	Solid-State Drives (SSD)		Hard disk Drives (HDD)	
	performance for a wide variety of workloads	latency or high-throughput workloads	frequently accessed, throughput-intensive workloads	accessed workloads
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 10,000 IOPS or 160 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
API Name	gp2	io1	st1	sc1
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max. IOPS**/Volume	10,000	32,000***	500	250
Max. Throughput/Volume	160 MiB/s****	500 MiB/s†	500 MiB/s	250 MiB/s
Max. IOPS/Instance	80,000	80,000	80,000	80,000
Max. Throughput/Instance††	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s

* Default volume type for EBS volumes created from the console is gp2. Volumes created using the CreateVolume API default to either gp2 or standard according to region:

- standard: us-east-1, eu-west-1, eu-central-1, us-west-2, us-west-1, sa-east-1, ap-northeast-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-south-1, us-gov-west-1, cn-north-1
- gp2: All other regions

** gp2/io1 based on 16 KiB I/O size, st1/sc1 based on 1 MiB I/O size

*** io1 volumes created in regions ap-northeast-3 and us-gov-west-1 are subject to a 20,000 IOPS limit.

**** General Purpose SSD (gp2) volumes have a throughput limit between 128 MB/s and 160 MB/s depending on volume size.

† An io1 volume created before 12/6/2017 will not achieve this throughput until modified in some way. For more information, see [Modifying the Size, IOPS, or Type of an EBS Volume on Linux](#).

†† To achieve this throughput, you must have an instance that supports it. For more information, see [Amazon EBS-Optimized Instances](#).

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](#).

Previous Generation Volumes	
Volume Type	EBS Magnetic
Description	Previous generation HDD
Use Cases	Workloads where data is infrequently accessed
API Name	standard
Volume Size	1 GiB-1 TiB
Max. IOPS/Volume	40–200
Max. Throughput/Volume	40–90 MiB/s
Max. IOPS/Instance	80,000
Max. Throughput/Instance	1,750 MiB/s
Dominant Performance Attribute	IOPS

Note

Linux AMIs require GPT partition tables and GRUB 2 for boot volumes 2 TiB (2048 GiB) or larger. Many Linux AMIs today use the MBR partitioning scheme, which only supports up to 2047 GiB boot volumes. If your instance does not boot with a boot volume that is 2 TiB or larger, the AMI you are using may be limited to a 2047 GiB boot volume size. Non-boot volumes do not have this limitation on Linux instances.

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 \(p. 824\)](#).

For more information about pricing for these volume types, see [Amazon EBS Pricing](#).

General Purpose SSD (gp2) Volumes

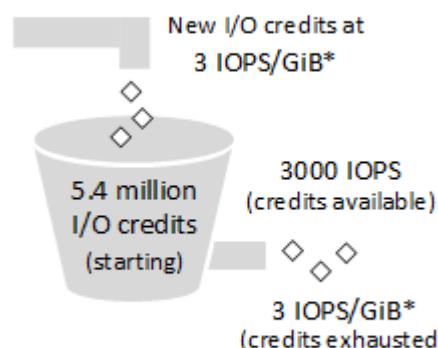
General Purpose SSD (gp2) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. Between a minimum of 100 IOPS (at 33.33 GiB and below) and a

maximum of 10,000 IOPS (at 3,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size. AWS designs gp2 volumes to deliver the 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

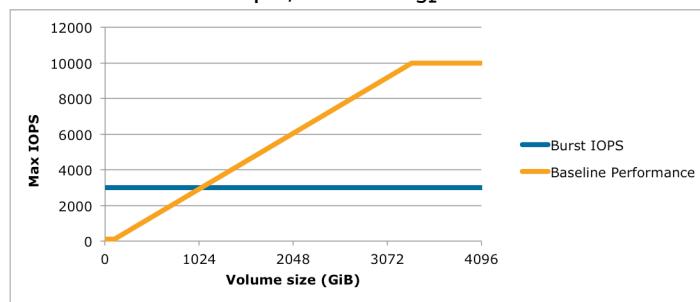
The performance of gp2 volumes is tied to volume size, which determines the baseline performance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher baseline performance levels and accumulate I/O credits faster. I/O credits represent the available bandwidth that 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 10,000 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 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. Volumes larger than 1,000 GiB have a baseline performance that is equal or greater than the maximum burst performance, and their I/O credit balance never depletes. 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).

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)	Maximum burst duration @ 3,000 IOPS (seconds)	Seconds to fill empty credit balance
1	100	1862	54,000
100	300	2,000	18,000
214 (Min. size for max. throughput)	642	2,290	8,412
250	750	2,400	7,200
500	1,500	3,600	3,600
750	2,250	7,200	2,400
1,000	3,000	N/A*	N/A*
3,334 (Min. size for max. IOPS)	10,000	N/A*	N/A*
16,384 (16 TiB, max. volume size)	10,000	N/A*	N/A*

* Bursting and I/O credits are only relevant to volumes under 1,000 GiB, where burst performance exceeds baseline performance.

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:

$$\text{Burst duration} = \frac{(\text{Credit balance})}{(\text{Burst IOPS}) - 3(\text{Volume size in GiB})}$$

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 160 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](#).

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 volume for workloads that require sustained IOPS performance greater than 10,000 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 \(p. 747\)](#).

Throughput Performance

Throughput for a gp2 volume can be calculated using the following formula, up to the throughput limit of 160 MiB/s:

$$(\text{Volume size in GiB}) \times (\text{IOPS per GiB}) \times (\text{I/O size in KiB}) = \text{Throughput in MiB/s}$$

Therefore the smallest volume size that achieves the maximum throughput is given by:

$$\frac{(160 \text{ MiB/s})}{(3 \text{ IOPS/GiB})} = 214 \text{ GiB}$$

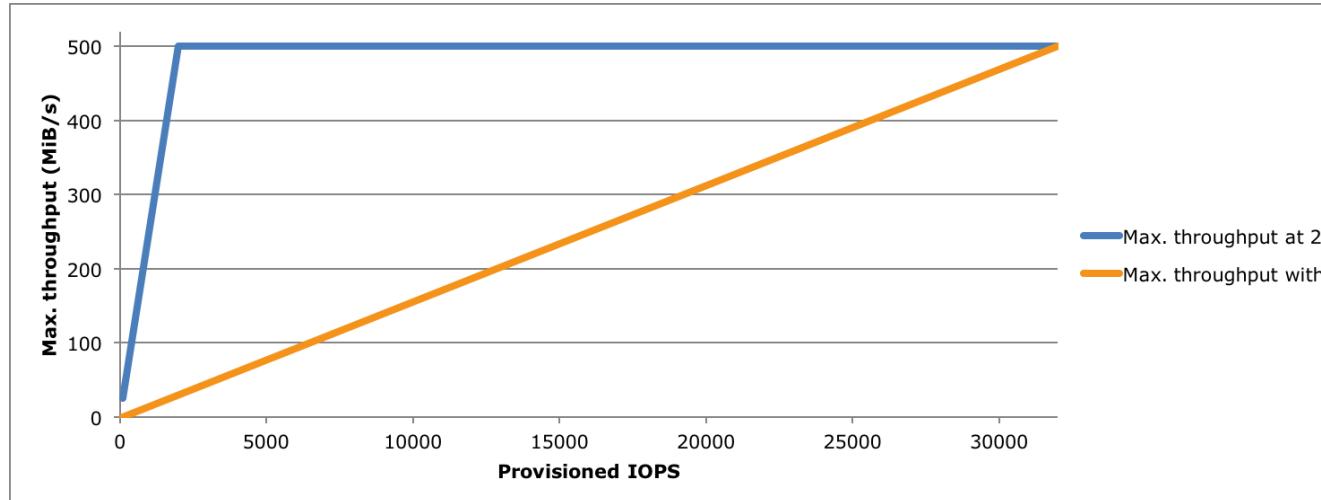
(256 KiB)

Provisioned IOPS SSD (io1) Volumes

Provisioned IOPS SSD (io1) 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, an io1 volume allows you to specify a consistent IOPS rate when you create the volume, and Amazon EBS delivers within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.

An io1 volume can range in size from 4 GiB to 16 TiB and you can provision 100 up to 32,000 IOPS per volume. The maximum ratio of provisioned IOPS to requested volume size (in GiB) is 50:1. For example, a 100 GiB volume can be provisioned with up to 5,000 IOPS. Any volume 640 GiB in size or greater allows provisioning up to the 32,000 IOPS maximum ($50 \times 640 \text{ GiB} = 32,000$).

The throughput limit of io1 volumes is 256 KiB/s for each IOPS provisioned, up to a maximum of 500 MiB/s (at 32,000 IOPS).



Your per-I/O latency experience depends on the IOPS provisioned and your workload pattern. For the best per-I/O latency experience, we recommend that you provision an IOPS-to-GiB ratio greater than 2:1. For example, a 2,000 IOPS volume should be smaller than 1,000 GiB.

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.

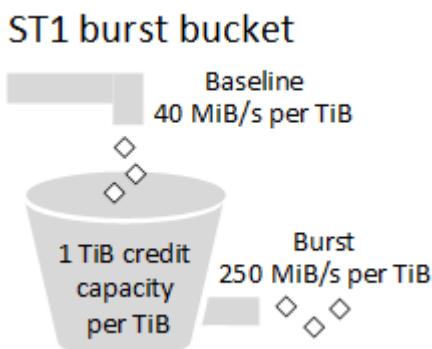
Throughput Optimized HDD (st1) volumes, though similar to Cold HDD (sc1) volumes, are designed to support *frequently* accessed data.

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 \(p. 746\)](#).

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.



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:

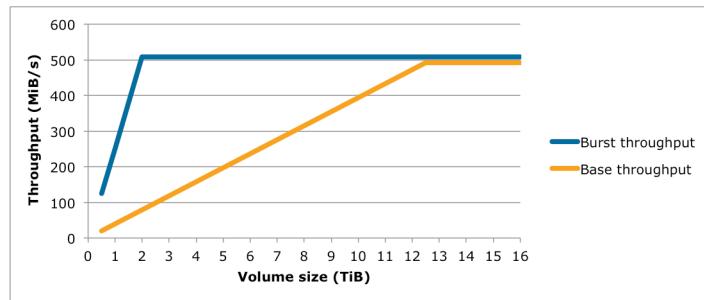
$$2 \text{ TiB} \times \frac{250 \text{ MiB/s}}{2 \text{ TiB}} = 500 \text{ MiB/s}$$

1 TiB

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
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 \(p. 747\)](#).

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 ideal 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.

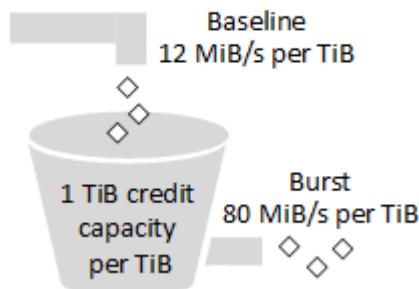
Note

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 \(p. 746\)](#).

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:

$$\frac{12 \text{ MiB/s}}{16 \text{ TiB}} \times \frac{}{1 \text{ TiB}} = 192 \text{ MiB/s}$$

Burst throughput varies from 40 MiB/s to a cap of 250 MiB/s, which is reached at 3.125 TiB as follows:

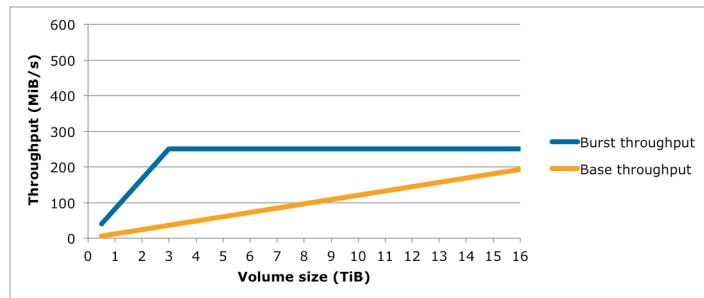
$$\frac{80 \text{ MiB/s}}{3.125 \text{ TiB}} \times \frac{}{1 \text{ TiB}} = 250 \text{ MiB/s}$$

1 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 \(p. 747\)](#).

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. 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 \(p. 747\)](#).

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:

$$\frac{\text{Volume size}}{\text{Throughput}} = \text{Scan time}$$

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.

$$\begin{aligned} \frac{5 \text{ TiB}}{500 \text{ MiB/s}} &= \frac{5 \text{ TiB}}{0.00047684 \text{ TiB/s}} = 10,486 \text{ s} = 2.91 \text{ hours (optimal)} \\ 2.91 \text{ hours} + \frac{2.91 \text{ hours}}{(0.90)(0.99)} &= 3.27 \text{ hours (minimum expected)} \end{aligned}$$

Similarly, an `sc1` customer with a 5-TiB volume can expect to complete a full volume scan in 5.83 to 6.54 hours.

$$\begin{aligned} \frac{5 \text{ TiB}}{0.000238418 \text{ TiB/s}} &= 20972 \text{ s} = 5.83 \text{ hours (optimal)} \\ 5.83 \text{ hours} + \frac{5.83 \text{ hours}}{(0.90)(0.99)} &= 6.54 \text{ hours (minimum expected)} \end{aligned}$$

(0.90)(0.99)

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
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](#).

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](#). 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](#).

Creating an Amazon EBS Volume

You can create an Amazon EBS volume that you can then attach to any EC2 instance within the same Availability Zone. You can choose to create an encrypted EBS volume, but encrypted volumes can only be attached to selected instance types. For more information, see [Supported Instance Types \(p. 820\)](#). You can use IAM policies to enforce encryption on new volumes. For more information, see the example IAM policies in [4. Working with Volumes \(p. 588\)](#) and [6: Launching Instances \(RunInstances\) \(p. 594\)](#).

You can also 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 \(p. 355\)](#) and [Block Device Mapping \(p. 866\)](#). You can restore volumes from previously created snapshots. For more information, see [Restoring an Amazon EBS Volume from a Snapshot \(p. 748\)](#).

You can apply tags to EBS volumes at the time of creation. With tagging, you can simplify tracking of your Amazon EC2 resource inventory. Tagging on creation can be combined with an IAM policy to enforce tagging on new volumes. For more information, see [Tagging Your Resources](#).

If you are creating a volume for a high-performance storage scenario, you should make sure to use a Provisioned IOPS SSD (`io1`) 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 EC2 Instance Configuration \(p. 827\)](#).

New 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 restored 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. For most applications, amortizing this cost over the lifetime of the volume is acceptable. Performance is restored after the data is accessed once. For more information, see [Initializing Amazon EBS Volumes \(p. 830\)](#).

To create an 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 \(p. 878\)](#).
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 \(p. 735\)](#).

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.

6. For **Size (GiB)**, type the size of the volume.
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) To create an encrypted volume, select the **Encrypted** box and choose the master key you want to use when encrypting the volume. You can choose the default master key for your account, or you can choose any customer master key (CMK) that you have previously created using the AWS Key Management Service. Available keys are visible in the **Master Key** menu, or you can paste the full ARN of any key that you have access to. For more information, see the [AWS Key Management Service Developer Guide](#).

Note

Encrypted volumes can only be attached to selected instance types. For more information, see [Supported Instance Types \(p. 820\)](#).

10. (Optional) Choose **Create additional tags** to add tags to the volume. For each tag, provide a tag key and a tag value.
11. Choose **Create Volume**.

To create 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 \(p. 3\)](#).

- [create-volume](#) (AWS CLI)
- [New-EC2Volume](#) (AWS Tools for Windows PowerShell)

Restoring an Amazon EBS Volume from a Snapshot

You can restore an Amazon EBS volume with data from a snapshot stored in Amazon S3. You need to know the ID of the snapshot you want to restore your volume from and you need to have access permissions for the snapshot. For more information on snapshots, see [Amazon EBS Snapshots \(p. 792\)](#).

New volumes created from existing EBS snapshots load lazily in the background. This means that after a volume is created from a snapshot, there is no need to wait for all of the data to transfer from Amazon S3 to your EBS volume before your attached instance can start accessing the 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 continues loading the rest of the data in the background.

EBS volumes that are restored from encrypted snapshots are automatically encrypted. Encrypted volumes can only be attached to selected instance types. For more information, see [Supported Instance Types \(p. 820\)](#).

Because of security constraints, you cannot directly restore an EBS volume from a shared encrypted snapshot that you do not own. You must first create a copy of the snapshot, which you will own. You can then restore a volume from that copy. For more information, see [Amazon EBS Encryption](#).

New 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 restored 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. Performance is restored after the data is accessed once.

For most applications, amortizing the initialization cost over the lifetime of the volume is acceptable. To ensure that your restored volume always functions at peak capacity in production, you can force the immediate initialization of the entire volume using **dd** or **fio**. For more information, see [Initializing Amazon EBS Volumes \(p. 830\)](#).

To restore 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 restore the snapshot to a volume in a different region, you can copy your snapshot to the new region and then restore it to a volume in that region. For more information, see [Copying an Amazon EBS Snapshot \(p. 799\)](#).

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 \(p. 735\)](#).

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.

6. For **Snapshot**, 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.

Volumes that are restored from encrypted snapshots can only be attached to instances that support Amazon EBS encryption. For more information, see [Supported Instance Types \(p. 820\)](#).

7. For **Size (GiB)**, type the size of the volume, or verify that the default size of the snapshot is adequate.

Note

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**. Any AWS Marketplace product codes from the snapshot are propagated to the volume.

8. With a Provisioned IOPS SSD volume, for **IOPS**, type the maximum number of input/output operations per second (IOPS) that the volume should support.
9. 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.
10. (Optional) Choose **Create additional tags** to add tags to the volume. For each tag, provide a tag key and a tag value.
11. Choose **Create Volume**.
12. After you've restored a volume from a snapshot, you can attach it to an instance to begin using it. For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 750\)](#).
13. If you restored a snapshot to a larger volume than the default for that snapshot, you must extend the file system on the volume to take advantage of the extra space. For more information, see [Modifying the Size, IOPS, or Type of an EBS Volume on Linux \(p. 769\)](#).

To restore 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 \(p. 3\)](#).

- [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 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 \(p. 863\)](#).
- 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 \(p. 820\)](#).
- 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 \(p. 864\)](#).
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 \(p. 751\)](#).

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)

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, and appears as a removable disk in Windows. 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 \(p. 792\)](#).

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*.

To make an EBS volume available for use on Linux

1. Connect to your instance using SSH. For more information, see [Connect to Your Linux Instance \(p. 397\)](#).
2. 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), even 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.

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.

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvdf  202:80   0 100G  0 disk
xvda1 202:1    0   8G  0 disk /
```

The output of `lsblk` removes the `/dev/` prefix from full device paths. In this example, `/dev/xvda1` is mounted as the root device (note that `MOUNTPOINT` is listed as `/`, the root of the Linux file system hierarchy), and `/dev/xvdf` is attached, but it has not been mounted yet.

EBS volumes are exposed as NVMe block devices for the following instances: C5, C5d, i3.metal, M5, M5d, R5, R5d, and z1d. The device names that you specify are renamed using NVMe device names (`/dev/nvme[0-26]n1`). For more information, see [Amazon EBS and NVMe \(p. 823\)](#).

3. Determine whether to create 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 have been restored 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 `sudo file -s device` command to list special information, such as file system type.

```
[ec2-user ~]$ sudo file -s /dev/xvdf
/dev/xvdf: data
```

If the output of the previous command shows simply `data` for the device, then there is no file system on the device and you must create one. You can go on to [Step 4 \(p. 752\)](#). If you run this command on a device that contains a file system, then your output will be different.

```
[ec2-user ~]$ sudo file -s /dev/xvda1
```

```
/dev/xvda1: Linux rev 1.0 ext4 filesystem data, UUID=1701d228-e1bd-4094-a14c-8c64d6819362 (needs journal recovery) (extents) (large files) (huge files)
```

In the previous example, the device contains Linux rev 1.0 ext4 filesystem data, so this volume does not need a file system created (you can skip [Step 4 \(p. 752\)](#) if your output shows file system data).

4. (Conditional) Use the following command to create an ext4 file system on the volume. Substitute the device name (such as /dev/xvdf) for **device_name**. Depending on the requirements of your application or the limitations of your operating system, you can choose a different file system type, such as ext3 or XFS.

Warning

This step assumes that you're mounting an empty volume. If you're mounting a volume that already has data on it (for example, a volume that was restored from a snapshot), don't use **mkfs** before mounting the volume (skip to the next step instead). Otherwise, you'll format the volume and delete the existing data.

```
[ec2-user ~]$ sudo mkfs -t ext4 device_name
```

5. Use the following 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. Substitute a location for **mount_point**, such as /data.

```
[ec2-user ~]$ sudo mkdir mount_point
```

6. Use the following command to mount the volume at the location you just created.

```
[ec2-user ~]$ sudo mount device_name mount_point
```

7. (Optional) To mount this 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 any text editor, such as **nano** or **vim**.

Note

You must open the file as root or by using the **sudo** command.

- c. Add a new line to the end of the file for your volume using the following format:

```
device_name 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 following example for them (**defaults**, **nofail 0 2**). For more information on /etc/fstab entries, see the **fstab** manual page (by entering **man fstab** on the command line).

You can use the system's current device name (/dev/sda1, /dev/xvda1, etc.) in /etc/fstab, but we recommend using the device's 128-bit universally unique identifier (UUID) instead. System-declared block-device names may change under a variety of circumstances, but the UUID is assigned to a volume partition when it is formatted and persists throughout the partition's service life. By using the UUID, you reduce the chances of the block-device mapping in /etc/fstab leaving the system unbootable after a hardware reconfiguration.

To find the UUID of a device, first display the available devices:

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

The following is example output:

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/xvda1	8123812	1876888	6146676	24%	/
devtmpfs	500712	56	500656	1%	/dev
tmpfs	509724	0	509724	0%	/dev/shm

Next, continuing this example, examine the output of either of two commands to find the UUID of /dev/xvda1:

- **sudo file -s /dev/xvda1**
- **ls -al /dev/disk/by-uuid/**

Assuming that you find /dev/xvda1 to have UUID de9a1ccd-a2dd-44f1-8be8-0123456abcdef, you would add the following entry to /etc/fstab to mount an ext4 file system at mount point /data:

```
UUID=de9a1ccd-a2dd-44f1-8be8-0123456abcdef      /data    ext4    defaults,nofail
0          2
```

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, including Ubuntu versions earlier than 16.04, must also add the `nobootwait` mount option.

- After you've added the new entry to /etc/fstab, you must 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.

- (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
```

- 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*.

Viewing Volume Information

You can view descriptive information for your Amazon EBS volumes in a selected region at a time in the AWS Management Console. You can also view detailed information about a single volume, including the 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.

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. To view more information about a volume, select it. In the details pane, you can inspect the information provided about the volume.

To view what EBS (or other) volumes are attached to an Amazon EC2 instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. To view more information about an instance, select it.
4. In the details pane, you can inspect the information provided about root and block devices.

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)

Monitoring the Status of Your Volumes

Amazon Web Services (AWS) automatically provides data, such as Amazon CloudWatch metrics and volume status checks, that you can use to monitor your Amazon Elastic Block Store (Amazon EBS) volumes.

Contents

- [Monitoring Volumes with CloudWatch \(p. 754\)](#)
- [Monitoring Volumes with Status Checks \(p. 759\)](#)
- [Monitoring Volume Events \(p. 761\)](#)
- [Working with an Impaired Volume \(p. 762\)](#)
- [Working with the AutoEnableIO Volume Attribute \(p. 765\)](#)

Monitoring Volumes with CloudWatch

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 (io1) 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) volumes automatically send one-minute metrics to CloudWatch. Data is only reported to CloudWatch when the volume is attached to an instance. For more information about how to monitor Amazon EBS, see [Monitoring the Status of Your Volumes](#) in the *Amazon EC2 User Guide for Linux Instances*.

The `AWS/EBS` namespace includes the following metrics.

Metric	Description
<code>VolumeReadBytes</code>	Provides information on the I/O operations in a specified period of time. The <code>Sum</code> statistic reports the total number of bytes transferred during the period. The <code>Average</code> statistic reports the average size of each I/O operation during the period, except on volumes attached to a C5, C5d, i3.meta1, M5, M5d, R5, R5d, and z1d instance, where the average represents the average over the specified period. The <code>SampleCount</code> statistic reports the total number of I/O operations during the period, except on volumes attached to a C5, C5d, i3.meta1, M5, M5d, R5, R5d, and z1d instance, where the sample count represents the number of data points used in the statistical calculation. Data is reported to CloudWatch only when the volume is active.
<code>VolumeWriteBytes</code>	The <code>Minimum</code> and <code>Maximum</code> statistics on this metric are supported only by volumes attached to a C5, C5d, i3.meta1, M5, M5d, R5, R5d, and z1d instance. Units: Bytes
<code>VolumeReadOps</code>	The total number of I/O operations in a specified period of time.
<code>VolumeWriteOps</code>	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.

Metric	Description
	<p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to a C5, C5d, i3 .metal, M5, M5d, R5, R5d, and z1d instance.</p> <p>Units: Count</p>
<code>VolumeTotalReadTime</code> <code>VolumeTotalWriteTime</code>	<p>The total number of seconds spent by all 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.</p> <p>The Average statistic on this metric is not relevant for volumes attached to a C5, C5d, i3 .metal, M5, M5d, R5, R5d, and z1d instance.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to a C5, C5d, i3 .metal, M5, M5d, R5, R5d, and z1d instance.</p> <p>Units: Seconds</p>
<code>VolumeIdleTime</code>	<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 a C5, C5d, i3 .metal, M5, M5d, R5, R5d, and z1d instance.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to a C5, C5d, i3 .metal, M5, M5d, R5, R5d, and z1d instance.</p> <p>Units: Seconds</p>
<code>VolumeQueueLength</code>	<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 a C5, C5d, i3 .metal, M5, M5d, R5, R5d, and z1d instance.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to a C5, C5d, i3 .metal, M5, M5d, R5, R5d, and z1d instance.</p> <p>Units: Count</p>

Metric	Description
VolumeThroughputPercentage	<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 within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.</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>
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 Sum statistic on this metric is not relevant for volumes attached to a C5, C5d, i3 .metal, M5, M5d, R5, R5d, and z1d instance.</p> <p>Units: Percent</p>

Dimensions for Amazon EBS Metrics

The only dimension that Amazon EBS sends to CloudWatch is the volume ID. This means that all available statistics are filtered by volume ID.

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)	$\text{Sum}(\text{VolumeWriteBytes}) / \text{Period} / 1024$

Graph	Description using raw metrics
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>Note For C5, C5d, i3.metal, M5, M5d, R5, R5d, and z1d instances, the following formula derives Average Read Size using CloudWatch Metric Math: $(\text{Sum}(\text{VolumeReadBytes}) / \text{Sum}(\text{VolumeReadOps})) / 1024$ 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>Note For C5, C5d, i3.metal, M5, M5d, R5, R5d, and z1d instances, the following formula derives Average Write Size using CloudWatch Metric Math: $(\text{Sum}(\text{VolumeWriteBytes}) / \text{Sum}(\text{VolumeWriteOps})) / 1024$ 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>Note For C5, C5d, i3.metal, M5, M5d, R5, R5d, and z1d instances, the following formula derives Average Read Latency using CloudWatch Metric Math: $(\text{Sum}(\text{VolumeTotalReadTime}) / \text{Sum}(\text{VolumeReadOps})) \times 1000$ 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>Note For C5, C5d, i3.metal, M5, M5d, R5, R5d, and z1d instances, the following formula derives Average Write Latency using CloudWatch Metric Math: $(\text{Sum}(\text{VolumeTotalWriteTime}) / \text{Sum}(\text{VolumeWriteOps})) \times 1000$ 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.

Monitoring Volumes with 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.)

If the consistency of a particular volume is not a concern for you, 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 `AutoEnableIO` volume attribute, 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 `io1` volumes that are attached to an instance and 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 a `io1` volume to an instance for this check to report the I/O performance status.

Important

While initializing `io1` 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` volumes while you are initializing them. For more information, see [Initializing Amazon EBS Volumes \(p. 830\)](#).

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)
<code>ok</code>	Enabled (I/O Enabled or I/O Auto-Enabled)	Normal (Volume performance is as expected)
<code>warning</code>	Enabled (I/O Enabled or I/O Auto-Enabled)	Degraded (Volume performance is below expectations)

Volume status	I/O enabled status	I/O performance status (only available for Provisioned IOPS volumes)
		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	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**.
3. On the **EBS Volumes** page, use the **Volume Status** column lists the operational status of each volume.
4. To view an individual volume's status, select the volume, and choose **Status Checks**.

The screenshot shows the Amazon EC2 console with the following details:

- Volumes:** vol-d882c69b
- Status Checks:** A yellow warning icon indicates that IO operations have been disabled since 16 hours and 58 minutes ago. The message reads: "IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistencies may occur." Below the icon are tabs for Description, Status Checks (which is selected), Monitoring, and Tags.
- Volume Status:** impaired
- IO Status:** Disabled
- Since:** December 23, 2013 7:06:41 PM UTC+2
- Description:** Awaiting Action: Enable IO
- Auto-Enabled IO:** Disabled [Edit](#)
- Find out more:** A link to learn about working with volume status checks and events.
- If you need technical assistance:** A link to post issues to the Developer Forums.

5. If you have a volume with a failed status check (status is **impaired**), see [Working with an Impaired Volume \(p. 762\)](#).

Alternatively, you can use the **Events** pane to view all events for your instances and volumes in a single pane. For more information, see [Monitoring Volume Events \(p. 761\)](#).

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)

Monitoring 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 `AutoEnableIO` volume attribute. For more information about changing this attribute, see [Working with an Impaired Volume \(p. 762\)](#).

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` volumes only. Volume performance is as expected.

Degraded

For `io1` volumes only. Volume performance is below expectations.

Severely Degraded

For `io1` volumes only. Volume performance is well below expectations.

Stalled

For `io1` 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**.

3. All instances and volumes that have events are listed. 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.

Event: vol-3682c675

A IO operations have been disabled since 30 days, 15 hours and 22 minutes ago. Data inconsistency may occur.

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 \(p. 762\)](#). 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 (e.g., 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

This section discusses your options if a volume is impaired because the volume's data is potentially inconsistent.

Options

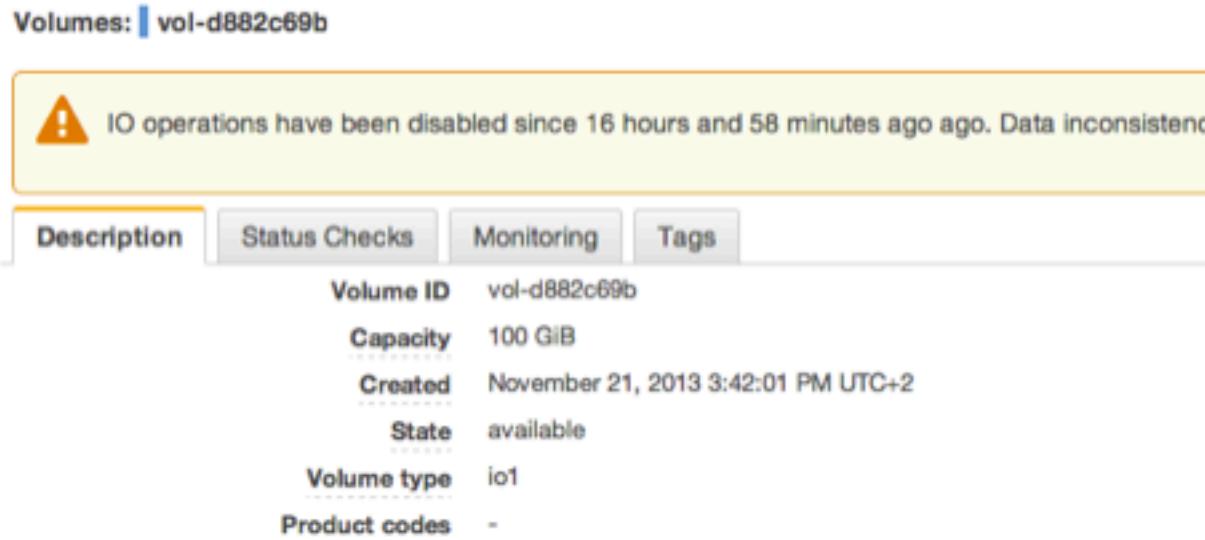
- [Option 1: Perform a Consistency Check on the Volume Attached to its Instance \(p. 763\)](#)
- [Option 2: Perform a Consistency Check on the Volume Using Another Instance \(p. 764\)](#)
- [Option 3: Delete the Volume If You No Longer Need It \(p. 765\)](#)

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**.



- e. In **Enable Volume IO**, choose **Yes, Enable**.
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 support. Choose **Troubleshoot**, and then on 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.

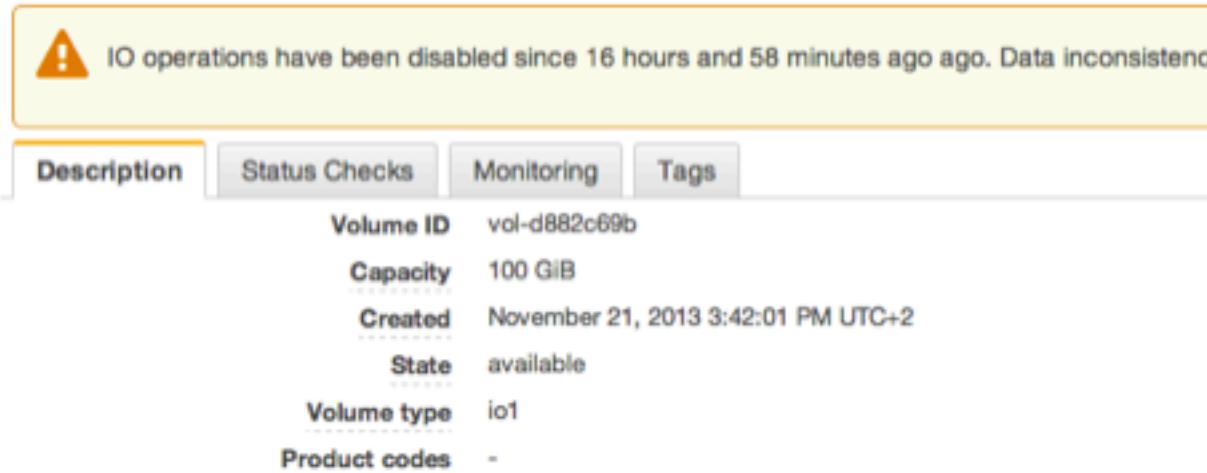
Important

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**.

Volumes: vol-d882c69b



- d. In the **Enable Volume IO** dialog box, choose **Yes, Enable**.
4. Attach the volume to another instance. For information, see [Launch Your Instance \(p. 354\)](#) and [Attaching an Amazon EBS Volume to an Instance \(p. 750\)](#).
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 support. 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 \(p. 769\)](#).

If you have a recent snapshot that backs up the data on the volume, you can create a new volume from the snapshot. For information about creating a volume from a snapshot, see [Restoring an Amazon EBS Volume from a Snapshot \(p. 748\)](#).

Working with the AutoEnableIO 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 `AutoEnableIO` volume attribute, 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 [Monitoring Volume Events \(p. 761\)](#).

This section explains how to view and modify the `AutoEnableIO` attribute of a volume using the Amazon EC2 console, the command line interface, or the API.

To view the AutoEnableIO 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.
4. In the lower pane, choose **Status Checks**.
5. In the **Status Checks** tab, **Auto-Enable IO** displays the current setting for your volume, either **Enabled** or **Disabled**.

Volumes: vol-d882c69b

A IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistencies may occur.

Description Status Checks Monitoring Tags

Volume Status Impaired

IO Status Disabled

Since December 23, 2013 7:06:41 PM UTC+2

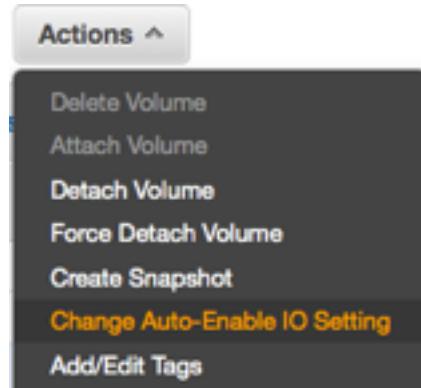
Description Awaiting Action: Enable IO

Auto-Enabled IO Disabled [Edit](#)

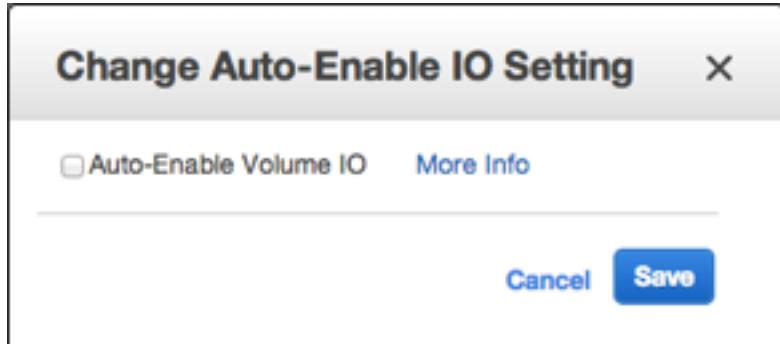
[Find out more](#) about working with volume status checks and events.
If you need technical assistance with your volume, post your issue to the [Developer Forums](#) or visit our [support forums](#).

To modify the AutoEnableIO 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.
4. At the top of the **Volumes** page, choose **Actions**.
5. Choose **Change Auto-Enable IO Setting**.



6. In the **Change Auto-Enable IO Setting** dialog box, select the **Auto-Enable Volume IO** option to automatically enable I/O for an impaired volume. To disable the feature, clear the option.



7. Choose **Save**.

Alternatively, instead of completing steps 4-6 in the previous procedure, choose **Status Checks, Edit**.

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 an 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 \(p. 769\)](#).

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 Volume Information \(p. 754\)](#).

You can reattach a volume that you detached (without unmounting it), but it might not get the same mount point and the data on the volume might be out of sync if there were writes to the volume in progress when it was detached.

To detach an EBS volume using the console

1. 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

You can use one of the following commands. 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.

To delete a volume, it must be in the available state (not attached to an instance). For more information, see [Detaching an Amazon EBS Volume from an Instance \(p. 767\)](#).

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**.
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)

Modifying the Size, IOPS, or Type of an EBS Volume on Linux

If your current-generation Amazon EBS volume is attached to a current-generation EC2 instance type, you can increase its size, change its volume type, or (for an io1 volume) adjust its IOPS performance, all without detaching it. You can apply these changes to detached volumes as well. For more information about the current generation instance types, see [Current Generation Instances](#).

The following previous-generation instance types support modification of EBS volumes without detachment: C1, C3, CC2, CR1, G2, I2, M1, M3, R3. You can safely ignore any warnings that occur with these instance types. If you are using an unsupported previous-generation instance type, or if you encounter an error while attempting a volume modification, follow the procedures in [Appendix: Starting and Stopping an Instance to Modify an EBS Volume \(p. 791\)](#).

In general, use the following steps when modifying a volume:

1. **Issue the modification command.** For more information, see [Modifying an EBS Volume from the Console \(p. 773\)](#) and [Modifying an EBS Volume from the Command Line \(p. 773\)](#).
2. **Monitor the progress of the modification.** For more information, see [Monitoring the Progress of Volume Modifications \(p. 774\)](#).
3. **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 the Volume \(p. 787\)](#).

Additionally, you can use [Amazon CloudWatch Events](#) and [AWS CloudFormation](#) to automate the actions associated with volume modification.

There is no charge to modify the configuration of a volume. You are charged at the new volume configuration price after a modification starts. For more information, see the *Amazon Elastic Block Store* section on the [Amazon EBS Pricing](#) page.

Important

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 information about EBS snapshots, see [Creating an Amazon EBS Snapshot](#).

Contents

- [Constraints on Modifying EBS Volume Size \(p. 770\)](#)
- [Modifying an EBS Volume from the Console \(p. 773\)](#)
- [Modifying an EBS Volume from the Command Line \(p. 773\)](#)
- [Monitoring the Progress of Volume Modifications \(p. 774\)](#)
- [Expanding a Linux Partition \(p. 778\)](#)
- [Extending a Linux File System after Resizing the Volume \(p. 787\)](#)
- [Limitations When Modifying EBS Volumes \(p. 790\)](#)
- [Appendix: Starting and Stopping an Instance to Modify an EBS Volume \(p. 791\)](#)

Constraints on Modifying EBS Volume Size

Modifications to the size of an Amazon EBS volume are constrained by the physics and arithmetic of block data storage, as well as by the implementation decisions of operating system and file system designers. AWS imposes additional limits on volume size to safeguard the reliability of its services.

As a service, 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 completely independent of each other. When modifying volume size, be aware of the capabilities and limits of both. For example, you can increase the size of an EBS volume to as much 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.

This section describes the most important factors that limit the usable size of an EBS volume.

AWS Service Limitations

Amazon EBS currently supports a maximum volume size of 16 TiB.

Amazon EC2 requires Windows boot volumes to use MBR partitioning. As discussed in [Partitioning Schemes \(p. 770\)](#), this means that boot volumes cannot be bigger than 2 TiB. Windows data volumes are not subject to this limitation and may be GPT-partitioned.

Linux boot volumes may be either MBR or GPT, and Linux GPT boot volumes are not subject to the 2-TiB limit.

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 \(p. 771\)](#). Two partitioning

schemes are in common use on Linux and Windows systems: master boot record (MBR) and GUID partition table (GPT). The important differences between the two 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:

$$(2^{64} - 1) \times 4,096 \text{ bytes} = 8 \text{ ZiB} - 4,096 \text{ bytes} = 8 \text{ billion TiB} - 4,096 \text{ bytes}$$

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 and Resulting Volume Capacity

Block size	Max. volume size
4 KiB (default)	16 TiB
8 KiB	32 TiB
16 KiB	64 TiB
32 KiB	128 TiB

Block size	Max. volume size
64 KiB (maximum)	256 TiB

The EBS-imposed limit on volume size is currently equal to the maximum size enabled by 4-KiB data blocks.

Summary

The following table summarizes the theoretical and implemented storage capacities for the most commonly used file systems on Amazon EBS.

MBR vs. GPT volume sizes for popular file systems, assuming 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}	$8 \text{ ZiB} = 8 \times 1024^3 \text{ TiB}$	$1 \text{ EiB} = 1024^2 \text{ TiB}$ (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>

Recommendations 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 that is 2 TiB or larger, the AMI you are using may be limited to a 2 TiB GiB boot volume size. Non-boot volumes do not have this limitation on Linux instances. For recommendations for Windows volumes, see [Recommendations 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
```

Modifying an EBS Volume from the Console

The following procedure shows how to apply available volume modifications from the Amazon EC2 console.

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 (IO1)** as your volume type, enter an allowed integer value for **IOPS**.
4. After you have specified all of the modifications to apply, choose **Modify, 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 the Volume \(p. 787\)](#).

Modifying an EBS Volume from the Command Line

The following example demonstrates how an EBS volume can be modified from the command line using the AWS CLI. Depending on your default configuration, you may need to specify information such as region and Availability Zone. The ID of the source volume being modified is required, and you must have appropriate permissions to carry out the action. When an `io1` volume is the modification target, you must specify its level of provisioned IOPS. Multiple modification actions (to change capacity, IOPS, or type) may be performed in a single command.

For example, an EBS volume is configured as follows:

- Volume ID: `vol-1111111111111111`
- Volume size: 100 GiB
- Volume type: `gp2`

You can change the volume configuration to the following:

- Volume size: 200 GiB
- Volume type: `io1`

- Provisioning level: 10,000 IOPS

Apply the above modifications with the following command:

```
aws ec2 modify-volume --region us-east-1 --volume-id vol-1111111111111111 --size 200 --volume-type io1 --iops 10000
```

The command yields output similar to the following:

```
{  
    "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  
    }  
}
```

Note

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 the Volume \(p. 787\)](#).

Monitoring the Progress of Volume Modifications

An EBS volume being modified goes through a sequence of states. After you issue a `ModifyVolume` directive, whether from the console, CLI, API, or SDK, the volume enters first the `Modifying` state, then the `Optimizing` state, and finally the `Complete` state. At this point, the volume is ready to be further modified. Rarely, a transient AWS fault can result in the `Failed` state. If this occurs, retry the modification.

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 may 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.

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.

You can monitor the progress of a modification by inspecting the AWS Management Console, by querying the volume's state with the Amazon EC2 API/CLI, or by accessing metrics sent to Amazon CloudWatch Events. The following procedures demonstrate these approaches.

To monitor progress of a modification from the console

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

2. Choose **Volumes**, and select the volume to inspect. The volume's status is displayed in the **State** column. In the example below, the modification state is **completed**. This state information is also displayed in the **State** field of the details pane.
3. Open the information icon next to the **State** field to display complete before and after information about the most recent modification action, as illustrated below.

Create Volume Actions ▾

Filter by tags and attributes or search by keyword

Volume ID	Size	Volume Type	IOPS	Sn
vol-065fc28c...	1000 GiB	gp2	3000	

Volumes: vol-065fc28c...

Description Status Checks Monitoring Tags

Volume ID	vol-065fc28c...
Size	1000 GiB
Created	January 25, 2017 at 4:26:36 PM UTC-8
State	available - completed (100%)
Attachment information	
Volume type	gp2
Product codes	-
IOPS	3000

Example To monitor progress of a modification from the command line

Use [describe-volumes-modifications \(p. 773\)](#) to view the progress of the modifications. In this example, volume `vol-1111111111111111` from above and another volume, `vol-2222222222222222`, are called.

```
aws ec2 describe-volumes-modifications --region us-east-1 --volume-id vol-1111111111111111 vol-2222222222222222
```

The command returns one or more `VolumesModification` objects. The following is example output. The first object is nearly identical to the original `modify-volume` command output shown above. No additional modifications have been applied, however.

```
{
    "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 queries for all volumes in a region 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" --region us-east-1 --query "VolumesModifications[?StartTime>='2017-02-01'].{ID:VolumeId,STATE:ModificationState}"
```

In this case the query returns information about two volumes:

```
[
    {
        "STATE": "optimizing",
        "ID": "vol-06397e7a0eEXAMPLE"
    },
    {
        "STATE": "completed",
        "ID": "vol-bEXAMPLE"
    }
]
```

]

To monitor progress of a modification with CloudWatch Events

With CloudWatch Events, you can create a notification rule for volume modification events to send a text message or execute a Lambda function.

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 code:

```
{  
    "source": [  
        "aws.ec2"  
    ],  
    "detail-type": [  
        "EBS Volume Notification"  
    ],  
    "detail": {  
        "event": [  
            "modifyVolume"  
        ]  
    }  
}
```

Choose **Save**.

The typical event output should look like the following:

```
Body:  
{  
    "version": "0",  
    "id": "1ea2ace2-7790-46ed-99ab-d07a8bd68685",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
    "account": "065441870323",  
    "time": "2017-01-12T21:09:07Z",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-east-1:065441870323:volume/vol-03a55cf56513fa1b6"  
    ],  
    "detail": {  
        "result": "optimizing",  
        "cause": "",  
        "event": "modifyVolume",  
        "request-id": "auto-58c08bad-d90b-11e6-a309-b51ed35473f8"  
    }  
}
```

You can use your rule to generate a notification message with [Amazon SNS](#) or to invoke a [Lambda function](#) in response to matching events.

Expanding a Linux Partition

Some Amazon EC2 root volumes and volumes that are restored from snapshots contain a partition that actually holds the file system and the data. If you think of a volume as a container, a partition is another container inside the volume, and the data resides on the partition. Growing the volume size does not

grow the partition; to take advantage of a larger volume, the partition must be expanded to the new size.

Note

Not all volumes restored from snapshots are partitioned, and this procedure may not apply to your volume. You may just need to resize the file system on your volume to make all of the space available. If you are not sure if your volume has a partition that needs resizing, see [To check if your volume partition needs resizing \(p. 788\)](#) for more information.

If the partition you want to expand is not the root partition, then you can simply unmount it and resize the partition from the instance itself. If the partition to resize is the root partition for an instance, the process becomes more complicated because you cannot unmount the root partition of a running instance. You have to perform the following procedures on another instance, which is referred to as a *secondary instance*.

Important

The following procedures were written for and tested on Amazon Linux. Other distributions with different tool sets and tool versions may behave differently.

Topics

- [Preparing a Linux Root Partition for Expansion \(p. 779\)](#)
- [Expanding a Linux Partition Using parted \(p. 780\)](#)
- [Expanding a Linux Partition Using gdisk \(p. 783\)](#)
- [Returning an Expanded Partition to Its Original Instance \(p. 787\)](#)

Preparing a Linux Root Partition for Expansion

There are several steps to take to expand the root partition of an instance. If the partition to expand is not the root partition, then this procedure is not necessary.

To prepare a Linux root partition for expansion

1. If your primary instance is running, stop it. You cannot perform the rest of this procedure on a running instance. For more information, see [Stop and Start Your Instance \(p. 415\)](#).
2. Check the integrity of your volume. File-system corruption that is picked up by the snapshot may render a restored root volume unbootable.
3. Take a snapshot of your volume. It can be easy to corrupt or lose your data in the following procedures. If you have a fresh snapshot, you can always start over in case of a mistake and your data is still safe. For more information, see [Creating an Amazon EBS Snapshot \(p. 795\)](#).
4. Record the device name that the volume is attached to. You can find this information on the **Root device** field of the instance's details pane. The value is likely /dev/sda1 or /dev/xvda.
5. Detach the volume from the primary instance. For more information, see [Detaching an Amazon EBS Volume from an Instance \(p. 767\)](#).
6. Attach the volume to another (secondary) instance in the same Availability Zone. For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 750\)](#). If your EBS volume is encrypted, you must use a secondary instance that supports Amazon EBS encryption; otherwise, you can use a t2.micro instance for this procedure. For more information, see [Supported Instance Types \(p. 820\)](#). If you do not already have a secondary instance, you must launch one. For more information, see [Launching an Instance Using the Launch Instance Wizard \(p. 355\)](#).

Important

The secondary instance must be running when you attach the volume, and you should not reboot the secondary instance while multiple root volumes are attached; booting an instance with multiple root volumes attached could cause the instance to boot to the wrong volume.

7. Log in to the secondary instance with SSH. For more information, see [Connect to Your Linux Instance \(p. 397\)](#). Continue with the next procedure.

Expanding a Linux Partition Using **parted**

The **parted** utility is a partition editing tool that is available on most Linux distributions. It can create and edit both MBR partition tables and GPT partition tables. Some versions of **parted** (newer than version 2.1) have limited support for GPT partition tables and they may cause boot issues if their version of **parted** is used to modify boot volumes. You can check your version of **parted** with the **parted --version** command.

If you are expanding a partition that resides on a GPT partitioned device, you should choose to use the **gdisk** utility instead. If you're not sure which disk label type your volume uses, you can check it with the **sudo fdisk -l** command. For more information, see [To expand a Linux partition using **gdisk** \(p. 784\)](#).

To expand a Linux partition using **parted**

If the partition to expand is the root partition, be sure to follow the steps in [To prepare a Linux root partition for expansion \(p. 779\)](#) first.

1. Identify the device that contains the partition that you want to expand. Use the **lsblk** command to list all devices and partitions attached to the instance.

```
[ec2-user ~]$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvdf      202:80   0 100G  0 disk
##xvdf1  202:81   0   8G  0 part /mnt
xvda1    202:1    0   30G  0 disk /
```

In this example, the **xvdf** device has 100 GiB of available storage and it contains an 8-GiB partition.

2. Unmount the partition if it is mounted. Run the **umount** command with the value of **MOUNTPOINT** from the **lsblk** command. In this example, the **MOUNTPOINT** value for the partition is **/mnt**.

```
[ec2-user ~]$ sudo umount /mnt
```

3. Take a snapshot of your volume (unless you just took one in the previous procedure). It can be easy to corrupt or lose your data in the following procedures. If you have a fresh snapshot, you can always start over in case of a mistake and your data is still safe. For more information, see [Creating an Amazon EBS Snapshot \(p. 795\)](#).
4. Run the **parted** command on the device (and not the partition on the device). Remember to add the **/dev/** prefix to the name that **lsblk** outputs.

```
[ec2-user ~]$ sudo parted /dev/xvdf
GNU Parted 2.1
Using /dev/xvdf
Welcome to GNU Parted! Type 'help' to view a list of commands.
```

5. Change the **parted** units of measure to sectors.

```
(parted) unit s
```

6. Run the **print** command to list the partitions on the device. For certain partition table types, you might be prompted to repair the partition table for the larger volume size. Answer 'Ignore' to any questions about fixing the existing partition table; you create a new table later.

```
(parted) print
```

- a. If you receive the following message, enter 'Ignore' to prevent the backup GPT location from changing.

```
Error: The backup GPT table is not at the end of the disk, as it should be. This
might mean that another operating
system believes the disk is smaller. Fix, by moving the backup to the end (and
removing the old backup)?
Fix/Ignore/Cancel? Ignore
```

- b. If you receive the following message, enter 'Ignore' again to keep the space on the drive the same.

```
Warning: Not all of the space available to /dev/xvdf appears to be used, you can
fix the GPT to use all of the
space (an extra 46137344 blocks) or continue with the current setting?
Fix/Ignore? Ignore
```

7. Examine the output for the total size of the disk, the partition table type, the number of the partition, the start point of the partition, and any flags, such as boot. For gpt partition tables, note the name of the partition; for msdos partition tables, note the Type field (primary or extended). These values are used in the upcoming steps.

The following is a gpt partition table example.

```
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdf: 209715200s
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size      File system  Name           Flags
128    2048s  4095s   2048s    ext4          BIOS Boot Partition bios_grub
1      4096s 16777182s 16773087s  ext4          Linux
```

The following is an msdos partition table example.

```
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdd: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End     Size      Type      File system  Flags
1       2048s 35649535s 35647488s primary  ext3
```

8. Delete the partition entry for the partition using the number (1) from the previous step.

```
(parted) rm 1
```

9. Create a new partition that extends to the end of the volume.

(For the gpt partition table example) Note the start point and name of partition 1 above. For the gpt example, there is a start point of 4096s, and the name Linux. Run the **mkpart** command with the start point of partition 1, the name, and 100% to use all of the available space.

```
(parted) mkpart Linux 4096s 100%
```

(For the msdos partition table example) Note the start point and the partition type of partition 1 above. For the msdos example, there is a start point of 2048s and a partition type of primary. Run the **mkpart** command with a primary partition type, the start point of partition 1, and 100% to use all of the available space.

```
(parted) mkpart primary 2048s 100%
```

- Run the **print** command again to verify your partition.

(For the gpt partition table example)

```
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdf: 209715200s
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size    File system  Name          Flags
128    2048s  4095s   2048s   BIOS Boot Partition bios_grub
      1        4096s 209713151s 209709056s   ext4          Linux
```

(For the msdos partition table example)

```
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdd: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End     Size    Type      File system  Flags
1        2048s 104857599s 104855552s  primary    ext3
```

- Check to see that any flags that were present earlier are still present for the partition that you expanded. In some cases, the boot flag may be lost. If a flag was dropped from the partition when it was expanded, add the flag with the following command, substituting your partition number and the flag name. For example, the following command adds the boot flag to partition 1.

```
(parted) set 1 boot on
```

You can run the **print** command again to verify your change.

- Run the **quit** command to exit **parted**.

```
(parted) quit
```

Note

Because you removed a partition and added a partition, **parted** may warn that you may need to update `/etc/fstab`. This is only required if the partition number changes.

- Check the file system to make sure that there are no errors (this is required before you may extend the file system). Note the file system type from the previous **print** commands. Choose one of the commands below based on your file system type; if you are using a different file system, consult the documentation for that file system to determine the correct check command.

(For ext3 or ext4 file systems)

```
[ec2-user ~]$ sudo e2fsck -f /dev/xvdf1
e2fsck 1.42.3 (14-May-2012)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
```

```
/: 31568/524288 files (0.4% non-contiguous), 266685/2096635 blocks
```

(For xfs file systems)

```
[ec2-user ~]$ sudo xfs_repair /dev/xvdf1
Phase 1 - find and verify superblock...
Phase 2 - using internal log
    - zero log...
    - scan filesystem freespace and inode maps...
    - found root inode chunk
Phase 3 - for each AG...
    - scan and clear agi unlinked lists...
    - process known inodes and perform inode discovery...
    - agno = 0
    - agno = 1
    - agno = 2
    - agno = 3
    - process newly discovered inodes...
Phase 4 - check for duplicate blocks...
    - setting up duplicate extent list...
    - check for inodes claiming duplicate blocks...
    - agno = 0
    - agno = 1
    - agno = 2
    - agno = 3
Phase 5 - rebuild AG headers and trees...
    - reset superblock...
Phase 6 - check inode connectivity...
    - resetting contents of realtime bitmap and summary inodes
    - traversing filesystem ...
    - traversal finished ...
    - moving disconnected inodes to lost+found ...
Phase 7 - verify and correct link counts...
done
```

14. The next steps differ depending on whether the expanded partition belongs on the current instance or if it is the root partition for another instance.
 - If this partition belongs on the current instance, remount the partition at the MOUNTPOINT identified in [Step 2 \(p. 780\)](#).

```
[ec2-user ~]$ sudo mount /dev/xvdf1 /mnt
```

After you have mounted the partition, extend the file system to use the newly available space by following the procedures in [Extending a Linux File System after Resizing the Volume \(p. 787\)](#).

- If this volume is the root partition for another instance, proceed to the procedures in [Returning an Expanded Partition to Its Original Instance \(p. 787\)](#).

Expanding a Linux Partition Using gdisk

The **gdisk** utility (sometimes called GPT fdisk) is a text-based, menu-driven tool for creating and editing partition tables, and it has better support for GPT partition tables than **parted** in some distributions. Many common Linux distributions (such as Amazon Linux and Ubuntu) provide **gdisk** by default. If your distribution does not provide the **gdisk** command, you can find out how to get it by visiting [Obtaining GPT fdisk](#); in many cases, it is much easier to launch an Amazon Linux instance to use as a secondary instance because the **gdisk** command is already available.

To expand a Linux partition using gdisk

If the partition to expand is the root partition, be sure to follow the steps in [To prepare a Linux root partition for expansion \(p. 779\)](#) first.

1. Identify the device that contains the partition that you want to expand. Use the **lsblk** command to list all devices and partitions attached to the instance.

```
[ec2-user ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO MOUNTPOINT
xvdf    202:80   0 100G  0
xvdf1   202:81   0  9.9G  0 /mnt
xvda1   202:1    0   30G  0 /
```

In this example, the **xvdf** device has 100 GiB of available storage and it contains a 9.9-GiB partition.

2. Unmount the partition if it is mounted. Run the **umount** command with the value of **MOUNTPOINT** from the **lsblk** command. In this example, the **MOUNTPOINT** value for the partition is **/mnt**.

```
[ec2-user ~]$ sudo umount /mnt
```

3. Take a snapshot of your volume (unless you just took one in the previous procedure). It can be easy to corrupt or lose your data in the following procedures. If you have a fresh snapshot, you can always start over in case of a mistake and your data is still safe. For more information, see [Creating an Amazon EBS Snapshot \(p. 795\)](#).
4. Run the **gdisk** command on the device (and not the partition on the device). Remember to add the **/dev/** prefix to the name that **lsblk** outputs.

```
[ec2-user ~]$ sudo gdisk /dev/xvdf
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.
```

5. Run the **p** command to print the partition table for the device.
6. Examine the output for the disk identifier, partition number, starting sector, code for the partition, and name of the partition. If your volume has multiple partitions, take note of each one.

```
Command (? for help): p
Disk /dev/xvdf: 209715200 sectors, 100.0 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): 947F4655-F3BF-4A1F-8203-000000000000
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 20705246
Partitions will be aligned on 2048-sector boundaries
Total free space is 2108 sectors (1.0 MiB)

Number  Start (sector)   End (sector)   Size            Code  Name
      1              2048           20705152   9.9 GiB       EF00  lxroot
```

In the preceding example, the disk identifier is **947F4655-F3BF-4A1F-8203-000000000000**, the partition number is **1**, the starting sector is **2048**, the code is **EF00**, and the name is **lxroot**. Your values will vary.

7. Because the existing partition table was originally created for a smaller volume, create a new partition table for the larger volume. Run the **o** command to create a new, empty partition table.

```
Command (? for help): o
This option deletes all partitions and creates a new protective MBR.
Proceed? (Y/N): Y
```

8. Use the **n** command to create a new partition entry for each partition on the device.
 - If your volume has only one partition, at each prompt, enter the values that you recorded earlier. For the last sector value, use the default value to expand to the entire volume size.

```
Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-209715166, default = 2048) or {+-}size{KMGTP}: 2048
Last sector (2048-209715166, default = 209715166) or {+-}size{KMGTP}: 209715166
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): EF00
Changed type of partition to 'EFI System'
```

- If your volume has more than one partition, there is likely a BIOS boot partition, and a main data partition. Create a new partition entry for the BIOS boot partition using the values that you recorded earlier. Create another new partition entry for the main data partition using the values that you recorded earlier. For the last sector value, use the default value to expand to the entire volume size.

```
Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-209715166, default = 2048) or {+-}size{KMGTP}: 2048
Last sector (2048-209715166, default = 209715166) or {+-}size{KMGTP}: 4095
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): EF02
Changed type of partition to 'BIOS boot partition'

Command (? for help): n
Partition number (2-128, default 2): 2
First sector (34-209715166, default = 4096) or {+-}size{KMGTP}: 4096
Last sector (4096-209715166, default = 209715166) or {+-}size{KMGTP}: 209715166
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): 0700
Changed type of partition to 'Microsoft basic data'
```

9. Use the **c** command to change the name of each partition to the name of the previous partition. If your partition did not have a name, simply type **Enter**.

```
Command (? for help): c
Using 1
Enter name: lxroot
```

10. Use the **x** command to enter the expert command menu.

11. Use the **g** command to change the disk identifier to the original value.

```
Expert command (? for help): g
Enter the disk's unique GUID ('R' to randomize): 947F4655-F3BF-4A1F-8203-000000000000
The new disk GUID is 947F4655-F3BF-4A1F-8203-000000000000
```

12. Use the **w** command to write the changes to the device and exit.

```
Expert command (? for help): w
Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
```

PARTITIONS!!

```
Do you want to proceed? (Y/N): Y
OK; writing new GUID partition table (GPT) to /dev/xvdf.
The operation has completed successfully.
```

13. Check the file system to make sure that there are no errors (this is required before you may extend the file system).

- a. Find the file system type with the following command, substituting the partition you just expanded (this may be `/dev/xvdf2` if your volume had multiple partitions).

```
[ec2-user ~]$ sudo file -sL /dev/xvdf1
```

- b. Choose one of the commands below based on your file system type. If you are using a different file system, consult the documentation for that file system to determine the correct check command.

(For ext3 or ext4 file systems)

```
[ec2-user ~]$ sudo e2fsck -f /dev/xvdf1
e2fsck 1.42.3 (14-May-2012)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/: 31568/524288 files (0.4% non-contiguous), 266685/2096635 blocks
```

(For xfs file systems)

Note

You may need to install the `xfsprogs` package to work with XFS file systems. Use the following command to add XFS support to your Amazon Linux instance.

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

```
[ec2-user ~]$ sudo xfs_repair /dev/xvdf1
Phase 1 - find and verify superblock...
Phase 2 - using internal log
        - zero log...
        - scan filesystem freespace and inode maps...
        - found root inode chunk
Phase 3 - for each AG...
        - scan and clear agi unlinked lists...
        - process known inodes and perform inode discovery...
        - agno = 0
        - agno = 1
        - agno = 2
        - agno = 3
        - process newly discovered inodes...
Phase 4 - check for duplicate blocks...
        - setting up duplicate extent list...
        - check for inodes claiming duplicate blocks...
        - agno = 0
        - agno = 1
        - agno = 2
        - agno = 3
Phase 5 - rebuild AG headers and trees...
        - reset superblock...
Phase 6 - check inode connectivity...
```

```
- resetting contents of realtime bitmap and summary inodes
- traversing filesystem ...
- traversal finished ...
- moving disconnected inodes to lost+found ...
Phase 7 - verify and correct link counts...
done
```

14. The next steps differ depending on whether the expanded partition belongs on the current instance or if it is the root partition for another instance.
 - If this partition belongs on the current instance, remount the partition at the MOUNTPOINT identified in [Step 2 \(p. 784\)](#).

```
[ec2-user ~]$ sudo mount /dev/xvdf1 /mnt
```

After you have mounted the partition, extend the file system to use the newly available space by following the procedures in [Extending a Linux File System after Resizing the Volume \(p. 787\)](#).

- If this volume is the root partition for another instance, proceed to the procedures in [Returning an Expanded Partition to Its Original Instance \(p. 787\)](#).

Returning an Expanded Partition to Its Original Instance

If you expanded a root partition from another instance, follow this procedure to return the volume to its original instance.

To return an expanded root partition to its original instance

1. Detach the expanded partition from its secondary instance. For more information, see [Detaching an Amazon EBS Volume from an Instance \(p. 767\)](#).
2. Reattach the volume to the primary instance using the device name that you identified in [Step 4 \(p. 779\)](#) of the preparation procedure (p. 779). For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 750\)](#).
3. Start the primary instance. For more information, see [Stop and Start Your Instance \(p. 415\)](#).
4. (Optional) If you launched a secondary instance for the sole purpose of expanding the partition, you can terminate the instance to stop incurring charges. For more information, see [Terminate Your Instance \(p. 421\)](#).
5. Connect to your primary instance and extend the file system to use the newly available space by following the procedures in [Extending a Linux File System after Resizing the Volume \(p. 787\)](#).

After you are finished with this expanding the file system, you can create an AMI from the instance that you can use to launch new instances with the desired partition size. For more information, see [Amazon Machine Images \(AMI\) \(p. 82\)](#).

Extending a Linux File System after Resizing the Volume

Use a file system-specific command to resize the file system to the larger size of the new volume. These commands work even if the volume to extend is the root volume. For ext2, ext3, and ext4 file systems, this command is **resize2fs**. For XFS file systems, this command is **xfs_growfs**. For other file systems, refer to the specific documentation for those file systems for instructions on extending them.

If you are unsure of which file system you are using, you can use the **file -s** command to list the file system data for a device. The following example shows a Linux ext4 file system and an SGI XFS file system.

```
[ec2-user ~]$ sudo file -s /dev/xvd*
```

```
/dev/xvda1: Linux rev 1.0 ext4 filesystem data ...
/dev/xvdf: SGI XFS filesystem data ...
```

Note

If the volume you are extending has been partitioned, you need to increase the size of the partition before you can resize the file system. You can also allocate additional partitions at this time. For more information, see [Expanding a Linux Partition](#).

You can begin resizing the file system as soon as the volume enters the Optimizing state.

Important

Before extending a file system that contains valuable data, it is a best practice to create a snapshot of the volume that contains it in case you need to roll back your changes. For information about EBS snapshots, see [Creating an Amazon EBS Snapshot](#).

For information about extending a Windows file system, see [Extending a Windows File System after Resizing the Volume](#) in the *Amazon EC2 User Guide for Windows Instances*.

To check if your volume partition needs resizing

- Use the `lsblk` command to list the block devices attached to your instance. The example below shows three volumes: `/dev/xvda`, `/dev/xvdb`, and `/dev/xvdf`.

```
[ec2-user ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   30G  0 disk 
##xvda1 202:1    0   30G  0 part /
xvdb    202:16   0   30G  0 disk /mnt
xvdf    202:80   0   35G  0 disk 
##xvdf1 202:81   0    8G  0 part
```

The root volume, `/dev/xvda1`, is a partition on `/dev/xvda`. Notice that they are both 30 GiB in size. In this case, the partition occupies all of the room on the device, so it does not need resizing.

The volume `/dev/xvdb` is not partitioned at all, so it does not need resizing.

However, `/dev/xvdf1` is an 8-GiB partition on a 35-GiB device and there are no other partitions on the volume. In this case, the partition must be resized in order to use the remaining space on the volume. For more information, see [Expanding a Linux Partition](#). After you resize the partition, you can follow the next procedure to extend the file system to occupy all of the space on the partition.

To extend a Linux file system

1. Log in to your Linux instance using an SSH client. For more information about connecting to a Linux instance, see [Connecting to Your Linux Instance Using SSH](#).
2. Use the `df -h` command to report the existing disk space usage on the file system. In this new example, `/dev/xvda1` device has already been expanded to 35 GiB, but the operating system still sees only an original 8 GiB ext4 file system. Similarly, the `/dev/xvdf` device has been expanded to 35 GiB, but the operating system still only sees an original 1 GiB XFS file system.

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1       8.0G  943M  6.9G  12% /
tmpfs           1.9G     0  1.9G  0% /dev/shm
/dev/xvdf        1014M   33M  982M  4% /mnt
```

3. Expand the modified partition using `growpart` (and note the unusual syntax of separating the device name from the partition number):

```
$sudo growpart /dev/xvdf 1
CHANGED: disk=/dev/xvdf partition=1: start=4096 old: size=16773086,end=16777182 new:
size=73396190,end=73400286
```

A look at the **lsblk** output confirms that the partition `/dev/xvdf1` now fills the available space on the volume `/dev/xvdf`:

```
[ec2-user ~]$ lsblk
NAME      MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
...
xvdf      202:80   0 35G  0 disk
##xvdf1  202:81   0 35G  0 part
```

4. Use a file system-specific command to resize each file system to the new volume capacity.

For a Linux ext2, ext3, or ext4 file system, use the following command, substituting the device name to extend:

```
[ec2-user ~]$ sudo resize2fs /dev/xvdf1
resize2fs 1.42.3 (14-May-2012)
old_desc_blocks = 1, new_desc_blocks = 3
The filesystem on /dev/xvdf1 is now 9174523 blocks long.
```

For an XFS file system, first install the XFS userspace tools:

```
[ec2-user ~]$ sudo yum install xfsprogs
```

Then use the following command, substituting the mount point of the file system (XFS file systems must be mounted to resize them):

```
[ec2-user ~]$ sudo xfs_growfs -d /mnt
meta-data=/dev/xvdf          isize=256    agcount=4, agsize=65536 blks
                           =           sectsz=512  attr=2
data        =           bsize=4096   blocks=262144, imaxpct=25
                           =           sunit=0    swidth=0 blks
naming      =version 2     bsize=4096   ascii-ci=0
log         =internal      bsize=4096   blocks=2560, version=2
                           =           sectsz=512  sunit=0 blks, lazy-count=1
realtime    =none          extsz=4096   blocks=0, rtextents=0
data blocks changed from 262144 to 26214400
```

Note

If you receive an `xfsctl failed: Cannot allocate memory` error, you may need to update the Linux kernel on your instance. For more information, refer to your specific operating system documentation.

If you receive a `The filesystem is already nnnnnnn blocks long. Nothing to do!` error, see [Expanding a Linux Partition](#).

5. Use the **df -h** command to report the existing file system disk space usage, in this example showing 70 GiB on the ext4 file system and 100 GiB on the XFS file system:

```
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1       70G  951M   69G  2% /
tmpfs            1.9G    0  1.9G  0% /dev/shm
/dev/xvdf       100G   45M  100G  1% /mnt
```

Tip

If the increased available space on your volume remains invisible to the system, try re-initializing the volume as described in [Initializing Amazon EBS Volumes](#).

Limitations When Modifying EBS Volumes

Be aware of the following limits and requirements when you modify an EBS volume:

- 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 more information, see [Detaching an Amazon EBS Volume from an Instance](#) and [Attaching an Amazon EBS Volume to an Instance](#).
 - For a root (boot) volume, stop the instance, apply the modifications, and then restart the instance. For more information, see [Appendix: Starting and Stopping an Instance to Modify an EBS Volume \(p. 791\)](#).
- 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.
- After modifying a volume, wait at least six hours before applying further modifications to the same volume.
- While m3.medium instances fully support volume modification, some m3.large, m3.xlarge, and m3.2xlarge instances might not support all volume modification features. If you encounter an error, see [Appendix: Starting and Stopping an Instance to Modify an EBS Volume \(p. 791\)](#).

Volume Modification Support on Older Volumes

Before you can modify a volume that was attached to an instance before November 1, 2016, you must initialize volume modification support using one of the following actions:

- Detach and attach the volume
- Restart the instance

To determine whether you must initialize volume modification support 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, regardless of when the volumes were attached.

Instance ID	Launch Time	Block Devices
i-e905622e	February 25, 2016 at 1:49:35 PM UTC-8	/dev/xvda=vol-e6b46410:attached:2
i-719f99a8	December 8, 2016 at 2:21:51 PM UTC-8	/dev/xvda=vol-bad60e7a:attached:2
i-006b02c1b78381e57	May 17, 2017 at 1:52:52 PM UTC-7	/dev/sda1=vol-0de9250441c73024c:attached:1
i-e3d172ed	May 17, 2017 at 2:48:54 PM UTC-7	/dev/sda1=vol-04c34d0b:attached:1

To determine whether you must initialize volume modification support using the CLI

To find an instance that was last started before the cutoff date with a volume that was attached before the cutoff date, use the following [describe-instances](#) command.

```
aws ec2 describe-instances --query "Reservations[*].Instances[*].[InstanceId,LaunchTime<=`2016-11-01`,BlockDeviceMappings[*][Ebs.AttachTime<=`2016-11-01`]]" --output text
```

The output for each instance shows its ID, whether it was started before the cutoff date (True or False), and whether its volumes were 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, regardless of when the volumes were attached.

```
i-e905622e      True
True
i-719f99a8      False
True
i-006b02c1b78381e57  False
False
i-e3d172ed      False
True
```

Appendix: Starting and Stopping an Instance to Modify an EBS Volume

If you are using a previous generation Amazon EC2 instance and you need to modify the root (boot) volume, you must stop the instance, apply the modifications, and then restart the instance. The procedure described here can be used to modify any EBS volume on any instance type.

When you stop and start an instance, be aware of the following:

- If your instance is running in a VPC and has a public IPv4 address, we release the address and give it a new public IPv4 address. The instance retains its private IPv4 addresses and any Elastic IP addresses.
- If your instance is running in EC2-Classic, we give it new public and private IPv4 addresses, and disassociate any Elastic IP address that's associated with the instance. You must re-associate any Elastic IP address after you restart your instance.
- If your instance is in an Auto Scaling group, Amazon EC2 Auto Scaling marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. To prevent this, you can temporarily suspend the Auto Scaling processes for the group. For more information, see [Suspending and Resuming Scaling Processes](#) in the *Amazon EC2 Auto Scaling User Guide*.

To modify the root volume of an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and select the instance with the volume to expand.
3. Verify that **Shutdown Behavior** is set to **Stop** and not **Terminate**.
 - a. Select the instance.
 - b. From the context (right-click) menu, choose **Instance Settings, Change Shutdown Behavior**.
 - c. If **Shutdown behavior** is set to **Terminate**, choose **Stop, Apply**.
If **Shutdown behavior** is already set to **Stop**, choose **Cancel**.
4. Stop the instance. For more information, see [Stopping and Starting Your Instances \(p. 417\)](#).

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

5. Modify your EBS volume as described in [Modifying an EBS Volume from the Console \(p. 773\)](#) or [Modifying an EBS Volume from the Command Line \(p. 773\)](#).
6. Restart the instance.
 - a. In the navigation pane, choose **Instances** and then select the instance to restart.
 - b. From the context (right-click) menu, choose **Instance State, Start**.
 - c. In the **Start Instances** dialog box, choose **Yes, Start**. If the instance fails to start, and the volume being expanded is a root volume, verify that you attached the expanded volume using the same device name as the original volume, for example /dev/sda1.

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 the Volume \(p. 787\)](#).

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. When you delete a snapshot, only the data unique to that snapshot is removed. Each snapshot contains all of the information 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 lazily 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 an Amazon EBS Snapshot \(p. 795\)](#).

You can track the status of your EBS snapshots through CloudWatch Events. For more information, see [Amazon CloudWatch Events for Amazon EBS](#).

Contents

- [How Incremental Snapshots Work \(p. 793\)](#)
- [Copying and Sharing Snapshots \(p. 795\)](#)
- [Encryption Support for Snapshots \(p. 795\)](#)
- [Creating an Amazon EBS Snapshot \(p. 795\)](#)
- [Deleting an Amazon EBS Snapshot \(p. 796\)](#)
- [Copying an Amazon EBS Snapshot \(p. 799\)](#)
- [Viewing Amazon EBS Snapshot Information \(p. 801\)](#)
- [Sharing an Amazon EBS Snapshot \(p. 802\)](#)
- [Automating the Amazon EBS Snapshot Lifecycle \(p. 804\)](#)

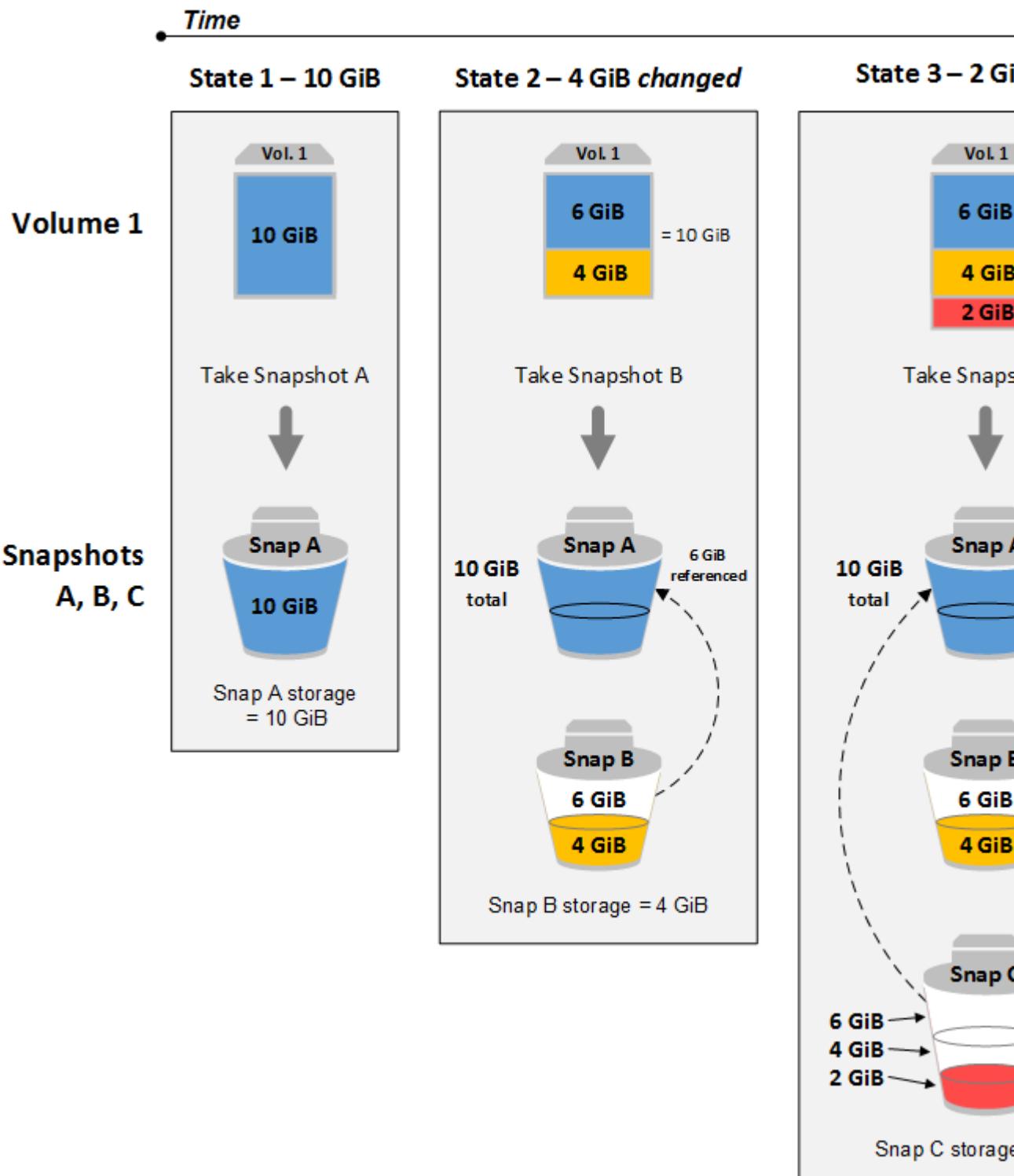
How Incremental Snapshots Work

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 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 the 4 GiB of data stored in Snap B, and the 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



For more information about how data is managed when you delete a snapshot, see [Deleting an Amazon EBS Snapshot \(p. 796\)](#).

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 \(p. 802\)](#).

A snapshot is constrained to the 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 [Restoring an Amazon EBS Volume from a Snapshot \(p. 748\)](#). 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 \(p. 799\)](#).

Encryption Support for Snapshots

EBS snapshots broadly support EBS encryption.

- Snapshots of encrypted volumes are automatically encrypted.
- Volumes that are created from encrypted snapshots are automatically encrypted.
- 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, you can reencrypt it with a different key during the copy process.

For more information, see [Amazon EBS Encryption](#).

Creating an Amazon EBS Snapshot

A point-in-time snapshot of an EBS volume, can be used as a baseline for new volumes or for data backup. If you make periodic snapshots of a volume, the snapshots are incremental—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 restore the entire volume.

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.

Important

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 may result in reduced volume performance until the snapshots complete.

There is a limit of five pending snapshots for a single gp2, io1, 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.

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](#).

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 \(p. 802\)](#).

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 to restore the volume. Your copy of a shared, encrypted snapshot can also be re-encrypted with a different key. For more information, see [Sharing an Amazon EBS Snapshot \(p. 802\)](#).

When a snapshot is created from a volume with an AWS Marketplace product code, the product code is propagated to the snapshot.

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 create a snapshot for an Amazon EBS volume that serves as a root device, you should stop the instance before taking the snapshot.

To unmount the volume in Linux, use the following command, where *device_name* is the device name (for example, /dev/sdh):

```
umount -d device_name
```

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 \(p. 887\)](#).

To create 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. Choose **Create Snapshot**.
4. On the **Create Snapshot** page, select the volume to create a snapshot for.
5. (Optional) Choose **Add tags to your snapshot**. For each tag, provide a tag key and a tag value.
6. 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 \(AWS Tools for Windows PowerShell\)](#)

Deleting an Amazon EBS Snapshot

When you delete a snapshot, only the data referenced exclusively by that snapshot is removed. Deleting previous snapshots of a volume does not affect your ability to restore volumes from later snapshots of that volume.

Deleting a snapshot of a volume has no effect on the volume. Deleting a volume has no effect on the snapshots made from it.

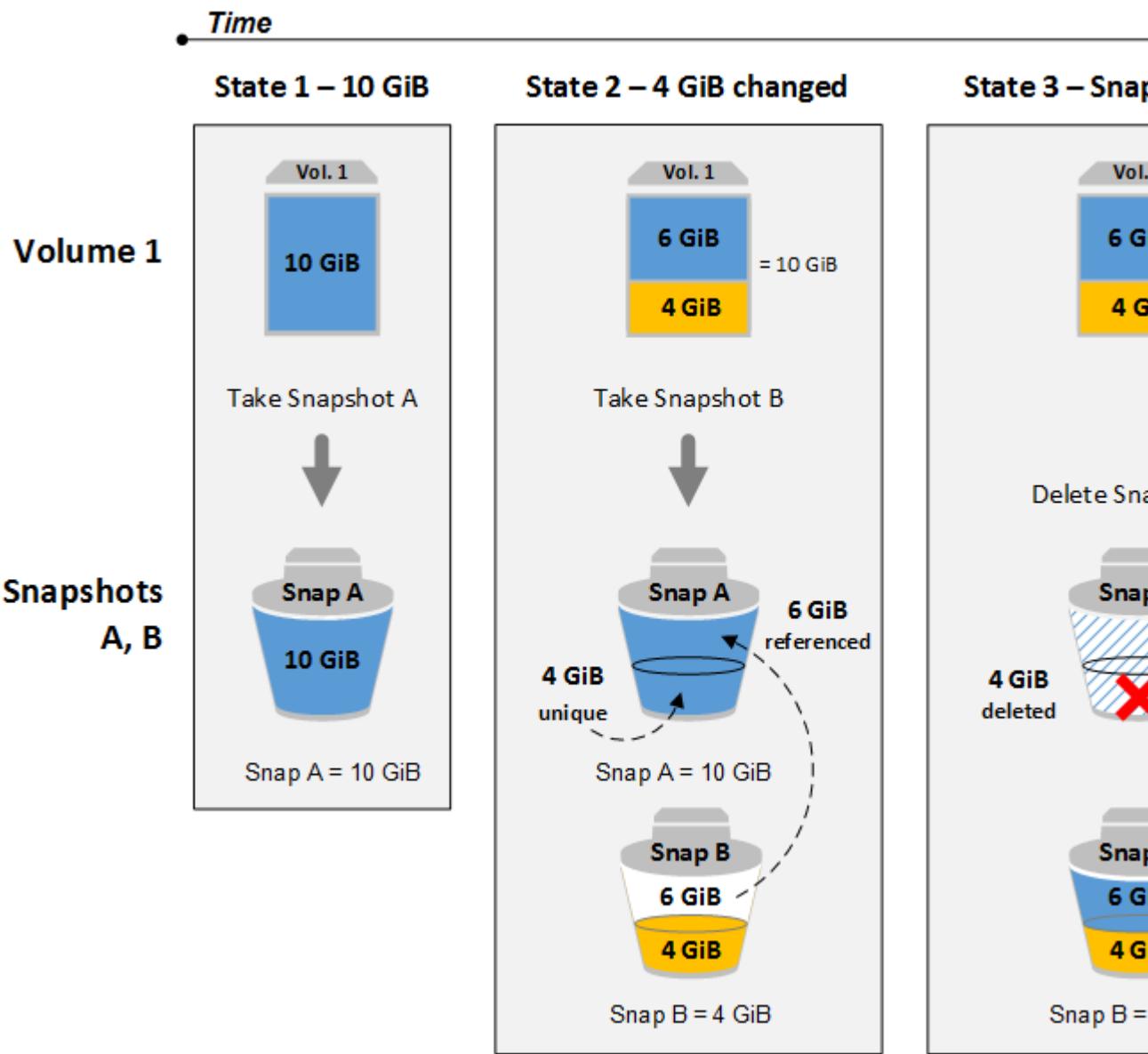
If you make periodic snapshots of a volume, the snapshots are *incremental*, which means that only the blocks on the device that have changed after your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the 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 \(p. 793\)](#) and the example below.

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.

Example 1: Deleting a Snapshot with Some of its Data Referenced by Another Snapshot



Note that 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 \(p. 144\)](#).

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 \(AWS Tools for Windows PowerShell\)](#)

Note

Although you can delete a snapshot that is still in progress, the snapshot must complete before the deletion takes effect. This may take a long time. If you are also at your concurrent snapshot limit (five snapshots in progress), and you attempt to take an additional snapshot, you may get the `ConcurrentSnapshotLimitExceeded` error.

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've created 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 than the ID of the original snapshot.

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 may copy it. For more information, see [Sharing an Amazon EBS Snapshot \(p. 802\)](#).

For pricing information about copying snapshots across 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. Copying a snapshot to a new region does incur new storage costs.

Use Cases

- Geographic expansion: Launch your applications in a new 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, for encrypted snapshots that have been shared with you, create a copy that you own in order to restore a volume from it.
- 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've 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 5 concurrent snapshot copy requests to a single destination region.

- User-defined tags are not copied from the source snapshot to the new snapshot. After the copy operation is complete, you can apply user-defined tags to the new snapshot. For more information, see [Tagging Your Amazon EC2 Resources \(p. 887\)](#).
- Snapshots created by the `CopySnapshot` action have an arbitrary volume ID that should not be used for any purpose.

Incremental Copying Across Regions

The first snapshot copy to another region is always a full copy. For unencrypted snapshots, each subsequent snapshot copy of the same volume is incremental, meaning that AWS copies only the data that changed since your last snapshot copy to the same destination region. This allows for faster copying and lower storage costs.

In the case of encrypted snapshots, you must encrypt to the same CMK that was used for previous copies to get incremental copies. The following examples illustrate how this works:

- If you copy an unencrypted snapshot from the US East (N. Virginia) region to the US West (Oregon) region, the first snapshot copy is a full copy and subsequent snapshot copies of the same volume transferred between the same regions are incremental.
- If you copy an encrypted snapshot from the US East (N. Virginia) region to the US West (Oregon) region, the first snapshot copy of the volume is a full copy.
 - If you encrypt to the same CMK in a subsequent snapshot copy for the same volume between the same regions, the copy is incremental.
 - If you encrypt to a different CMK in a subsequent snapshot copy for the same volume between the same regions, the copy is a new full copy of the snapshot.

For more information, see [Encrypt a Snapshot to a New CMK](#).

Encrypted Snapshots

When you copy a snapshot, you can choose to encrypt the copy (if the original snapshot was not encrypted) 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 you created using the snapshot. For more information, see [Sharing an Amazon EBS Snapshot \(p. 802\)](#).

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. *You cannot strip encryption from an encrypted snapshot.*
 - **Master Key:** The customer master key (CMK) that to be used to encrypt this snapshot. You can select from master keys in your account or type/paste the ARN of a key from a different account. You can create a new master encryption key in the IAM console.
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. For 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 filter your snapshots further by using the advanced search options. Choose the search bar to view the filters available.
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 \(AWS Tools for Windows PowerShell\)](#)

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 custom CMK used to encrypt the snapshot. You can apply cross-account permissions to a custom CMK either when it is created or at a later time.

Important

When you share a snapshot, you are giving others access to all the data on the snapshot. Share snapshots only with people with whom you want to share *all* 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 \(p. 799\)](#).
- If your snapshot uses the longer resource ID format, you can only share it with another account that also supports longer IDs. For more information, see [Resource IDs](#).
- 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 custom 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](#).
- If you have access to a shared encrypted snapshot and you want to restore a volume from it, you must create a personal copy of the snapshot and then use that copy to restore the volume. We recommend that you re-encrypt the snapshot during the copy process with a different key that you control. This protects your access to the volume if the original key is compromised, or if the owner revokes the key for any reason.

Sharing an Unencrypted Snapshot

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**, type 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 encrypted snapshot that was shared with me

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

To share an encrypted snapshot using the console

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Encryption keys** in the navigation pane.
3. Choose the alias of the custom key that you used to encrypt the snapshot.
4. For each AWS account, choose **Add External Accounts** and type the AWS account ID where prompted. When you have added all AWS accounts, choose **Save Changes**.
5. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
6. Choose **Snapshots** in the navigation pane.
7. Select the snapshot and then choose **Actions, Modify Permissions**.
8. For each AWS account, type 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 me

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. We recommend that you re-encrypt the snapshot with a different key that you own. This protects you if the original key is compromised, or if the owner revokes the key, which could cause you to lose access to any encrypted volumes you create from the snapshot.
 - a. Select the snapshot and choose **Actions, Copy**.
 - b. (Optional) Select a destination region.
 - c. Select a custom CMK that you own.
 - d. Choose **Copy**.

Sharing an Snapshot Using the Command Line

To view and modify snapshot permissions using the command line

To view the `createVolumePermission` attribute of a snapshot, you can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-snapshot-attribute](#) (AWS CLI)
- [Get-EC2SnapshotAttribute](#) (AWS Tools for Windows PowerShell)

To modify the `createVolumePermission` attribute of a snapshot, you can use one of the following commands.

- [modify-snapshot-attribute](#) (AWS CLI)
- [Edit-EC2SnapshotAttribute](#) (AWS Tools for Windows PowerShell)

Automating the Amazon EBS Snapshot Lifecycle

You can use Amazon Data Lifecycle Manager (Amazon DLM) to automate the creation, retention, and deletion of snapshots taken to back up your Amazon EBS volumes. Automating snapshot management 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.

Combined with the monitoring features of Amazon CloudWatch Events and AWS CloudTrail, Amazon DLM provides a complete backup solution for EBS volumes at no additional cost.

Understanding Amazon DLM

The following are the key elements that you should understand before you get started with Amazon DLM.

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](#).

Volume Tags

Amazon DLM 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 DLM policy (described below) targets a volume for backup using a single unique 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 about tagging Amazon EC2 objects, see [Tagging Your Amazon EC2 Resources](#).

Snapshot Tags

Amazon DLM 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`

You can also specify custom tags to be applied to snapshots on creation.

You can't use a '\' or '=' character in a tag key.

Volume tags are not inherited by the snapshots that Amazon DLM creates.

Lifecycle Policies

A lifecycle policy consists of these core settings:

- Resource type—The AWS resource managed by the policy, in this case, EBS volumes.
- Target tag—The tag that must be associated with an EBS volume for it to be managed by the policy.
- Schedule—Defines how often to create snapshots and the maximum number of snapshots to keep. Snapshot creation starts within an hour of the specified start time. If creating a new snapshot exceeds the maximum number of snapshots to keep for the volume, the oldest snapshot is deleted.

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.
- Snapshots begin to be created by a policy within one hour following the specified start 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.
- You can create multiple policies to back up an EBS volume, as long as each policy targets a unique tag on the volume. Target tags cannot be reused across policies, even disabled policies. 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 DLM creates snapshots according to the schedules for both policies.
- When you copy a snapshot created by a policy, the retention schedule is not carried over to the copy. This ensures that Amazon DLM does not delete snapshots that should be retained for a longer period of time.

For example, you could create a policy that manages all EBS volumes with the tag `account=Finance`, creates snapshots every 24 hours at 0900, and retains the five most recent snapshots. Snapshot creation could start as late as 0959.

Permissions for Amazon DLM

Amazon DLM uses an IAM role to get the permissions that are required to manage snapshots on your behalf. Amazon DLM creates the **AWSDataLifecycleManagerDefaultRole** role the first time that you create a lifecycle policy using the AWS Management Console. You can also create this role using the [create-default-role](#) command as follows:

```
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>DeleteSnapshot",
    "ec2:DescribeVolumes",
    "ec2:DescribeSnapshots"
],
"Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "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 DLM:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "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*.

Limits

Your AWS account has the following limits related to Amazon DLM:

- You can create up to 100 lifecycle policies per region.
- You can add up to 50 tags per resource.
- You can create one schedule per lifecycle policy.

Working with Amazon DLM Using the Console

The following examples show how to use Amazon DLM to perform typical procedures to manage the backups of your EBS volumes.

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, Create snapshot lifecycle policy**.
3. Provide the following information for your policy as needed:
 - **Description**—A description of the policy.
 - **Target volumes with tags**—The resource tags that identify the volumes to back up.
 - **Schedule Name**—A name for the backup schedule.
 - **Create snapshots every *n* Hours**—The number of hours between policy runs. The supported values are 12 and 24.
 - **Snapshot creation start time *hh:mm UTC***—The time of day when policy runs are scheduled to start. The policy runs start within an hour after the scheduled time.
 - **Retention rule**—The maximum number of snapshots to retain for each volume. The supported range is 1 to 1000. After the limit is reached, the oldest snapshot is deleted when a new one is created.
 - **Tag created snapshots**—The resource tags to apply to the snapshots that are created. These tags are in addition to the tags applied by Amazon DLM.
 - **IAM role**—An 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.
 - **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.
4. Choose **Create Policy**.

To display 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 the following information about the policy:
 - **Policy ID**
 - **Date created**
 - **Date modified**
 - **Target volumes with these tags**
 - **Rule summary**
 - **Description**

- **Policy state**
- **Tags added to snapshots**

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 policy**.
5. In an existing lifecycle policy, you can modify the following policy values:
 - **Description**—A description of the policy.
 - **Target volumes with tags**—The resource tags that identify the volumes to back up.
 - **Schedule Name**—A name for the backup schedule.
 - **Create snapshots every *n* Hours**—The number of hours between policy runs. The supported values are 12 and 24.
 - **Snapshot creation start time *hh:mm UTC***—The time of day when policy runs are scheduled to start. The policy runs start within an hour after the scheduled time.
 - **Retention rule**—The maximum number of snapshots to retain for each volume. The supported range is 1 to 1000. After the limit is reached, the oldest snapshot is deleted when a new one is created.
 - **Tag created snapshots**—The resource tags to apply to the snapshots that are created. These tags are in addition to the tags applied by Amazon DLM.
 - **IAM role**—An 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.
 - **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.

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 policy**.

Working with Amazon DLM Using the Command Line

The following examples show how to use Amazon DLM to perform typical procedures to manage the backups of your EBS volumes.

Example Example: Create a lifecycle policy

Use the `create-lifecycle-policy` command to create a lifecycle policy. To simplify the syntax, this example references a JSON file, `policyDetails.json`, that includes the policy details.

```
aws dlm create-lifecycle-policy --description "My first 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  
            }  
        }  
    ]  
}
```

Upon success, the command returns the ID of the newly created policy. The following is example output:

```
{  
    "PolicyId": "policy-0123456789abcdef0"  
}
```

Example Example: 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": {  
            "ResourceTypes": [  
                "VOLUME"  
            ],  
            "TargetTags": [  
            ]  
        }  
    }  
}
```

```
{  
    "Value": "115",  
    "Key": "costcenter"  
}  
,  
"Schedules": [  
    {  
        "TagsToAdd": [  
            {  
                "Value": "myDailySnapshot",  
                "Key": "type"  
            }  
        ],  
        "RetainRule": {  
            "Count": 5  
        },  
        "CreateRule": {  
            "Interval": 24,  
            "IntervalUnit": "HOURS",  
            "Times": [  
                "03:00"  
            ]  
        },  
        "Name": "DailySnapshots"  
    }  
]  
}  
}
```

Example To 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
        }
    ]
}
```

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.

Example Example: 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.

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

Working with Amazon DLM 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 DLM Query API.

Alternatively, you can 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](#).

Monitoring the Snapshot Lifecycle

You can use the following features to monitor the lifecycle of your snapshots.

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 time stamp 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 [To display a lifecycle policy \(p. 807\)](#). For information about displaying information about lifecycle policies using the CLI, see [Example: Display a lifecycle policy \(p. 809\)](#).

CloudWatch Events

Amazon EBS and Amazon DLM 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](#).
- `DLM Policy State Change`—A Amazon DLM 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–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, with options between 425 Mbps and 14,000 Mbps, depending on the instance type you use. When attached to an EBS–optimized instance, General Purpose SSD (gp2) volumes are designed to deliver within 10% of their baseline and burst performance 99% of the time in a given year, and Provisioned IOPS SSD (io1) volumes are designed to deliver within 10% of their provisioned performance 99.9% of the time in a given year. Both Throughput Optimized HDD (st1) and Cold HDD (sc1) guarantee performance consistency of 90% of burst throughput 99% of the time in a given year. Non-compliant periods are approximately uniformly distributed, targeting 99% of expected total throughput each hour. For more information, see [Amazon EBS Volume Types \(p. 735\)](#).

Contents

- [Instance Types that Support EBS Optimization \(p. 813\)](#)
- [Enabling Amazon EBS Optimization at Launch \(p. 818\)](#)
- [Modifying Amazon EBS Optimization for a Running Instance \(p. 818\)](#)

Instance Types that Support EBS Optimization

The following tables show which instance types support EBS optimization, the dedicated bandwidth to Amazon EBS, the maximum number of IOPS the instance can support if you are using a 16 KB I/O size, and the typical maximum aggregate throughput that can be achieved on that connection in MiB/s with a streaming read workload and 128 KB 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.

The instance types that are EBS-optimized by default, there is no need to enable EBS optimization and no effect if you disable EBS optimization. For instances that are not EBS-optimized by default, you can enable EBS optimization when you launch the instances, or enable EBS optimization after the instances are running. Instances must have EBS optimization enabled to achieve the level of performance described in the table below.

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 page for On-Demand instances](#).

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.

Supported Current Generation Instance Types

The following table lists current-generation instance types that support EBS optimization.

Instance type	EBS-optimized by default	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KB I/O)	Maximum IOPS (16 KB I/O)
<code>c4.large</code>	Yes	500	62.5	4,000
<code>c4.xlarge</code>	Yes	750	93.75	6,000
<code>c4.2xlarge</code>	Yes	1,000	125	8,000
<code>c4.4xlarge</code>	Yes	2,000	250	16,000
<code>c4.8xlarge</code>	Yes	4,000	500	32,000
<code>c5.large</code> *	Yes	2,250	281.25	16,000
<code>c5.xlarge</code> *	Yes	2,250	281.25	16,000
<code>c5.2xlarge</code> *	Yes	2,250	281.25	16,000
<code>c5.4xlarge</code>	Yes	2,250	281.25	16,000
<code>c5.9xlarge</code>	Yes	4,500	562.5	32,000
<code>c5.18xlarge</code>	Yes	9,000	1,125	64,000
<code>c5d.large</code> *	Yes	2,250	281.25	16,000
<code>c5d.xlarge</code> *	Yes	2,250	281.25	16,000
<code>c5d.2xlarge</code> *	Yes	2,250	281.25	16,000
<code>c5d.4xlarge</code>	Yes	2,250	281.25	16,000

Instance type	EBS-optimized by default	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KB I/O)	Maximum IOPS (16 KB I/O)
c5d.9xlarge	Yes	4,500	562.5	32,000
c5d.18xlarge	Yes	9,000	1,125	64,000
d2.xlarge	Yes	750	93.75	6,000
d2.2xlarge	Yes	1,000	125	8,000
d2.4xlarge	Yes	2,000	250	16,000
d2.8xlarge	Yes	4,000	500	32,000
f1.2xlarge	Yes	1,700	212.5	12,000
f1.16xlarge	Yes	14,000	1,750	75,000
g3.4xlarge	Yes	3,500	437.5	20,000
g3.8xlarge	Yes	7,000	875	40,000
g3.16xlarge	Yes	14,000	1,750	80,000
h1.2xlarge	Yes	1,750	218.75	12,000
h1.4xlarge	Yes	3,500	437.5	20,000
h1.8xlarge	Yes	7,000	875	40,000
h1.16xlarge	Yes	14,000	1,750	80,000
i3.large	Yes	425	53.13	3000
i3.xlarge	Yes	850	106.25	6000
i3.2xlarge	Yes	1,700	212.5	12,000
i3.4xlarge	Yes	3,500	437.5	16,000
i3.8xlarge	Yes	7,000	875	32,500
i3.16xlarge	Yes	14,000	1,750	65,000
i3.metal	Yes	10,000	1,250	64,000
m4.large	Yes	450	56.25	3,600
m4.xlarge	Yes	750	93.75	6,000
m4.2xlarge	Yes	1,000	125	8,000
m4.4xlarge	Yes	2,000	250	16,000
m4.10xlarge	Yes	4,000	500	32,000
m4.16xlarge	Yes	10,000	1,250	65,000
m5.large *	Yes	2,120	265	16,000
m5.xlarge *	Yes	2,120	265	16,000

Instance type	EBS-optimized by default	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KB I/O)	Maximum IOPS (16 KB I/O)
m5.2xlarge *	Yes	2,120	265	16,000
m5.4xlarge	Yes	2,120	265	16,000
m5.12xlarge	Yes	5,000	625	32,500
m5.24xlarge	Yes	10,000	1,250	65,000
m5d.large *	Yes	2,120	265	16,000
m5d.xlarge *	Yes	2,120	265	16,000
m5d.2xlarge *	Yes	2,120	265	16,000
m5d.4xlarge	Yes	2,120	265	16,000
m5d.12xlarge	Yes	5,000	625	32,500
m5d.24xlarge	Yes	10,000	1,250	65,000
p2.xlarge	Yes	750	93.75	6,000
p2.8xlarge	Yes	5,000	625	32,500
p2.16xlarge	Yes	10,000	1,250	65,000
p3.2xlarge	Yes	1,750	218	10,000
p3.8xlarge	Yes	7,000	875	40,000
p3.16xlarge	Yes	14,000	1,750	80,000
r4.large	Yes	425	53.13	3,000
r4.xlarge	Yes	850	106.25	6,000
r4.2xlarge	Yes	1,700	212.5	12,000
r4.4xlarge	Yes	3,500	437.5	18,750
r4.8xlarge	Yes	7,000	875	37,500
r4.16xlarge	Yes	14,000	1,750	75,000
r5.large *	Yes	3,500	437.5	18,750
r5.xlarge *	Yes	3,500	437.5	18,750
r5.2xlarge *	Yes	3,500	437.5	18,750
r5.4xlarge	Yes	3,500	437.5	18,750
r5.12xlarge	Yes	7,000	875	40,000
r5.24xlarge	Yes	14,000	1,750	80,000
r5d.large *	Yes	3,500	437.5	18,750
r5d.xlarge *	Yes	3,500	437.5	18,750

Instance type	EBS-optimized by default	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KB I/O)	Maximum IOPS (16 KB I/O)
r5d.2xlarge *	Yes	3,500	437.5	18,750
r5d.4xlarge	Yes	3,500	437.5	18,750
r5d.12xlarge	Yes	7,000	875	40,000
r5d.24xlarge	Yes	14,000	1,750	80,000
x1.16xlarge	Yes	7,000	875	40,000
x1.32xlarge	Yes	14,000	1,750	80,000
x1e.xlarge	Yes	500	62.5	3,700
x1e.2xlarge	Yes	1,000	125	7,400
x1e.4xlarge	Yes	1,750	218.75	10,000
x1e.8xlarge	Yes	3,500	437.5	20,000
x1e.16xlarge	Yes	7,000	875	40,000
x1e.32xlarge	Yes	14,000	1,750	80,000
z1d.large *	Yes	2,333	291	13,333
z1d.xlarge *	Yes	2,333	291	13,333
z1d.2xlarge	Yes	2,333	292	13,333
z1d.3xlarge	Yes	3,500	438	20,000
z1d.6xlarge	Yes	7,000	875	40,000
z1d.12xlarge	Yes	14,000	1,750	80,000

* These instance types can support maximum performance for 30 minutes at least once every 24 hours. For example, c5.large instances can deliver 281 MB/s 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 shown in this table.

Instance type	Baseline bandwidth (Mbps)	Baseline throughput (MB/s, 128 KB I/O)	Baseline IOPS (16 KB I/O)
c5.large	525	66	4,000
c5.xlarge	800	100	6,000
c5.2xlarge	1,125	141	8,000
c5d.large	525	66	4,000
c5d.xlarge	800	100	6,000
c5d.2xlarge	1,125	141	8,000

Instance type	Baseline bandwidth (Mbps)	Baseline throughput (MB/s, 128 KB I/O)	Baseline IOPS (16 KB I/O)
m5.large	480	60	3,600
m5.xlarge	800	100	6,000
m5.2xlarge	1,166	146	8,333
m5d.large	480	60	3,600
m5d.xlarge	800	100	6,000
m5d.2xlarge	1,166	146	8,333
r5.large	480	60	3,600
r5.xlarge	850	106.25	6,000
r5.2xlarge	1,700	212.5	12,000
r5d.large	480	60	3,600
r5d.xlarge	850	106.25	6,000
r5d.2xlarge	1,700	212.5	12,000
z1d.large	583	73	3,333
z1d.xlarge	1,167	146	6,667

The `EBSIOBalance%` and `EBSByteBalance%` metrics can help you determine if 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 for upsizing. Instances where the balance percentage never drops below 100% are candidates for downsizing. For more information, see [Monitoring Your Instances Using CloudWatch \(p. 483\)](#).

Supported Previous Generation Instance Types

The following table lists previous-generation instance types that support EBS optimization.

Previous Generation Instances

Instance type	EBS-optimized by default	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KB I/O)	Maximum IOPS (16 KB I/O)
c1.xlarge	No	1,000	125	8,000
c3.xlarge	No	500	62.5	4,000
c3.2xlarge	No	1,000	125	8,000
c3.4xlarge	No	2,000	250	16,000
g2.2xlarge	No	1,000	125	8,000
i2.xlarge	No	500	62.5	4,000
i2.2xlarge	No	1,000	125	8,000

Instance type	EBS-optimized by default	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KB I/O)	Maximum IOPS (16 KB I/O)
i2.4xlarge	No	2,000	250	16,000
m1.large	No	500	62.5	4,000
m1.xlarge	No	1,000	125	8,000
m2.2xlarge	No	500	62.5	4,000
m2.4xlarge	No	1,000	125	8,000
m3.xlarge	No	500	62.5	4,000
m3.2xlarge	No	1,000	125	8,000
r3.xlarge	No	500	62.5	4,000
r3.2xlarge	No	1,000	125	8,000
r3.4xlarge	No	2,000	250	16,000

Enabling Amazon EBS Optimization at Launch

You can enable optimization for an instance by setting its Amazon EBS-optimized attribute.

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 options with the corresponding command. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

- `--ebs-optimized` with [run-instances](#) (AWS CLI)
- `-EbsOptimized` with [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

Modifying Amazon EBS Optimization for a Running Instance

You can enable or disable optimization for a running instance by modifying its Amazon EBS-optimized instance attribute.

To enable EBS optimization for a running instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, click **Instances**, and select the instance.
3. Click **Actions**, select **Instance State**, and then click **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

4. In the confirmation dialog box, click **Yes, Stop**. It can take a few minutes for the instance to stop.
5. With the instance still selected, click **Actions**, select **Instance Settings**, and then click **Change Instance Type**.
6. In the **Change Instance Type** dialog box, 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, 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, and then choose **EBS-optimized, Apply**.
7. Choose **Actions, Instance State, Start**.

To enable EBS optimization for a running instance using the command line

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

- `--ebs-optimized` with [modify-instance-attribute](#) (AWS CLI)
- `-EbsOptimized` with [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

Amazon EBS Encryption

Amazon EBS encryption offers a simple encryption solution for your EBS volumes without the need to build, maintain, and secure your own key management infrastructure. 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

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.

Encryption is supported by all EBS volume types (General Purpose SSD [gp2], Provisioned IOPS SSD [io1], Throughput Optimized HDD [st1], Cold HDD [sc1], and Magnetic [standard]). 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.

Public snapshots of encrypted volumes are not supported, but you can share an encrypted snapshot with specific accounts. For more information about sharing encrypted snapshots, see [Sharing an Amazon EBS Snapshot](#).

Amazon EBS encryption is only available on certain instance types. You can attach both encrypted and unencrypted volumes to a supported instance type. For more information, see [Supported Instance Types \(p. 820\)](#).

Contents

- [Encryption Key Management \(p. 820\)](#)
- [Supported Instance Types \(p. 820\)](#)
- [Changing the Encryption State of Your Data \(p. 821\)](#)
- [Amazon EBS Encryption and CloudWatch Events \(p. 823\)](#)

Encryption Key Management

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) customer master keys (CMKs) when creating encrypted volumes and any snapshots created from them. A unique AWS-managed CMK is created for you automatically in each region where you store AWS assets. This key is used for Amazon EBS encryption unless you specify a customer-managed CMK that you created separately using AWS KMS.

Note

Creating your own CMK gives you more flexibility, including the ability to create, rotate, and disable keys to define access controls. For more information, see the [AWS Key Management Service Developer Guide](#).

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 uses the new CMK.

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—it never appears there in plaintext. The same data key is shared by snapshots of the volume and any subsequent volumes created from those snapshots.

For more information about key management and key access permissions, see [How Amazon Elastic Block Store \(Amazon EBS\) Uses AWS KMS](#) and [Authentication and Access Control for AWS KMS](#) in the [AWS Key Management Service Developer Guide](#).

Supported Instance Types

Amazon EBS encryption is available on the current-generation instance types listed below. These instance types leverage the Intel AES New Instructions (AES-NI) instruction set to provide faster and simpler data protection. You can attach both encrypted and unencrypted volumes to these instance types simultaneously.

- General purpose: T2, M4, M5, M5d
- Compute optimized: C4, C5, C5d
- Memory optimized: R4, R5, R5d, X1, X1e, z1d, cr1.8xlarge
- Storage optimized: D2, I3, h1.2xlarge, h1.4xlarge, i3.metal

- Accelerated computing: F1, G3, P2, P3

For more information about these instance types, see [Amazon EC2 Instance Types](#).

Changing the Encryption State of Your Data

There is no direct way to encrypt an existing unencrypted volume, or to remove encryption from an encrypted volume. However, you can migrate data between encrypted and unencrypted volumes. You can also apply a new encryption status while copying a snapshot:

- While copying an unencrypted snapshot of an unencrypted volume, you can encrypt the copy. Volumes restored from this encrypted copy are also encrypted.
- While copying an encrypted snapshot of an encrypted volume, you can associate the copy with a different CMK. Volumes restored from the encrypted copy are only accessible using the newly applied CMK.

You cannot remove encryption from an encrypted snapshot.

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.

To migrate data between encrypted and unencrypted volumes

1. Create your destination volume (encrypted or unencrypted, depending on your need) by following the procedures in [Creating an Amazon EBS Volume \(p. 747\)](#).
2. Attach the destination volume to the instance that hosts the data to migrate. For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 750\)](#).
3. Make the destination volume available by following the procedures in [Making an Amazon EBS Volume Available for Use on Linux \(p. 751\)](#). For Linux instances, you can create a mount point at `/mnt/destination` and mount the destination volume there.
4. Copy the data from your source directory to the destination volume. It may be most convenient to use a bulk-copy utility for this.

Linux

Use the **rsync** command as follows to copy the data from your source to the destination volume. In this example, 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/
```

Windows

At a command prompt, use the **robocopy** command to copy the data from your source to the destination volume. In this example, the source data is located in `D:\` and the destination volume is mounted at `E:\`.

```
PS C:\> robocopy D:\<sourcefolder> E:\<destinationfolder> /e /copyall /eta
```

Note

We recommend explicitly naming folders rather than copying the entire volume in order to avoid potential problems with hidden folders.

Apply Encryption While Copying a Snapshot

Because you can apply encryption to a snapshot while copying it, another path to encrypting your data is the following procedure.

To encrypt a volume's data by means of snapshot copying

1. Create a snapshot of your unencrypted EBS volume. This snapshot is also unencrypted.
2. Copy the snapshot while applying encryption parameters. The resulting target snapshot is encrypted.
3. Restore the encrypted snapshot to a new volume, which is also encrypted.

For more information, see [Copying an Amazon EBS Snapshot](#).

Encrypt a Snapshot to a New CMK

The ability to encrypt a snapshot during copying also allows you to apply a new CMK to an already-encrypted snapshot that you own. Volumes restored from the resulting copy are only accessible using the new CMK.

Note

If you copy a snapshot to a new CMK, a complete (non-incremental) copy will always be created, resulting in additional storage costs.

In a related scenario, you may choose to apply new encryption parameters to a copy of a snapshot that has been shared with you. Before you can restore a volume from a shared encrypted snapshot, you must create your own copy of it. 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.

The following procedure demonstrates how to create a copy of a shared snapshot to a customer-managed CMK that you own.

To copy a snapshot that you own to a new custom CMK using the console

1. Create a customer-managed CMK. For more information, see [AWS Key Management Service Developer Guide](#).
2. Create an EBS volume encrypted to (for this example) your AWS-managed CMK.
3. Create a snapshot of your encrypted EBS volume. This snapshot is also encrypted to your AWS-managed CMK.
4. On the **Snapshots** page, choose **Actions, Copy**.
5. In the **Copy Snapshot** window, supply the complete ARN for your customer-managed CMK (in the form `arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef`) in the **Master Key** field, or choose it from the menu. Choose **Copy**.

The resulting copy of the snapshot—and all volumes restored from it—are encrypted to your customer-managed CMK.

The following procedure demonstrates how to make a copy of a shared encrypted snapshot to a new CMK that you own. For this to work, you also need access permissions to both the shared encrypted snapshot and to the CMK to which it was originally encrypted.

To copy a shared snapshot to a CMK that you own using the console

1. Select the shared encrypted snapshot on the **Snapshots** page and choose **Actions, Copy**.

2. In the **Copy Snapshot** window, supply the complete ARN for a CMK that you own (in the form `arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef`) in the **Master Key** field, or choose it from the menu. Choose **Copy**.

The resulting copy of the snapshot—and all volumes restored from it—are encrypted to the CMK that you supplied. Changes to the original shared snapshot, its encryption status, or the shared CMK have no effect on your copy.

For more information, see [Copying an Amazon EBS Snapshot](#).

Amazon EBS Encryption and CloudWatch Events

Amazon EBS supports Amazon CloudWatch Events for certain encryption-related scenarios. For more information, see [Amazon CloudWatch Events for Amazon EBS](#).

Amazon EBS and NVMe

With the following instances, EBS volumes are exposed as NVMe block devices: C5, C5d, i3.metal, M5, M5d, R5, R5d, and z1d. 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`).

Note

The EBS performance guarantees stated in [Amazon EBS Product Details](#) are valid regardless of the block-device interface.

Identifying the EBS Device

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.

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.

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/nvme0n1` is attached but not mounted.

```
[ec2-user ~]$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1    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

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/nvme0n1
```

Before you detach an NVMe EBS volume, you should sync and unmount it. When you detach an NVMe EBS volume, the force option is implicitly enabled. Therefore, the instance does not have an opportunity to flush file system caches or metadata before detaching the volume.

I/O Operation Timeout

NVMe EBS volumes 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 (or the `nvme.io_timeout` boot parameter for Linux kernels prior to version 4.6). For an experience similar to EBS volumes attached to Xen instances, we recommend setting this to the highest value possible. For Amazon Linux AMI 2017.09.01 (or greater), and for Linux kernels with version 4.15 or greater, the maximum is 4294967295. Prior to Linux 4.15, the maximum is 255 seconds. If you are using a current version of the Amazon Linux AMI, we have already increased the timeout.

With Linux kernel 4.14 and later, you can also configure the number of times that I/O operations can be retried. The default is five retries. You can configure a different value using the `nvme_core.max_retries` kernel boot parameter, or at runtime using the following command:

```
[ec2-user ~]$ echo 10 > /sys/module/nvme_core/parameters/max_retries
```

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.

Contents

- [Amazon EBS Performance Tips \(p. 825\)](#)
- [Amazon EC2 Instance Configuration \(p. 827\)](#)
- [I/O Characteristics and Monitoring \(p. 828\)](#)
- [Initializing Amazon EBS Volumes \(p. 830\)](#)
- [RAID Configuration on Linux \(p. 831\)](#)
- [Benchmark EBS Volumes \(p. 836\)](#)

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 EC2 Instance Configuration \(p. 827\)](#).

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 \(p. 828\)](#).

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 \(p. 836\)](#).

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 restored from a snapshot. You can avoid this performance hit by accessing 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 \(p. 830\)](#).

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 restored 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](#).

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 \(p. 828\)](#)

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. The following examples assume that you are on an Amazon Linux instance.

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 3.11 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 \(p. 828\)](#).

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](#). 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, 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 \(p. 831\)](#).

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 \(p. 754\)](#).

Amazon EC2 Instance Configuration

When you plan and configure EBS volumes for your application, it is important to consider the configuration of the instances that you will attach the volumes to. In order to get the most performance out of your EBS volumes, you should attach them to an instance with enough bandwidth to support your volumes, such as an EBS-optimized instance or an instance with 10 Gigabit network connectivity. This is especially important when you stripe multiple volumes together in a RAID configuration.

Use EBS-Optimized or 10 Gigabit Network Instances

Any performance-sensitive workloads that require minimal variability and dedicated Amazon EC2 to Amazon EBS traffic, such as production databases or business applications, should use volumes that are attached to an EBS-optimized instance or an instance with 10 Gigabit network connectivity. EC2 instances that do not meet this criteria offer no guarantee of network resources. The only way to ensure sustained reliable network bandwidth between your EC2 instance and your EBS volumes is to launch the EC2 instance as EBS-optimized or choose an instance type with 10 Gigabit network connectivity. To see which instance types include 10 Gigabit network connectivity, see [Amazon EC2 Instance Types](#). For information about configuring EBS-optimized instances, see [Amazon EBS-Optimized Instances](#).

Choose an EC2 Instance with Enough Bandwidth

Launching an instance that is EBS-optimized provides you with a dedicated connection between your EC2 instance and your EBS volume. However, it is still possible to provision EBS volumes that exceed the available bandwidth for certain instance types, especially when multiple volumes are striped in a RAID configuration. For information about the instance types available to be launched as EBS-optimized, the dedicated throughput to these instance types, the dedicated bandwidth to Amazon EBS, the maximum amount of IOPS the instance can support if you are using a 16 KB I/O size, and the approximate I/O bandwidth available on that connection, see [Instance Types that Support EBS Optimization \(p. 813\)](#).

Be sure to choose an EBS-optimized instance that provides more dedicated EBS throughput than your application needs; otherwise, the Amazon EBS to Amazon EC2 connection becomes a performance bottleneck.

Note that some instances with 10-gigabit network interfaces do not offer EBS-optimization, and therefore do not have dedicated EBS bandwidth available. However, you can use all of that bandwidth for traffic to Amazon EBS if your application isn't pushing other network traffic that contends with Amazon EBS. Some 10-gigabit network instances offer dedicated Amazon EBS bandwidth in addition to a 10-gigabit interface which is used exclusively for network traffic.

If an instance type has a maximum 16 KB IOPS value of 4,000, that value is an absolute best-case scenario and is not guaranteed unless the instance is launched as EBS-optimized. To consistently achieve the best performance, you must launch instances as EBS-optimized. However, if you attach a 4,000 IOPS `io1` volume to an EBS-optimized instance with a 16 KB IOPS value of 4,000, the Amazon EC2 to Amazon EBS connection bandwidth limit prevents this volume from providing the 500 MB/s maximum aggregate throughput available to it. In this case, we must use an EBS-optimized EC2 instance that supports at least 500 MB/s of throughput.

Volumes of type General Purpose SSD (`gp2`) have a throughput limit between 128 MB/s and 160 MB/s per volume (depending on volume size), which pairs well with a 1,000 Mbps EBS-optimized connection. Instance types that offer more than 1,000 Mbps of throughput to Amazon EBS can use more than one `gp2` volume to take advantage of the available throughput. Volumes of type Provisioned IOPS SSD (`io1`) have a throughput limit range of 256 KiB for each IOPS provisioned, up to a maximum of 500 MiB/s (at 32,000 IOPS). For more information, see [Amazon EBS Volume Types \(p. 735\)](#).

Instance types with 10 Gigabit network connectivity support up to 800 MB/s of throughput and 48,000 16K IOPS for unencrypted Amazon EBS volumes and up to 25,000 16K IOPS for encrypted Amazon EBS volumes. Because the maximum `io1` value for EBS volumes is 32,000 for `io1` volumes and 10,000 for `gp2` volumes, you can use several EBS volumes simultaneously to reach the level of I/O performance available to these instance types. For more information about which instance types include 10 Gigabit network connectivity, see [Amazon EC2 Instance Types](#).

You should use EBS-optimized instances when available to get the full performance benefits of Amazon EBS `gp2` and `io1` volumes. For more information, see [Amazon EBS-Optimized Instances \(p. 812\)](#).

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`)—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 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 I/O operations at 32 KiB each count as 8 operations. 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` 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 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 160 MiB/s. If you are using a 256 KiB I/O size, your volume reaches its throughput limit at 640 IOPS ($640 \times 256 \text{ KiB} = 160 \text{ MiB}$). For smaller I/O sizes (such as 16 KiB), this same volume can sustain 3,000 IOPS because the throughput is well below 160 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 \(p. 735\)](#).

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 EC2 Instance Configuration \(p. 827\)](#). 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 with CloudWatch

You can monitor these I/O characteristics with each volume's [CloudWatch metrics \(p. 754\)](#). Important metrics to consider include:

- `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 credits (for `gp2` volumes) or volume throughput credits (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 may 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` volume with more provisioned IOPS to achieve faster latencies.

For more information about Amazon EBS I/O characteristics, see the [Amazon EBS: Designing for Performance](#) re:Invent presentation on this topic.

Initializing Amazon EBS Volumes

New 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 restored 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. For most applications, amortizing this cost over the lifetime of the volume is acceptable. Performance is restored after the data is accessed once.

You can avoid this performance hit in a production environment by reading from all of the blocks on your volume before you use it; this process is called *initialization*. For a new volume created from a snapshot, you should read all the blocks that have data before using the volume.

Important

While initializing `io1` 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` volumes while you are initializing them. For more information, see [Monitoring Volumes with Status Checks \(p. 759\)](#).

Initializing Amazon EBS Volumes on Linux

New 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 restored 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.

To initialize a volume restored 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 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 \(p. 751\)](#).

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 \(p. 832\)](#)
- [Creating a RAID Array on Linux \(p. 833\)](#)
- [Creating Snapshots of Volumes in a RAID Array \(p. 835\)](#)

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.	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 MB/s of throughput or a 500 GiB RAID 1 array with an available bandwidth of 4,000 IOPS and 500 MB/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 \(p. 747\)](#).

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 EC2 Instance Configuration \(p. 827\)](#).

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 \(p. 750\)](#).
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 snapshots of these volumes are created independently, not as a whole. Restoring 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, stop applications from writing to the RAID array and flush all caches to disk. To stop writes to the RAID array, you can take steps such as stopping the applications, stopping the instance, or unmounting the RAID array. After you've stopped all I/O activity, you can create the snapshots.

When restoring the EBS volumes in a RAID array from a set of snapshots, stop all I/O activity as you did when you created the snapshots and then restore the volumes from the snapshots.

Benchmark EBS Volumes

This section demonstrates how 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 described in this topic will 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 options between 425 and 14,000 Mbps, depending on the instance type.

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 \(p. 812\)](#).

Setting up Provisioned IOPS SSD (io1) volumes

To create an io1 volume, choose **Provisioned IOPS SSD** when creating the volume using the Amazon EC2 console, or, at the command line, specify **--type io1 --iops n** where *n* is an integer between 100 and 32000. For information about creating EBS volumes, see [Creating an Amazon EBS Volume \(p. 747\)](#). For information about attaching these volumes to your instance, see [Attaching an Amazon EBS Volume to an Instance \(p. 750\)](#).

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 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_volo && sudo mkfs.xfs /dev/md0
[ec2-user ~]$ sudo mount -t xfs /dev/md0 /mnt/p_iops_volo
[ec2-user ~]$ sudo chown ec2-user:ec2-user /mnt/p_iops_volo/
```

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 \(p. 747\)](#). For information about attaching these volumes to your instance, see [Attaching an Amazon EBS Volume to an Instance \(p. 750\)](#).

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 easy 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 with 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 \(p. 837\)](#).

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	For benchmarking I/O performance. (Note that fio has a dependency on libaio-devel .)

Tool	Description
	To install fio on Amazon Linux, run the following command: <pre>[ec2-user ~]\$ sudo yum install -y fio</pre>
	To install fio on Ubuntu, run the following command: <pre>sudo apt-get install -y fio</pre>
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 500 IOPS available (baseline for gp2 volumes and the provisioned amount for io1 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 1,000 provisioned IOPS should target a queue length of 2. 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.

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 restored from snapshots, be sure to initialize them before benchmarking. For more information, see [Initializing Amazon EBS Volumes \(p. 830\)](#).

When you are finished testing your volumes, see the following topics for help cleaning up: [Deleting an Amazon EBS Volume \(p. 769\)](#) and [Terminate Your Instance \(p. 421\)](#).

Benchmarking io1 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 \(p. 826\)](#).

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 the [Inspecting disk I/O performance with fio](#) tutorial.

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 Events for Amazon EBS

Amazon EBS emits notifications based on Amazon CloudWatch Events for a variety of snapshot and encryption status changes. With CloudWatch Events, you can establish rules that trigger programmatic actions in response to a change in 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.

For more information, see [Using Events](#) in the *Amazon CloudWatch User Guide*.

EBS Volume Events

This section defines the supported Amazon EBS volume events and provides examples of event output for specific scenarios. Events in CloudWatch are represented as JSON objects. For more information about the format and content of event objects, see [Events and Event Patterns](#) in the *Amazon CloudWatch Events User Guide*.

Note

Additional information about EBS volumes that is not captured by Cloudwatch is available through the [DescribeVolumes API](#) and the [describe-volumes CLI command](#).

The fields that are unique to EBS events are contained in the "detail" section of the JSON objects shown below. The "event" field contains the event name. The "result" field contains the completed status of the action that triggered the event.

Create Volume (`createVolume`)

The `createVolume` event is sent to your AWS account when an action to create a volume completes. 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. 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. 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/01234567-0123-0123-0123456789ab"
    ],
}
```

```
"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

Create Snapshot (`createSnapshot`)

The `createSnapshot` event is sent to your AWS account when an action to create a 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 `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": "",  
        "start-time": "2015-06-10T16:45:00Z",  
        "end-time": "2015-06-10T16:45:05Z"  
    }  
}
```

```
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": "arn:aws:ec2::us-west-2:volume/vol-01234567",
    "StartTime": "yyyy-mm-ddThh:mm:ssZ",
    "EndTime": "yyyy-mm-ddThh:mm:ssZ" }
```

Copy Snapshot (copySnapshot)

The copySnapshot event is sent to your AWS account when an action to copy a 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 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 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": "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. The result is always succeeded.

Event Data

The listing below 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": "01234567-01234-0123-0123-012345678901",  
    "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"  
    }  
}
```

Using Amazon 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**.

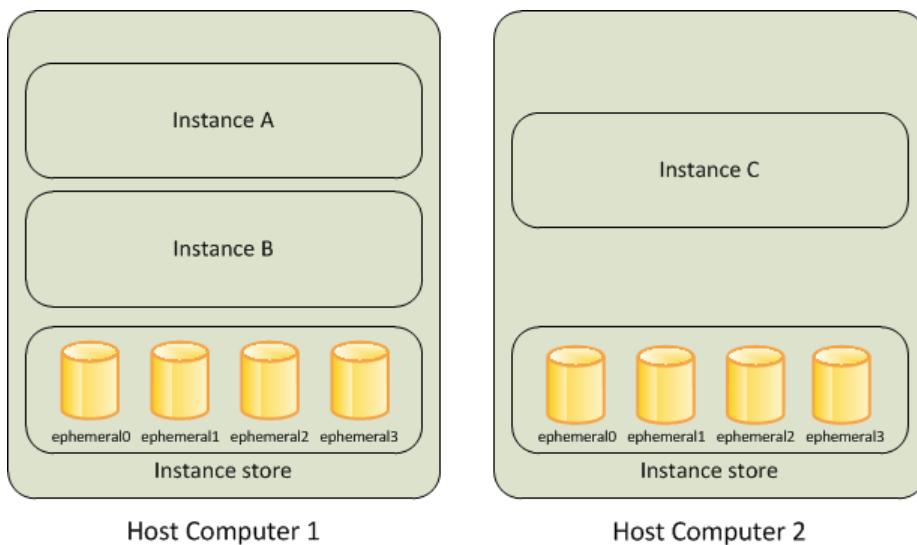
Your rule should now appear on the **Rules** tab. In the example shown, the event that you configured should be emitted by EBS the next time you copy a snapshot.

Amazon 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. While an instance store is dedicated to a particular instance, the disk subsystem is shared among instances on a host computer.

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 \(p. 848\)](#)
- [Instance Store Volumes \(p. 848\)](#)
- [Add Instance Store Volumes to Your EC2 Instance \(p. 852\)](#)
- [SSD Instance Store Volumes \(p. 854\)](#)
- [Instance Store Swap Volumes \(p. 856\)](#)
- [Optimizing Disk Performance for Instance Store Volumes \(p. 858\)](#)

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.

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 \(p. 852\)](#).

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 \(p. 854\)](#).

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 (1,680 GB)	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	✓	
c5d.large	1 x 50 GB	NVMe SSD		✓
c5d.xlarge	1 x 100 GB	NVMe SSD		✓
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.18xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓
cc2.8xlarge	4 x 840 GB (3,360 GB)	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		✓
f1.16xlarge	4 x 940 GB	NVMe SSD		✓
g2.2xlarge	1 x 60 GB	SSD	✓	
g2.8xlarge	2 x 120 GB (240 GB)	SSD	✓	

Amazon Elastic Compute Cloud
User Guide for Linux Instances
Instance Store Volumes

Instance Type	Instance Store Volumes	Type	Needs Initialization*	TRIM Support**
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 (1,600 GB)	SSD		✓
i2.4xlarge	4 x 800 GB (3,200 GB)	SSD		✓
i2.8xlarge	8 x 800 GB (6,400 GB)	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		✓
m1.small	1 x 160 GB†	HDD	✓	
m1.medium	1 x 410 GB	HDD	✓	
m1.large	2 x 420 GB (840 GB)	HDD	✓	
m1.xlarge	4 x 420 GB (1,680 GB)	HDD	✓	
m2.xlarge	1 x 420 GB	HDD	✓	
m2.2xlarge	1 x 850 GB	HDD	✓	
m2.4xlarge	2 x 840 GB (1,680 GB)	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	✓	
m5d.large	1 x 75 GB	NVMe SSD		✓
m5d.xlarge	1 x 150 GB	NVMe SSD		✓
m5d.2xlarge	1 x 300 GB	NVMe SSD		✓

Instance Type	Instance Store Volumes	Type	Needs Initialization*	TRIM Support**
m5d.4xlarge	2 x 300 GB	NVMe SSD		✓
m5d.12xlarge	2 x 900 GB	NVMe SSD		✓
m5d.24xlarge	4 x 900 GB (3.6 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		✓
r5d.large	1 x 75 GB	NVMe SSD		✓
r5d.xlarge	1 x 150 GB	NVMe SSD		✓
r5d.2xlarge	1 x 300 GB	NVMe SSD		✓
r5d.4xlarge	2 x 300 GB (600 GB)	NVMe SSD		✓
r5d.12xlarge	2 x 900 GB (1,800 GB)	NVMe SSD		✓
r5d.24xlarge	4 x 900 GB (3,600 GB)	NVMe SSD		✓
x1.16xlarge	1 x 1,920 GB	SSD		
x1.32xlarge	2 x 1,920 GB (3,840 GB)	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		
x1e.32xlarge	2 x 1,920 GB (3,840 GB)	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 (1,800 GB)	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 \(p. 858\)](#).

** For more information, see [Instance Store Volume TRIM Support \(p. 855\)](#).

† 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 \(p. 856\)](#).

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 \(p. 866\)](#).

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 \(p. 84\)](#). 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 \(p. 733\)](#).

You can specify the instance store volumes for your instance only when you launch an instance. You can't attach instance store volumes to an instance after you've launched it.

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. For more information about the instance store volumes support by each instance type, see [Instance Store Volumes \(p. 848\)](#). If the instance type you choose for your instance supports instance store volumes, you must add them to the block device mapping for the instance when you launch it. After you launch the 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 \(p. 852\)](#)
- [Adding Instance Store Volumes to an Instance \(p. 853\)](#)
- [Making Instance Store Volumes Available on Your Instance \(p. 854\)](#)

Adding Instance Store Volumes to an AMI

You can create an AMI with a block device mapping that includes instance store volumes. After you add instance store volumes to an AMI, any instance that you launch from the AMI includes these instance store volumes. When you launch an instance, you can omit volumes specified in the AMI block device mapping and add new volumes.

Important

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.

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 \(p. 864\)](#).) 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.

Important

For M3 instances, you might receive instance store volumes even if you do not specify them in the block device mapping for the instance.

Important

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.

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 \(p. 855\)](#). 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 \(p. 874\)](#).

For Windows instances, you can also view the instance store volumes using Windows Disk Management. For more information, see [Listing the 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.
2. Use the `df -h` command to view the volumes that are formatted and mounted. Use the `lsblk` to view any volumes that were mapped at launch but not formatted and mounted.
3. 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.
 - b. Create a directory on which to mount the device using the `mkdir` command.
 - c. Mount the device on the newly created directory using the `mount` command.

SSD Instance Store Volumes

The following instances support instance store volumes that use solid state drives (SSD) to deliver high random I/O performance: C3, C5d, F1, G2, I2, I3, M3, M5d, R3, X1, and z1d. For more information about the instance store volumes support by each instance type, see [Instance Store Volumes \(p. 848\)](#).

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 \(p. 852\)](#).

NVMe SSD Volumes

The following instances offer non-volatile memory express (NVMe) SSD instance store volumes: C5d, I3, F1, M5d, and z1d. To access the NVMe volumes, you must use an operating system that supports NVMe. The following are the recommended operating systems:

- The most recent version of Amazon Linux 2 or Amazon Linux AMI
- Ubuntu version 16.04 or latest LTS release provided by AWS. If you are using a different version, we recommend that you turn off memory hot add.
- Red Hat Enterprise Linux version 7.4
- CentOS 7 version 1708_11
- SUSE Linux Enterprise Server 12 SP2 or later
- FreeBSD 11.1 or later (does not support F1 instances)

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 **lsmod** command.

```
[ec2-user ~]$ lsmod | grep nvme
nvme           48813   0
```

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.

Instance Store Volume TRIM Support

The following instances support SSD volumes with TRIM: C5d, F1, I2, I3, M5d, R3, R5d, and z1d.

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 \(p. 228\)](#).

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 \(p. 848\)](#).

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 -
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 \(p. 857\)](#). If you stop and restart an instance that has been configured to use instance store swap space, you must repeat [Step 1 \(p. 857\)](#) through [Step 5 \(p. 857\)](#) 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 most 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 \(p. 848\)](#).

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
```

Amazon Elastic File System (Amazon EFS)

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 \(p. 859\)](#)
- [Step 1: Create an EFS File System \(p. 859\)](#)
- [Step 2: Mount the File System \(p. 860\)](#)
- [Step 3: Test the File System \(p. 861\)](#)
- [Step 4: Clean Up \(p. 861\)](#)

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 file system \(p. 544\)](#), and [Security Groups for Amazon EC2 Instances and Mount Targets](#) 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 \(p. 21\)](#).

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. On the **Configure file system access** page, do the following:
 - a. For **VPC**, select the VPC to use for your instances.
 - b. For **Create mount targets**, select all the Availability Zones.
 - c. For each Availability Zone, ensure that the value for **Security group** is the security group that you created in [Prerequisites \(p. 859\)](#).
 - d. Choose **Next Step**.
4. On the **Configure optional settings** page, do the following:
 - a. For the tag with **Key=Name**, type a name for the file system in **Value**.
 - b. For **Choose performance mode**, keep the default option, **General Purpose**.
 - c. Choose **Next Step**.
5. On the **Review and create** page, choose **Create File System**.
6. After the file system is created, note the file system ID, as you'll use it later in this tutorial.

Step 2: Mount the File System

Use the following procedure to launch two t2.micro instances. The user data script mounts the file system to both instances during launch and updates /etc/fstab to ensure that the file system is remounted after an instance reboot. 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. On the **Choose an Amazon Machine Image** page, select an Amazon Linux AMI with the HVM virtualization type.
4. On the **Choose an Instance Type** page, keep the default instance type, t2.micro and choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, do the following:
 - a. For **Number of instances**, type 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. Under **Advanced Details**, select **As text**, and paste the following script into **User data**. Update **FILE_SYSTEM_ID** with the ID of your file system. You can optionally update **MOUNT_POINT** with a directory for your mounted file system.

```
#!/bin/bash
yum update -y
yum install -y nfs-utils
FILE_SYSTEM_ID=fs-xxxxxxxx
AVAILABILITY_ZONE=$(curl -s http://169.254.169.254/latest/meta-data/placement/
availability-zone )
REGION=${AVAILABILITY_ZONE:0:-1}
MOUNT_POINT=/mnt/efs
mkdir -p ${MOUNT_POINT}
chown ec2-user:ec2-user ${MOUNT_POINT}
echo ${FILE_SYSTEM_ID}.efs.${REGION}.amazonaws.com:/ ${MOUNT_POINT} nfs4
    nfsvers=4.1,rsize=1048576,wszie=1048576,hard,timeo=600,retrans=2,_netdev 0 0 >> /
etc/fstab
mount -a -t nfs4
```

- 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 \(p. 859\)](#), and 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.

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 \(p. 397\)](#).
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, type the ID of the file system and choose **Delete File System**.

Amazon Simple Storage Service (Amazon S3)

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 easy 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 \(p. 731\)](#).

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 HTTP URL address. For example, if an object with a key value /photos/mygarden.jpg is stored in the myawsbucket bucket, then it is addressable using the URL <http://myawsbucket.s3.amazonaws.com/photos/mygarden.jpg>.

For more information about the features of Amazon S3, see the [Amazon S3 product page](#).

Amazon S3 and Amazon EC2

Given the benefits of Amazon S3 for storage, you may 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.

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

- [Linux-Specific Volume Limits \(p. 863\)](#)
- [Windows-Specific Volume Limits \(p. 864\)](#)
- [Instance Type Limits \(p. 864\)](#)
- [Bandwidth versus Capacity \(p. 864\)](#)

Linux-Specific Volume Limits

Attaching more than 40 volumes can cause boot failures. Note that 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.

Windows-Specific Volume Limits

The following table shows the volume limits for Windows instances based on the driver used. Note that these numbers include the root volume, plus any attached instance store volumes and EBS volumes.

Important

Attaching more than the following volumes to a Windows instance is supported on a best effort basis only and is not guaranteed.

Driver	Volume Limit
AWS PV	26
Citrix PV	26
Red Hat PV	17

We do not recommend that you give a Windows instance more than 26 volumes with AWS PV or Citrix PV drivers, as it is likely to cause performance issues.

To determine which PV drivers your instance is using, or to upgrade your Windows instance from Red Hat to Citrix PV drivers, see [Upgrading PV Drivers on Your Windows Instance](#).

For more information about how device names related to volumes, see [Mapping Disks to Volumes on Your Windows EC2 Instance](#) in the *Amazon EC2 User Guide for Windows Instances*.

Instance Type Limits

C5, C5d, M5, M5d, R5, R5d, and z1d instances support a maximum of 28 attachments, and every instance has at least one network interface attachment. If you have no additional network interface attachments on these instances, you could attach 27 EBS volumes. For more information, see [Elastic Network Interfaces \(p. 683\)](#).

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 EC2 Instance Configuration \(p. 827\)](#) 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.

Contents

- [Available Device Names \(p. 865\)](#)
- [Device Name Considerations \(p. 865\)](#)

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

The following table lists the available device names for Linux instances. The number of volumes that you can attach to your instance is determined by the operating system. For more information, see [Instance Volume Limits \(p. 863\)](#).

Virtualization Type	Available	Reserved for Root	Recommended for EBS Volumes	Instance Store Volumes	NVMe 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] /dev/sd[b-y](hs1.8xlarge)	Not available
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-e] /dev/sd[b-h](h1.16xlarge) /dev/sd[b-y](d2.8xlarge) /dev/sd[b-y](hs1.8xlarge) /dev/sd[b-i](i2.8xlarge)	/dev/nvme[0-26]n1 *

* NVMe instance store volumes are automatically enumerated and assigned a device name.

For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 847\)](#).

For more information about NVMe EBS volumes, see [Amazon EBS and NVMe \(p. 823\)](#).

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.
- 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), even 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.
- The number of NVMe instance store volumes for an instance depends on the size of the instance. The device names are `/dev/nvme0n1`, `/dev/nvme1n1`, and so on.
- 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. Some instance types support both PV and HVM, some support HVM only, and others support PV only. 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 \(p. 86\)](#).
- 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`).
- Hardware virtual machine (HVM) AMIs don't support the use of trailing numbers on device names, except for the device name that's reserved for the root device.
- 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, 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 \(p. 750\)](#). However, the only way to attach instance store volumes to an instance is to use block device mapping to attach them as the instance is launched.

For more information about root device volumes, see [Changing the Root Device Volume to Persist \(p. 16\)](#).

Contents

- [Block Device Mapping Concepts \(p. 866\)](#)
- [AMI Block Device Mapping \(p. 869\)](#)
- [Instance Block Device Mapping \(p. 871\)](#)

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 \(p. 867\)](#)
- [Block Device Mapping Instance Store Caveats \(p. 867\)](#)
- [Example Block Device Mapping \(p. 868\)](#)
- [How Devices Are Made Available in the Operating System \(p. 869\)](#)

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, and the name assigned can be different from the name that Amazon EC2 recommends. For more information, see [Device Naming on Linux Instances \(p. 864\)](#).
- [Instance store volumes] The virtual device: `ephemeral[0-23]`. Note that the number and size of available instance store volumes for your instance varies by instance type.
- [NVMe instance store volumes] These volumes are automatically enumerated and assigned a device name; including them in your block device mapping has no effect.
- [EBS volumes] The ID of the snapshot to use to create the block device (`snap-xxxxxxx`). This value is optional as long as you specify a volume size.
- [EBS volumes] The size of the volume, in GiB. The specified size must be greater than or equal to the size of the specified snapshot.
- [EBS volumes] 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.
- [EBS volumes] The volume type, which can be `gp2` for General Purpose SSD, `io1` for Provisioned IOPS SSD, `st1` for Throughput Optimized HDD, `sc1` for Cold HDD, or `standard` for Magnetic. The default value is `gp2` in the Amazon EC2 console, and `standard` in the AWS SDKs and the AWS CLI.
- [EBS volumes] 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.

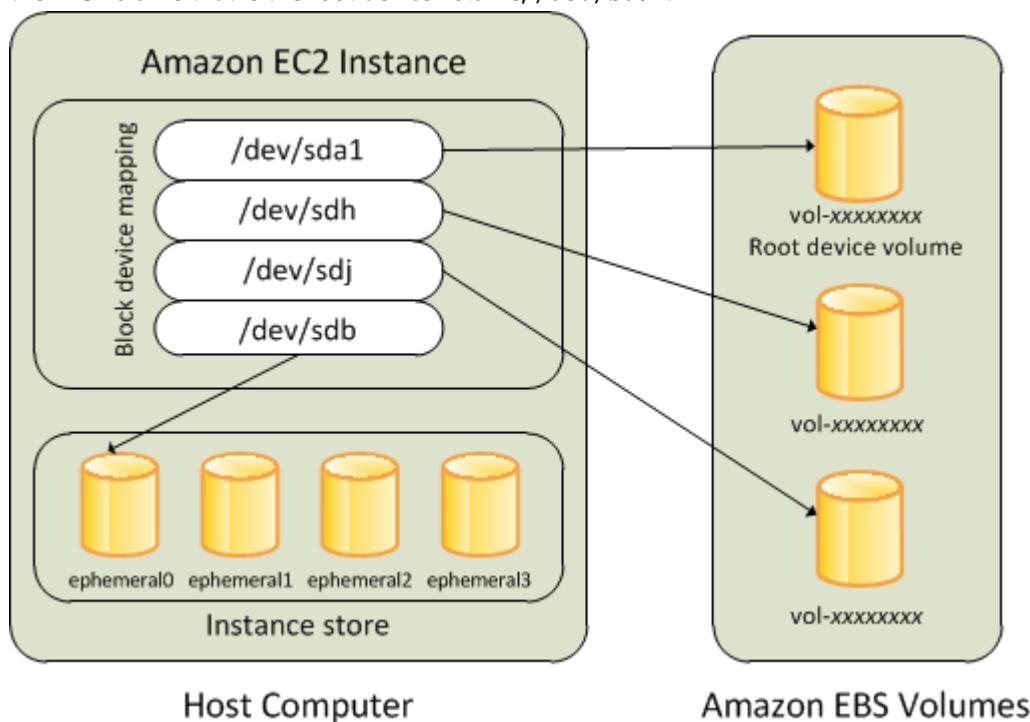
Note

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 ephemeral0 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 \(p. 869\)](#) and [Updating the Block Device Mapping when Launching an Instance \(p. 871\)](#).

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 \(p. 847\)](#). 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 \(p. 750\)](#).

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 \(p. 869\)](#)
- [Viewing the EBS Volumes in an AMI Block Device Mapping \(p. 870\)](#)

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 following parameter:

```
--block-device-mappings [mapping, ...]
```

To add an instance store volume, use the following mapping:

```
{  
    "DeviceName": "/dev/sdf",  
    "VirtualName": "ephemeral0"  
}
```

To add an empty 100 GiB Magnetic 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. (Recall that 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.

Limits

- 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 \(p. 871\)](#)
- [Updating the Block Device Mapping of a Running Instance \(p. 873\)](#)
- [Viewing the EBS Volumes in an Instance Block Device Mapping \(p. 873\)](#)
- [Viewing the Instance Block Device Mapping for Instance Store Volumes \(p. 874\)](#)

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 command line

Use the [run-instances](#) AWS CLI command to specify a block device mapping for an instance.

Specify the block device mapping using the following parameter:

```
--block-device-mappings [mapping, ...]
```

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 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.

```
{  
    "DeviceName": "/dev/sdc",  
}
```

```
        "VirtualName": "ephemeral1"
    }
```

Alternatively, you can 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 following [modify-instance-attribute](#) AWS CLI command to update the block device mapping of a running instance. Note that 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 bar, type **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. (Recall that this dialog box 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, select 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 complete block device mapping. The base URI for all requests for instance metadata is <http://169.254.169.254/latest/>.

First, connect to your running instance. From the instance, use this query to get its block device mapping.

```
[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.

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

For more information, see [Instance Metadata and User Data \(p. 454\)](#).

Using Public Data Sets

Amazon Web Services provides a repository of public data sets that can be seamlessly integrated into AWS cloud-based applications. Amazon stores the data sets at no charge to the community and, as with all AWS services, you pay only for the compute and storage you use for your own applications.

Contents

- [Public Data Set Concepts \(p. 875\)](#)
- [Finding Public Data Sets \(p. 875\)](#)
- [Creating a Public Data Set Volume from a Snapshot \(p. 876\)](#)
- [Attaching and Mounting the Public Data Set Volume \(p. 876\)](#)

Public Data Set Concepts

Previously, large data sets such as the mapping of the Human Genome and the US Census data required hours or days to locate, download, customize, and analyze. Now, anyone can access these data sets from an EC2 instance and start computing on the data within minutes. You can also leverage the entire AWS ecosystem and easily collaborate with other AWS users. For example, you can produce or use prebuilt server images with tools and applications to analyze the data sets. By hosting this important and useful data with cost-efficient services such as Amazon EC2, AWS hopes to provide researchers across a variety of disciplines and industries with tools to enable more innovation, more quickly.

For more information, go to the [AWS Public Datasets](#) page.

Available Public Data Sets

Public data sets are currently available in the following categories:

- **Biology**—Includes Human Genome Project, GenBank, and other content.
- **Chemistry**—Includes multiple versions of PubChem and other content.
- **Economics**—Includes census data, labor statistics, transportation statistics, and other content.
- **Encyclopedic**—Includes Wikipedia content from multiple sources and other content.

Finding Public Data Sets

Before you can use a public data set, you must locate the data set and determine which format the data set is hosted in. The data sets are available in two possible formats: Amazon EBS snapshots or Amazon S3 buckets.

To find a public data set and determine its format

1. Go to the [AWS Public Datasets](#) page to see a listing of all available public data sets. You can also enter a search phrase on this page to query the available public data set listings.
2. Click the name of a data set to see its detail page.
3. On the data set detail page, look for a snapshot ID listing to identify an Amazon EBS formatted data set or an Amazon S3 URL.

Data sets that are in snapshot format are used to create new EBS volumes that you attach to an EC2 instance. For more information, see [Creating a Public Data Set Volume from a Snapshot \(p. 876\)](#).

For data sets that are in Amazon S3 format, you can use the AWS SDKs or the HTTP query API to access the information, or you can use the AWS CLI to copy or synchronize the data to and from your instance. For more information, see [Amazon S3 and Amazon EC2 \(p. 862\)](#).

You can also use Amazon EMR to analyze and work with public data sets. For more information, see [What is Amazon EMR?](#).

Creating a Public Data Set Volume from a Snapshot

To use a public data set that is in snapshot format, you create a new volume, specifying the snapshot ID of the public data set. You can create your new volume using the AWS Management Console as follows. If you prefer, you can use the `create-volume` AWS CLI command instead.

To create a public data set volume from a snapshot

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region that your data set snapshot is located in.

If you need to create this volume in a different region, you can copy the snapshot to that region and then use it to create a volume in that region. For more information, see [Copying an Amazon EBS Snapshot \(p. 799\)](#).

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 \(p. 735\)](#).
6. For **Snapshot**, start typing the ID or description of the snapshot that has the data set, and choose it from the list.

If the snapshot that you are expecting to see does not appear, you might not have selected the region it is in. If the data set you identified in [Finding Public Data Sets \(p. 875\)](#) does not specify a region on its detail page, it is likely contained in the us-east-1 US East (N. Virginia) region.

7. For **Size (GiB)**, type the size of the volume, or verify that the default size of the snapshot is adequate.

Note

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**.

8. With a Provisioned IOPS SSD volume, for **IOPS**, type the maximum number of input/output operations per second (IOPS) that the volume should support.
9. For **Availability Zone**, choose the Availability Zone in which to create the volume. EBS volumes can only be attached to instances in the same Availability Zone.
10. (Optional) Choose **Create additional tags** to add tags to the volume. For each tag, provide a tag key and a tag value.
11. Choose **Create Volume**.

Attaching and Mounting the Public Data Set Volume

After you have created your new data set volume, you need to attach it to an EC2 instance to access the data (this instance must also be in the same Availability Zone as the new volume). For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 750\)](#).

After you have attached the volume to an instance, you need to mount the volume on the instance. For more information, see [Making an Amazon EBS Volume Available for Use on Linux \(p. 751\)](#).

If you restored a snapshot to a larger volume than the default for that snapshot, you must extend the file system on the volume to take advantage of the extra space. For more information, see [Modifying the Size, IOPS, or Type of an EBS Volume on Linux \(p. 769\)](#).

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 \(p. 878\)](#)
- [Resource IDs \(p. 879\)](#)
- [Listing and Filtering Your Resources \(p. 884\)](#)
- [Tagging Your Amazon EC2 Resources \(p. 887\)](#)
- [Amazon EC2 Service Limits \(p. 896\)](#)
- [Amazon EC2 Usage Reports \(p. 898\)](#)

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 (p. 521).</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. 139) .

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. 799) .
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. 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

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. Starting in January 2016, we're gradually introducing longer length IDs for Amazon EC2 and Amazon EBS resource types. The length of the alphanumeric character combination was in an 8-character format; the new IDs are in a 17-character format, for example, `i-1234567890abcdef0` for an instance ID.

Supported resource types have an opt-in period, during which you can choose a resource ID format, and a deadline date, after which the resource defaults to the longer ID format. After the deadline has passed for a specific resource type, you can no longer disable the longer ID format for that resource type.

Different resource types have different opt-in periods and deadline dates. The following table lists the supported resource types, along with their opt-in periods and deadline dates.

Resource type	Opt-in period	Deadline date
<code>instance snapshot reservation volume</code>	No longer available	December 15, 2016
<code>bundle conversion-task customer-gateway dhcp-options elastic-ip-allocation </code>	February 09, 2018 - June 30, 2018	June 30, 2018

Resource type	Opt-in period	Deadline date
<code>elastic-ip-association export-task flow-log image import-task </code> <code>internet-gateway network-acl network-acl-association </code> <code>network-interface network-interface-attachment prefix-list </code> <code>route-table route-table-association security-group subnet </code> <code>subnet-cidr-block-association vpc vpc-cidr-block-association vpc-endpoint </code> <code>vpc-peering-connection vpn-connection vpn-gateway</code>		

During the Opt-in Period

You can enable or disable longer IDs for a resource at any time during the opt-in period. After you've enabled longer IDs for a resource type, any new resources that you create are created with a longer ID.

Note

A resource ID does not change after it's created. Therefore, enabling or disabling longer IDs during the opt-in period does not affect your existing resource IDs.

Depending on when you created your AWS account, supported resource types may default to using longer IDs. However, you can opt out of using longer IDs until the deadline date for that resource type. For more information, see [Longer EC2 and EBS Resource IDs](#) in the *Amazon EC2 FAQs*.

After the Deadline Date

You can't disable longer IDs for a resource type after its deadline date has passed. Any new resources that you create are created with a longer ID.

Working with Longer IDs

You can enable or disable longer IDs per IAM user and IAM role. By default, an IAM user or role defaults to the same settings as the root user.

Topics

- [Viewing Longer ID Settings \(p. 880\)](#)
- [Modifying Longer ID Settings \(p. 881\)](#)

Viewing Longer ID Settings

You can use the console and command line tools to view the resource types that support longer IDs.

To view your longer ID settings using the 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 the region for which to view your longer ID settings.
3. From the dashboard, under **Account Attributes**, choose **Resource ID length management**.

4. Expand **Advanced Resource ID Management** to view the resource types that support longer IDs and their deadline dates.

To view your longer ID settings using the command line

Use one of the following commands:

- [describe-id-format \(AWS CLI\)](#)

```
aws ec2 describe-id-format --region region
```

- [Get-EC2IdFormat \(AWS Tools for Windows PowerShell\)](#)

```
Get-EC2IdFormat -Region region
```

To view longer ID settings for a specific IAM user or IAM role using the command line

Use one of the following commands and specify the ARN of an IAM user, IAM role, or root account user in the request.

- [describe-identity-id-format \(AWS CLI\)](#)

```
aws ec2 describe-identity-id-format --principal-arn arn-of-iam-principal --region region
```

- [Get-EC2IdentityIdFormat \(AWS Tools for Windows PowerShell\)](#)

```
Get-EC2IdentityIdFormat -PrincipalArn arn-of-iam-principal -Region region
```

To view the aggregated longer ID settings for a specific region using the command line

Use the [describe-aggregate-id-format](#) AWS CLI command to view the aggregated longer ID setting for the entire region, as well as the aggregated longer ID setting of all ARNs for each resource type. This command is useful for performing a quick audit to determine whether a specific region is fully opted in for longer IDs.

```
aws ec2 describe-aggregate-id-format --region region
```

To identify users who have explicitly defined custom longer ID settings

Use the [describe-principal-id-format](#) AWS CLI command to view the longer ID format settings for the root user and all IAM roles and IAM users that have explicitly specified a longer ID preference. This command is useful for identifying IAM users and IAM roles that have overridden the default longer ID settings.

```
aws ec2 describe-principal-id-format --region region
```

Modifying Longer ID Settings

You can use the console and command line tools to modify longer ID settings for resource types that are still within their opt-in period.

Note

The AWS CLI and AWS Tools for Windows PowerShell commands in this section are per-region only. They apply to the default region unless otherwise specified. To modify the settings for other regions, include the `region` parameter in the command.

To modify longer ID settings using the 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 the region for which to modify the longer ID settings.
3. From the dashboard, under **Account Attributes**, choose **Resource ID length management**.
4. Do one of the following:
 - To enable longer IDs for all supported resource types for all IAM users across all regions, choose **Switch to longer IDs, Yes, switch to longer IDs**.

Important

IAM users and IAM roles need the `ec2:ModifyIdentityIdFormat` permission to perform this action.

- To modify longer ID settings for a specific resource type for your IAM user account, expand **Advanced Resource ID Management**, and then select the corresponding check box in the **My IAM Role/User** column to enable longer IDs, or clear the check box to disable longer IDs.
- To modify longer ID settings for a specific resource type for all IAM users, expand **Advanced Resource ID Management**, and then select the corresponding check box in the **All IAM Roles/Users** column to enable longer IDs, or clear the check box to disable longer IDs.

To modify longer ID settings for your IAM user account using the command line

Use one of the following commands:

Note

If you're using these commands as the root user, then changes apply to the entire AWS account, unless an IAM user or role explicitly overrides these settings for themselves.

- **modify-id-format** (AWS CLI)

```
aws ec2 modify-id-format --resource resource_type --use-long-ids
```

You can also use the command to modify the longer ID settings for all supported resource types. To do this, replace the `resource_type` parameter with `all-current`.

```
aws ec2 modify-id-format --resource all-current --use-long-ids
```

Note

To disable longer IDs, replace the `use-long-ids` parameter with `no-use-long-ids`.

- **Edit-EC2IdFormat** (AWS Tools for Windows PowerShell)

```
Edit-EC2IdFormat -Resource resource_type -UseLongId boolean
```

You can also use the command to modify the longer ID settings for all supported resource types. To do this, replace the `resource_type` parameter with `all-current`.

```
Edit-EC2IdFormat -Resource all-current -UseLongId boolean
```

To modify longer ID settings for a specific IAM user or IAM role using the command line

Use one of the following commands and specify the ARN of an IAM user, IAM role, or root user in the request.

- [modify-identity-id-format \(AWS CLI\)](#)

```
aws ec2 modify-identity-id-format --principal-arn arn-of-iam-principal --  
resource resource_type --use-long-ids
```

You can also use the command to modify the longer ID settings for all supported resource types. To do this, specify `all-current` for the `--resource` parameter.

```
aws ec2 modify-identity-id-format --principal-arn arn-of-iam-principal --resource all-  
current --use-long-ids
```

Note

To disable longer IDs, replace the `use-long-ids` parameter with `no-use-long-ids`.

- [Edit-EC2IdentityIdFormat \(AWS Tools for Windows PowerShell\)](#)

```
Edit-EC2IdentityIdFormat -PrincipalArn arn-of-iam-principal -Resource resource_type -  
UseLongId boolean
```

You can also use the command to modify the longer ID settings for all supported resource types. To do this, specify `all-current` for the `-Resource` parameter.

```
Edit-EC2IdentityIdFormat -PrincipalArn arn-of-iam-principal -Resource all-current -  
UseLongId boolean
```

Controlling Access to Longer ID Settings

By default, IAM users and roles do not have permission to use the following actions unless they're explicitly granted permission through their associated IAM policies:

- `ec2:DescribeIdFormat`
- `ec2:DescribeIdentityIdFormat`
- `ec2:DescribeAggregateIdFormat`
- `ec2:DescribePrincipalIdFormat`
- `ec2:ModifyIdFormat`
- `ec2:ModifyIdentityIdFormat`

For example, an IAM role may have permission to use all Amazon EC2 actions through an "Action": "`ec2:*`" element in the policy statement.

To prevent IAM users and roles from viewing or modifying the longer resource ID settings for themselves or other users and roles in your account, ensure that the IAM policy contains the following statement:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Action": [  
                "ec2:ModifyIdFormat",  
                "ec2:DescribeIdFormat",  
                "ec2:ModifyIdentityIdFormat",  
                "ec2:DescribeIdentityIdFormat",  
                "ec2:DescribeAggregateIdFormat",  
                "ec2:DescribePrincipalIdFormat"  
            ]  
        }  
    ]  
}
```

```
        "ec2:DescribeAggregateIdFormat",
        "ec2:DescribePrincipalIdFormat"
    ],
    "Resource": "*"
}
}
```

We do not support resource-level permissions for the following actions:

- `ec2:DescribeIdFormat`
- `ec2:DescribeIdentityIdFormat`
- `ec2:DescribeAggregateIdFormat`
- `ec2:DescribePrincipalIdFormat`
- `ec2:ModifyIdFormat`
- `ec2:ModifyIdentityIdFormat`

Listing and Filtering Your Resources

You can get a list of some types of resource 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

- [Advanced Search \(p. 884\)](#)
- [Listing Resources Using the Console \(p. 885\)](#)
- [Filtering Resources Using the Console \(p. 886\)](#)
- [Listing and Filtering Using the CLI and API \(p. 887\)](#)

Advanced Search

Advanced search allows you to search using a combination of filters to achieve precise results. You can filter by keywords, user-defined tag keys, and predefined resource attributes.

The specific search types available are:

- **Search by keyword**

To search by keyword, type or paste what you're looking for in the search box, and then choose Enter. For example, to search for a specific instance, you can type the instance ID.

- **Search by fields**

You can also search by fields, tags, and attributes associated with a resource. For example, to find all instances in the stopped state:

1. In the search box, start typing **Instance State**. As you type, you'll see a list of suggested fields.
2. Select **Instance State** from the list.
3. Select **Stopped** from the list of suggested values.
4. To further refine your list, select the search box for more search options.

- **Advanced search**

You can create advanced queries by adding multiple filters. For example, you can search by tags and see instances for the Flying Mountain project running in the Production stack, and then search by attributes to see all t2.micro instances, or all instances in us-west-2a, or both.

- **Inverse search**

You can search for resources that do not match a specified value. For example, to list all instances that are not terminated, search by the **Instance State** field, and prefix the Terminated value with an exclamation mark (!).

- **Partial search**

When searching by field, you can also enter a partial string to find all resources that contain the string in that field. For example, search by **Instance Type**, and then type **t2** to find all t2.micro, t2.small or t2.medium instances.

- **Regular expression**

Regular expressions are useful when you need to match the values in a field with a specific pattern. For example, search by the Name tag, and then type **^s.*** to see all instances with a Name tag that starts with an 's'. Regular expression search is not case-sensitive.

After you have the precise results of your search, you can bookmark the URL for easy reference. In situations where you have thousands of instances, filters and bookmarks can save you a great deal of time; you don't have to run searches repeatedly.

Combining search filters

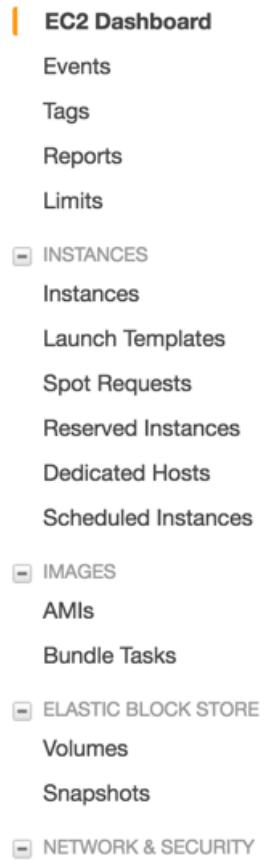
In general, multiple filters with the same key field (for example, tag:Name, search, Instance State) are automatically joined with OR. This is intentional, as the vast majority of filters would not be logical if they were joined with AND. For example, you would get zero results for a search on Instance State=running AND Instance State=stopped. In many cases, you can granulate the results by using complementary search terms on different key fields, where the AND rule is automatically applied instead. If you search for tag: Name:=All values and tag:Instance State=running, you get search results that contain both those criteria. To fine-tune your results, simply remove one filter in the string until the results fit your requirements.

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, such as **AMIs** or **Instances**.



3. The page displays all the available resources.

Filtering Resources Using the Console

You can perform filtering and sorting of the most common resource types using the Amazon EC2 console. For example, you can use the search bar on the instances page to sort instances by tags, attributes, or keywords.

You can also use the search field on each page to find resources with specific attributes or values. You can use regular expressions to search on partial or multiple strings. For example, to find all instances that are using the MySG security group, enter `MySG` in the search field. The results will include any values that contain `MySG` as a part of the string, such as `MySG2` and `MySG3`. To limit your results to `MySG` only, enter `\bMySG\b` in the search field. To list all the instances whose type is either `m1.small` or `m1.large`, enter `m1.small|m1.large` in the search field.

To list volumes in the `us-east-1b` Availability Zone with a status of `available`

1. In the navigation pane, choose **Volumes**.
2. Click on the search box, select **Attachment Status** from the menu, and then select **Detached**. (A detached volume is available to be attached to an instance in the same Availability Zone.)
3. Click on the search box again, select **State**, and then select **Available**.
4. Click on the search box again, select **Availability Zone**, and then select `us-east-1b`.
5. Any volumes that meet this criteria are displayed.

To list public 64-bit Linux AMIs backed by Amazon EBS

1. In the navigation pane, choose **AMIs**.
2. In the **Filter** pane, select **Public images**, **EBS images**, and then your Linux distribution from the **Filter** lists.
3. Type `x86_64` in the search field.
4. Any AMIs that meet this criteria are displayed.

Listing and Filtering Using the CLI and API

Each resource type has a corresponding CLI command or API request that you use to list resources of that type. For example, you can list Amazon Machine Images (AMIs) using `ec2-describe-images` or `DescribeImages`. The response contains information for all your resources.

The resulting lists of resources can be long, so you might want to filter the results to include only the resources that match certain criteria. You can specify multiple filter values, and you can also specify multiple filters. For example, you can list all the instances whose type is either `m1.small` or `m1.large`, and that have an attached EBS volume that is set to delete when the instance terminates. The instance must match all your filters to be included in the results.

You can also use wildcards with the filter values. An asterisk (*) matches zero or more characters, and a question mark (?) matches zero or one character.

For example, you can use `database` as the filter value to get only the EBS snapshots whose description equals `database`. If you specify `*database*`, then all snapshots whose description includes `database` are returned. If you specify `database?`, then only the snapshots whose description matches one of the following patterns are returned: `equals database` or `equals database` followed by one character.

The number of question marks determines the maximum number of characters to include in results. For example, if you specify `database????`, then only the snapshots whose description equals `database` followed by up to four characters are returned. Descriptions with five or more characters following `database` are excluded from the search results.

Filter values are case sensitive. We support only exact string matching, or substring matching (with wildcards). If a resulting list of resources is long, using an exact string filter may return the response faster.

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?\\`.

For a list of supported filters per Amazon EC2 resource, see the relevant documentation:

- For the AWS CLI, see the relevant `describe` command in the [AWS CLI Command Reference](#).
- For Windows PowerShell, see the relevant `Get` command in the [AWS Tools for PowerShell Cmdlet Reference](#).
- For the Query API, see the relevant `Describe` API action in the [Amazon EC2 API Reference](#).

Tagging Your Amazon EC2 Resources

To help you manage your instances, images, and other Amazon EC2 resources, you can optionally assign your own metadata to each resource in the form of *tags*. This topic describes tags and shows you how to create them.

Contents

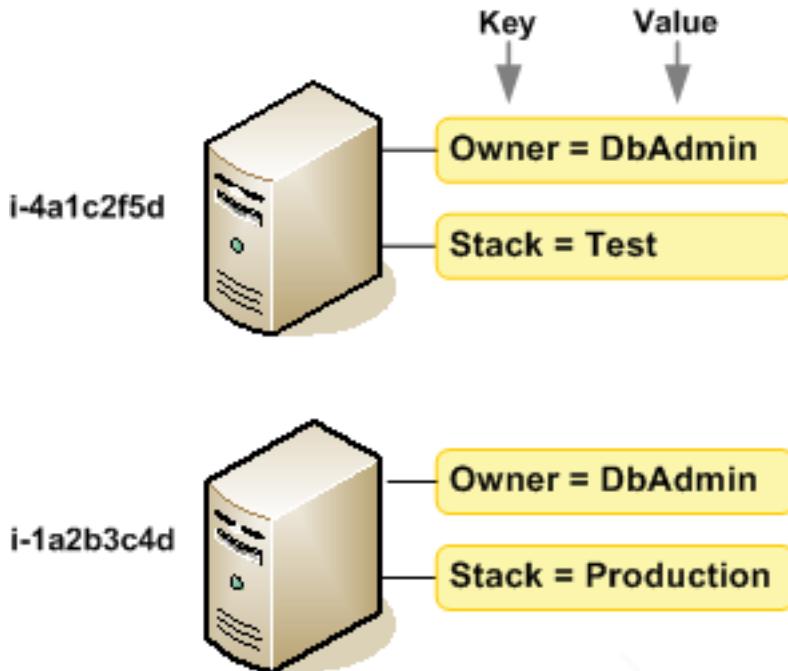
- [Tag Basics \(p. 888\)](#)
- [Tagging Your Resources \(p. 889\)](#)
- [Tag Restrictions \(p. 891\)](#)
- [Tagging Your Resources for Billing \(p. 891\)](#)
- [Working with Tags Using the Console \(p. 892\)](#)
- [Working with Tags Using the CLI or API \(p. 895\)](#)

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. This is useful when you have many resources of the same type—you can quickly identify a specific resource based on the tags you've assigned to it. 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.

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 [Controlling Access to Amazon EC2 Resources \(p. 545\)](#).

Tagging Your Resources

You can tag most Amazon EC2 resources that already exist in your account. The [table \(p. 889\)](#) 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.

Tagging Support for Amazon EC2 Resources

Resource	Supports tags	Supports tagging on creation (Amazon EC2 API, AWS CLI, AWS SDK)
AFI	Yes	No
AMI	Yes	No
Bundle task	No	No
Customer gateway	Yes	No
Dedicated Host	Yes	No
DHCP option	Yes	No
EBS snapshot	Yes	Yes
EBS volume	Yes	Yes
EC2 Fleet	Yes	Yes
Egress-only internet gateway	No	No
Elastic IP address	Yes	No
Instance	Yes	Yes

Resource	Supports tags	Supports tagging on creation (Amazon EC2 API, AWS CLI, AWS SDK)
Instance store volume	N/A	N/A
Internet gateway	Yes	No
Key pair	No	No
Launch template	Yes	No
Launch template version	No	No
NAT gateway	Yes	No
Network ACL	Yes	No
Network interface	Yes	No
Placement group	No	No
Reserved Instance	Yes	No
Reserved Instance listing	No	No
Route table	Yes	No
Spot Instance request	Yes	No
Security group–EC2-Classic	Yes	No
Security group–VPC	Yes	No
Subnet	Yes	No
Virtual private gateway	Yes	No
VPC	Yes	No
VPC endpoint	No	No
VPC endpoint service	No	No
VPC flow log	No	No
VPC peering connection	Yes	No
VPN connection	Yes	No

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 [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 557\)](#) and [Example Policies for Working with the AWS CLI or an AWS SDK \(p. 586\)](#).

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
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Don't use the `aws :` prefix for either keys or values; it's reserved for AWS use. You can't edit or delete tag keys or values with this prefix. Tags with this 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 \(p. 889\)](#).

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](#).

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

If you've just enabled reporting, data for the current month is available for viewing after 24 hours.

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 \(p. 884\)](#).

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*.

Contents

- [Displaying Tags \(p. 892\)](#)
- [Adding and Deleting Tags on an Individual Resource \(p. 893\)](#)
- [Adding and Deleting Tags to a Group of Resources \(p. 893\)](#)
- [Adding a Tag When You Launch an Instance \(p. 894\)](#)
- [Filtering a List of Resources by Tag \(p. 894\)](#)

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 a list of 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 in the **Tags** tab on the details pane.

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 the following data:

Tag Key	Tag Value	Total	Instances	AMIs	Volumes
Manage Tag	Name	1	1	0	0
Manage Tag	Owner	2	0	0	2
Manage Tag	Owner	2	0	0	2
Manage Tag	Purpose	1	0	0	1
Manage Tag	Purpose	1	0	0	1
Manage Tag	Purpose	1	1	0	0
Manage Tag	Purpose	2	0	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 \(p. 878\)](#).
3. In the navigation pane, select a resource type (for example, [Instances](#)).
4. Select the resource from the resource list and choose [Tags, Add/Edit Tags](#).
5. In the [Add/Edit Tags](#) dialog box, specify the key and value for each tag, and then 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 \(p. 878\)](#).
3. In the navigation pane, choose a resource type (for example, [Instances](#)).
4. Select the resource from the resource list and choose [Tags](#).
5. Choose [Add/Edit Tags](#), select the [Delete](#) icon for the tag, and 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 \(p. 878\)](#).
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) to which to add tags.

6. In the resources list, select the check box next to each resource to which to add tags.
7. Under **Add Tag**, for **Key** and **Value**, type the tag key and values, and then 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 \(p. 878\)](#).
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) from which to remove tags.
6. In the resource list, select the check box next to each resource from which to remove tags.
7. Under **Remove Tag**, for **Key**, type the tag's name 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 \(p. 878\)](#).
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 a Linux AMI \(p. 87\)](#).
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. Display a column for the tag as follows:

- a. Select a resource.
- b. In the details pane, choose **Tags**.
- c. Locate the tag in the list and choose **Show Column**.
2. Choose the filter icon in the top right corner of the column for the tag to display the filter list.
3. Select the tag values, and then choose **Apply Filter** to filter the results list.

Note

For more information about filters, see [Listing and Filtering Your Resources \(p. 884\)](#).

Working with Tags Using the CLI or API

Use the following to add, update, list, and delete the tags for your resources. The corresponding documentation provides examples.

Task	AWS CLI	AWS Tools for Windows PowerShell	API Action
Add or overwrite one or more tags.	create-tags	New-EC2Tag	CreateTags
Delete one or more tags.	delete-tags	Remove-EC2Tag	DeleteTags
Describe one or more tags.	describe-tags	Get-EC2Tag	DescribeTags

You can also filter a list of resources according to their tags. The following examples demonstrate how to filter your instances using tags with the [describe-instances](#) command.

Note

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 the single quote ('') 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 1: 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 2: 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 3: 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
```

Some resource-creating actions enable you to specify tags when you create the resource. The following actions support tagging on creation.

Task	AWS CLI	AWS Tools for Windows PowerShell	API Action
Launch one or more instances.	run-instances	New-EC2Instance	RunInstances
Create an EBS volume.	create-volume	New-EC2Volume	CreateVolume

The following examples demonstrate how to apply tags when you create resources.

Example 4: Launch an instance and apply tags to the instance and volume

The following command launches an instance and applies a tag with a key of `webserver` and value of `production` to the instance. The command also applies a tag with a key of `cost-center` and a value of `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 5: Create a volume and apply a tag

The following 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}]'
```

Amazon EC2 Service Limits

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 limits on these resources on a per-region basis. For example, there is a limit on the number of instances that you can launch in a region. Therefore, when you launch an instance in the US West (Oregon) region, the request must not cause your usage to exceed your current instance limit 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.

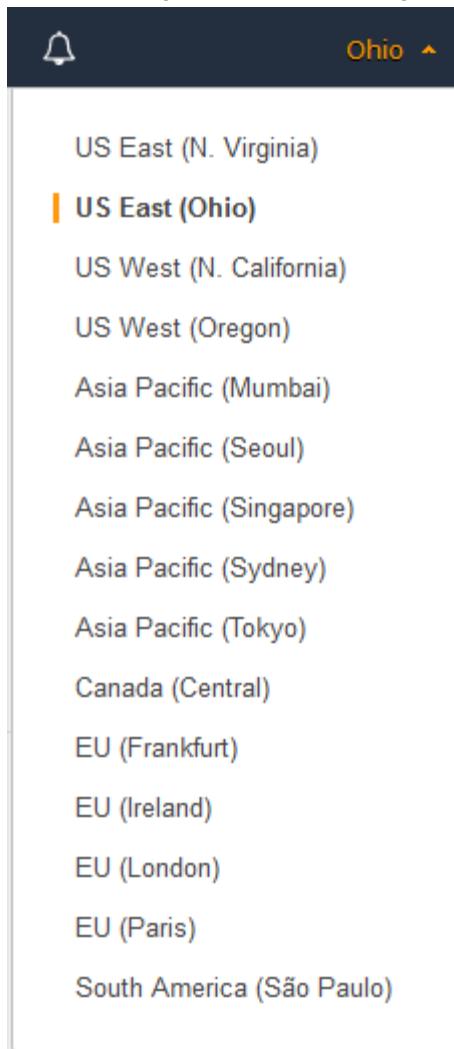
For more information about the limits for other services, see [AWS Service Limits](#) in the *Amazon Web Services General Reference*.

Viewing Your Current Limits

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.

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.



3. From the navigation pane, choose **Limits**.
4. Locate the resource in the list. The **Current Limit** column displays the current maximum for that resource for your account.

Requesting a Limit Increase

Use the **Limits** page in the Amazon EC2 console to request an increase in the limits for resources provided by Amazon EC2 or Amazon VPC, on a per-region basis.

To request a limit increase

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. Locate the resource in the list. Choose **Request limit increase**.
5. Complete the required fields on the limit increase form. We'll respond to you using the contact method that you specified.

Amazon EC2 Usage Reports

AWS provides a free reporting tool called Cost Explorer that enables you to analyze the cost and usage of your EC2 instances and the usage of your Reserved Instances.

Cost Explorer is a free tool that you can use to view charts of your usage and costs. 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?

To view an Amazon EC2 report in Cost Explorer

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reports** and select the report to view.

The report opens in Cost Explorer. It provides a preconfigured view, based on fixed filter settings, that displays information about your usage and cost trends.

For more information about working with reports in Cost Explorer, including saving reports, see [Analyzing Your Costs with Cost Explorer](#).

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.

Note

If you are using a Windows instance, see [EC2Rescue for Windows Server](#).

Contents

- [Installing EC2Rescue for Linux \(p. 899\)](#)
- [Working with EC2Rescue for Linux \(p. 900\)](#)
- [Developing EC2Rescue Modules \(p. 902\)](#)

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+
 - SLES 12+
 - RHEL 7+
 - Ubuntu 16.04+
- Software requirements:
 - Python 2.7.9+ or 3.2+

To install EC2Rescue for Linux

1. From a working Linux instance, download the [EC2Rescue for Linux](#) tool:

```
[ec2-user ~]$ curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz
```

2. Download the sha256 hash file:

```
[ec2-user ~]$ curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz.sha256
```

3. Verify the integrity of the tarball:

```
[ec2-user ~]$ sha256sum -c ec2rl.tgz.sha256
```

4. Unpack the tarball:

```
[ec2-user ~]$ tar -xvf ec2rl.tgz
```

5. Verify the installation by listing out the help file:

```
[ec2-user ~]$ cd ec2rl-<version_number>  
[ec2-user ~]$ ./ec2rl help
```

Working with EC2Rescue for Linux

The following are common tasks to get you started using this tool.

Topics

- [Getting Help \(p. 900\)](#)
- [Running a Module \(p. 900\)](#)
- [Uploading the Results \(p. 901\)](#)
- [Creating Backups \(p. 901\)](#)

Getting Help

EC2Rescue for Linux includes a help file that gives you information and syntax for each available command.

To use the help command

- Use the `help` command to get general help:

```
[ec2-user ~]$ ./ec2rl help
```

To list the available modules

- List all available modules:

```
[ec2-user ~]$ ./ec2rl list
```

To get help for a specific module

- List the help details for a specific command:

```
[ec2-user ~]$ ./ec2rl help module_name
```

Example command for showing the help file for the `dig` module:

```
[ec2-user ~]$ ./ec2rl help dig
```

Running a Module

You can run an EC2Rescue for Linux using these steps.

To run a module

1. Run a module:

```
[ec2-user ~]$ ./ec2rl run --only-modules=module_name --arguments
```

Example command, using the dig module to query the amazon.com domain:

```
[ec2-user ~]$ ./ec2rl run --only-modules=dig --domain=amazon.com
```

2. View the results:

```
[ec2-user ~]$ cat /var/tmp/ec2rl/logfile_location
```

Example:

```
[ec2-user ~]$ 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 using the EC2Rescue for Linux CLI tool. The output of the EC2Rescue for Linux commands should provide the commands to achieve this.

To upload to support

- Upload the results to support:

```
[ec2-user ~]$ ./ec2rl upload --upload-directory=/var/tmp/  
ec2rl/2017-05-11T15_39_21.893145 --support-url="URLProvidedByAWSSupport"
```

To upload to an S3 bucket

- To upload the results to an S3 bucket:

```
[ec2-user ~]$ ./ec2rl upload --upload-directory=/var/tmp/  
ec2rl/2017-05-11T15_39_21.893145 --presigned-url="YourPresignedS3URL"
```

Note

For more information about generating presigned 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.

To back up an instance

- Create a backup of your instance:

```
[ec2-user ~]$ ./ec2rl run --backup=ami
```

To back up all volumes

- Create a backup of all volumes associated with the instance:

```
[ec2-user ~]$ ./ec2rl run --backup=allvolumes
```

To back up a volume

- Create a backup of a single volume:

```
[ec2-user ~]$ ./ec2rl run --backup=volume ID
```

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	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. For example: <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>

Attribute	Description
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	<p>Indicates whether the module supports remediation. Supported values are <code>True</code> or <code>False</code>.</p> <p>The module will default to <code>False</code> if this is absent, making it an optional attribute for those modules that do not support remediation.</p>
content	The entirety of the script code.
constraint	The name of the object containing the constraint values.
domain	<p>A descriptor of how the module is grouped or classified. The set of included modules uses the following domains:</p> <ul style="list-style-type: none"> • application • net • os • performance
class	<p>A descriptor of the type of task performed by the module. The set of included modules uses the following classes:</p> <ul style="list-style-type: none"> • collect (collects output from programs) • diagnose (pass/fail based on a set of criteria) • gather (copies files and writes to specific file)

Attribute	Description
distro	The list of Linux distributions that this module supports. The set of included modules use the following distributions: <ul style="list-style-type: none"> • alami (Amazon Linux) • rhel • ubuntu • suse
required	The required arguments that the module is consuming from the CLI options.
optional	The optional arguments that the module can use.
software	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.
package	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.
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 <code>ec2rl.py</code> . This path can be used to locate the <code>lib</code> directory and utilize vendored Python modules.

Environment Variable	Description
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&tstamp>.
EC2RL_GATHEREDDIR	The root directory for placing gathered module data. Default value: /var/tmp/ec2rl/<date&tstamp>/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.
EC2RL_VIRT_TYPE	The virtualization type as provided by the instance metadata. Examples: <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 functions.bash 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.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
```

Troubleshooting Instances

The following documentation can help you troubleshoot problems that you might have with your instance.

Contents

- [Troubleshooting Instance Launch Issues \(p. 907\)](#)
- [Troubleshooting Connecting to Your Instance \(p. 909\)](#)
- [Troubleshooting Stopping Your Instance \(p. 916\)](#)
- [Troubleshooting Terminating \(Shutting Down\) Your Instance \(p. 917\)](#)
- [Troubleshooting Instance Recovery Failures \(p. 918\)](#)
- [Troubleshooting Instances with Failed Status Checks \(p. 918\)](#)
- [Getting Console Output and Rebooting Instances \(p. 940\)](#)
- [Booting from the Wrong Volume \(p. 943\)](#)

For additional help with Windows instances, see [Troubleshooting Windows Instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

You can also search for answers and post questions on the [Amazon EC2 forum](#).

Troubleshooting Instance Launch Issues

The following issues prevent you from launching an instance.

Launch Issues

- [Instance Limit Exceeded \(p. 907\)](#)
- [Insufficient Instance Capacity \(p. 908\)](#)
- [Instance Terminates Immediately \(p. 908\)](#)

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 Limits \(p. 896\)](#).

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 \(p. 228\)](#).
- 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 \(p. 702\)](#).
- 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 \(p. 863\)](#). 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 Limits \(p. 896\)](#).
- 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 are possible problems you may have and error messages you may see while trying to connect to your instance.

Contents

- [Error connecting to your instance: Connection timed out \(p. 910\)](#)
- [Error: User key not recognized by server \(p. 911\)](#)
- [Error: Host key not found, Permission denied \(publickey\), or Authentication failed, permission denied \(p. 913\)](#)
- [Error: Unprotected Private Key File \(p. 914\)](#)
- [Error: Server refused our key or No supported authentication methods available \(p. 914\)](#)
- [Error Using MindTerm on Safari Browser \(p. 915\)](#)
- [Error Using macOS RDP Client \(p. 915\)](#)

- [Cannot Ping Instance \(p. 915\)](#)

For additional help with Windows instances, see [Troubleshooting Windows Instances in the Amazon EC2 User Guide for Windows Instances](#).

You can also search for answers and post questions on the [Amazon EC2 forum](#).

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, next to **Security groups**, choose **view rules** to display the list of rules that are in effect.
 4. For Linux instances: Verify that there is a rule that allows traffic from your computer to port 22 (SSH).

For Windows instances: Verify that there is a rule that allows traffic from your computer to port 3389 (RDP).

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 \(p. 628\)](#).

- [EC2-VPC] 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. Otherwise, choose the ID of the route table (`rtb-xxxxxx`) to navigate to the **Routes** tab for the route table, choose **Edit, Add another route**, enter `0.0.0.0/0` in **Destination**, select your internet gateway from **Target**, and then choose **Save**.

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. If not, add a route with `::/0` as the destination, and the internet gateway as the target.

- [EC2-VPC] 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 **Summary** 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 \(p. 674\)](#).
- 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 \(p. 483\)](#).
 - 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 \(p. 228\)](#).

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: 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 [your key name].pem ec2-user@[public DNS address of your instance].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 SSH (MindTerm) to connect to your instance

- If Java is not enabled, the server does not recognize the user key. To enable Java, go to [How do I enable Java in my web browser?](#) in the Java documentation.

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 \(p. 402\)](#).

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` or `root`.
 - 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 \(p. 628\)](#).

Error: Host key not found, Permission denied (publickey), or Authentication failed, permission denied

If you connect to your instance using SSH and get any of the following errors, Host key not found in [directory], Permission denied (publickey), or Authentication failed, permission denied, 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. For MindTerm clients, enter the user name in the **User name** box in the **Connect To Your Instance** 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` or `root`.
- 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 ec2-user@public-dns-hostname
```

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 \(p. 528\)](#).

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/ec2-user/.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 \(p. 415\)](#) and [Detaching an Amazon EBS Volume from an Instance \(p. 767\)](#).
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 \(p. 750\)](#).

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 \(p. 751\)](#).
4. From the temporary instance, check the permissions of the /home/ec2-user/ directory of the attached volume. If necessary, adjust the permissions as follows:

```
[ec2-user ~]$ chmod 600 mount_point/home/ec2-user/.ssh/authorized_keys
```

```
[ec2-user ~]$ chmod 700 mount_point/home/ec2-user/.ssh
```

```
[ec2-user ~]$ chmod 700 mount_point/home/ec2-user
```

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: 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. Enter the user name in the **User name** box 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` or `root`.
- 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 \(p. 402\)](#).

Error Using MindTerm on Safari Browser

If you use MindTerm to connect to your instance, and are using the Safari web browser, you may get the following error:

```
Error connecting to your_instance_ip, reason:  
-> Key exchange failed: Host authentication failed
```

You must update the browser's security settings to allow the AWS Management Console to run the Java plugin in unsafe mode.

To enable the Java plugin to run in unsafe mode

1. In Safari, keep the Amazon EC2 console open, and choose **Safari, Preferences, Security**.
2. Choose **Plug-in Settings** (or **Manage Website Settings** on older versions of Safari).
3. Choose the **Java** plugin on the left.
4. For **Currently Open Websites**, select the AWS Management Console URL and choose **Run in Unsafe Mode**.
5. When prompted, choose **Trust** in the warning dialog box and choose **Done**.

Error Using macOS RDP Client

If you are connecting to a Windows Server 2012 R2 instance using the Remote Desktop Connection client from the Microsoft website, you may get the following error:

```
Remote Desktop Connection cannot verify the identity of the computer that you want to  
connect to.
```

Download the Microsoft Remote Desktop app from the Apple iTunes store, and use the app to connect to your instance.

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.

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 \(p. 102\)](#)

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.

Automatically Launch or Terminate Instances

If you terminate all your instances, you may see that we launch a new instance for you. If you launch an instance, you may see that we terminate one of your instances. If you stop an instance, you may see that we terminate the instance and launch a new instance. Generally, these behaviors mean that you've used Amazon EC2 Auto Scaling or Elastic Beanstalk to scale your computing resources automatically based on criteria that you've defined.

For more information, see the [Amazon EC2 Auto Scaling User Guide](#) or the [AWS Elastic Beanstalk Developer Guide](#).

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 start and stop the instance. For more information, see [Stop and Start Your Instance \(p. 415\)](#).

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.

Troubleshooting Instances with Failed Status Checks

Topics

- [Initial Steps \(p. 919\)](#)
- [Retrieving System Logs \(p. 920\)](#)
- [Troubleshooting System Log Errors for Linux-Based Instances \(p. 920\)](#)
- [Out of memory: kill process \(p. 921\)](#)

- [ERROR: mmu_update failed \(Memory management update failed\) \(p. 922\)](#)
- [I/O Error \(Block Device Failure\) \(p. 922\)](#)
- [I/O ERROR: neither local nor remote disk \(Broken distributed block device\) \(p. 923\)](#)
- [request_module: runaway loop modprobe \(Looping legacy kernel modprobe on older Linux versions\) \(p. 924\)](#)
- ["FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" \(Kernel and AMI mismatch\) \(p. 925\)](#)
- ["FATAL: Could not load /lib/modules" or "BusyBox" \(Missing kernel modules\) \(p. 926\)](#)
- [ERROR Invalid kernel \(EC2 incompatible kernel\) \(p. 927\)](#)
- [request_module: runaway loop modprobe \(Looping legacy kernel modprobe on older Linux versions\) \(p. 928\)](#)
- [fsck: No such file or directory while trying to open... \(File system not found\) \(p. 929\)](#)
- [General error mounting filesystems \(Failed mount\) \(p. 930\)](#)
- [VFS: Unable to mount root fs on unknown-block \(Root filesystem mismatch\) \(p. 932\)](#)
- [Error: Unable to determine major/minor number of root device... \(Root file system/device mismatch\) \(p. 933\)](#)
- [XENBUS: Device with no driver... \(p. 934\)](#)
- [... days without being checked, check forced \(File system check required\) \(p. 934\)](#)
- [fsck died with exit status... \(Missing device\) \(p. 935\)](#)
- [GRUB prompt \(grubdom>\) \(p. 936\)](#)
- [Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. \(Hard-coded MAC address\) \(p. 938\)](#)
- [Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. \(SELinux misconfiguration\) \(p. 939\)](#)
- [XENBUS: Timeout connecting to devices \(Xenbus timeout\) \(p. 940\)](#)

Initial Steps

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, follow these steps:

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, or Recover an Instance](#) in the *Amazon CloudWatch User Guide*.
- 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](#).
- Retrieve the system log and look for errors.

Retrieving 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 \(p. 921\)](#)
- [ERROR: mmu_update failed \(Memory management update failed\) \(p. 922\)](#)

Device Errors

- [I/O Error \(Block Device Failure\) \(p. 922\)](#)
- [I/O ERROR: neither local nor remote disk \(Broken distributed block device\) \(p. 923\)](#)

Kernel Errors

- [request_module: runaway loop modprobe \(Looping legacy kernel modprobe on older Linux versions\) \(p. 924\)](#)
- ["FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" \(Kernel and AMI mismatch\) \(p. 925\)](#)
- ["FATAL: Could not load /lib/modules" or "BusyBox" \(Missing kernel modules\) \(p. 926\)](#)
- [ERROR Invalid kernel \(EC2 incompatible kernel\) \(p. 927\)](#)

File System Errors

- [request_module: runaway loop modprobe \(Looping legacy kernel modprobe on older Linux versions\) \(p. 928\)](#)

- fsck: No such file or directory while trying to open... (File system not found) (p. 929)
- General error mounting filesystems (Failed mount) (p. 930)
- VFS: Unable to mount root fs on unknown-block (Root filesystem mismatch) (p. 932)
- Error: Unable to determine major/minor number of root device... (Root file system/device mismatch) (p. 933)
- XENBUS: Device with no driver... (p. 934)
- ... days without being checked, check forced (File system check required) (p. 934)
- fsck died with exit status... (Missing device) (p. 935)

Operating System Errors

- GRUB prompt (grubdom>) (p. 936)
- Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address) (p. 938)
- Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration) (p. 939)
- XENBUS: Timeout connecting to devices (Xenbus timeout) (p. 940)

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.

For this instance type	Do this
	<ul style="list-style-type: none">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
```

Amazon Elastic Compute Cloud
User Guide for Linux Instances
I/O ERROR: neither local nor remote
disk (Broken distributed block device)

```
[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. Detach the volume.
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)

0MB HIGHMEM available.
...

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c
request_module: runaway loop modprobe binfmt-464c
request_module: runaway loop modprobe binfmt-464c
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:

For this instance type	Do this
	Option 1: Terminate the instance and launch a new instance, specifying the <code>-kernel</code> and <code>-ramdisk</code> parameters. 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.

```
[    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.

For this instance type	Do this
	4. Attach the volume to the instance. 5. Start the instance. 6. Modify the AMI to use the corrected ramdisk.
Instance store-backed	Use the following procedure: 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	Use the following procedure: 1. Stop the instance. 2. Replace with working kernel.

For this instance type	Do this
	3. Install a fallback kernel. 4. Modify the AMI by correcting the kernel.
Instance store-backed	Use the following procedure: 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 .

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 - 0000000026700000 (usable)

0MB HIGHMEM available.
...

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. Option 2: 1. Stop the instance. 2. Modify the kernel and ramdisk attributes to use a newer kernel.

For this instance type	Do this
	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.

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 (fsck -a /dev/...).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 not is 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.

Getting Console Output and Rebooting Instances

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.

Similarly, the ability to reboot instances that are otherwise unreachable is valuable for both troubleshooting and general instance management.

EC2 instances do not have a physical monitor through which you can view their console output. They also lack physical controls that allow you to power up, reboot, or shut them down. Instead, you perform these tasks through the Amazon EC2 API and the command line interface (CLI).

Instance Reboot

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 \(p. 418\)](#).

Warning

For Windows instances, this operation performs a hard reboot that might result in data corruption.

Instance Console Output

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 displays 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 instance types that use the Nitro hypervisor. 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 \(p. 920\)](#).

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. This provides visibility as to the status of the instance, and allows for quicker troubleshooting.

There is no data transfer cost for this screenshot. The image is generated in JPG format, no larger than 100 KB.

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.
5. [EC2-Classic] If the instance had an associated Elastic IP address, you must reassociate it with the instance.

For more information, see [Stop and Start Your Instance \(p. 415\)](#).

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.
7. [EC2-Classic] If the original instance had an associated Elastic IP address, you must associate it with the new instance.

For more information, see [Creating an Instance Store-Backed Linux AMI \(p. 105\)](#).

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 /.

Document History

The following table describes important additions to the Amazon EC2 documentation. We also update the documentation frequently to address the feedback that you send us.

Current API version: 2016-11-15.

Feature	API Version	Description	Release Date
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 Strategy for Spot Instances (p. 376) .	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 Spot Fleet Allocation Strategy (p. 283) .	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. 197) .	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. 197) .	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. 804) .	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. 363) .	11 July 2018
Tag Dedicated Hosts	2016-11-15	You can tag your Dedicated Hosts. For more information, see Tagging Dedicated Hosts (p. 340) .	3 July 2018

Feature	API Version	Description	Release Date
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. 202) .	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. 232) .	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 an EC2 Fleet (p. 374) .	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. 282) .	2 May 2018
Tag EBS snapshots on creation	2016-11-15	You can apply tags to snapshots during creation. For more information, see Creating an Amazon EBS Snapshot (p. 795) .	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. 704) .	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. 879) .	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. 166) .	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. 678) .	21 December 2017
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. 146) .	13 December 2017

Feature	API Version	Description	Release Date
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. 439) .	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 T2 Unlimited (p. 180) .	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. 361) .	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. 702) .	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. 202) .	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. 326) .	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. 316) .	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. 197) .	28 November 2017
C5 instances	2016-11-15	C5 instances are designed for compute-heavy applications. For more information, see Compute Optimized Instances (p. 194) .	6 November 2017

Feature	API Version	Description	Release Date
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. 270) .	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. 208) .	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. 350) .	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 Behavior (p. 325) .	18 September 2017
Tag NAT gateways	2016-11-15	You can tag your NAT gateway. For more information, see Tagging Your Resources (p. 889) .	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. 532) .	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. 682) .	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. 208) .	13 July 2017
SSL/TLS tutorial update	2016-11-15	Set up SSL/TLS support on your EC2 webserver using Let's Encrypt. For more information, see Tutorial: Configure Apache Web Server on Amazon Linux 2 to Use SSL/TLS (p. 59) .	25 April 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. 208) .	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. 889) . In addition, you can use tag-based resource-level permissions to control the tags that are applied. For more information see, Resource-Level Permissions for Tagging (p. 584) .	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. 202) .	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 Modifying the Size, IOPS, or Type of an EBS Volume on Linux (p. 769) .	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. 620) .	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. 291) .	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. 658) .	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. 197)	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 T2 Instances (p. 168) .	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. 208) .	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. 316) .	1 September 2016
Elastic Network Adapter (ENA)	2016-04-01	You can now use ENA for enhanced networking. For more information, see Enhanced Networking Types (p. 709) .	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. 879) .	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. 799) .	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. 941) .	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. 197) .	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. 735) .	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. 485) .	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. 314) .	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. 274) .	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. 879) .	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. 644) .	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 T2 Instances (p. 168) .	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. 332) .	23 November 2015
Spot instance duration	2015-10-01	You can now specify a duration for your Spot instances. For more information, see Specifying a Duration for Your Spot Instances (p. 290) .	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. 304) .	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 Spot Fleet Allocation Strategy (p. 283) .	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 your bid price for each Spot pool accordingly. For more information, see Spot Fleet Instance Weighting (p. 284) .	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. 502) .	23 July 2015
New <code>t2.large</code> 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 T2 Instances (p. 168) .	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. 282) .	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 the EC2-Classic platform to the EC2-VPC platform. For more information, see Migrating an Elastic IP Address from EC2-Classic to EC2-VPC (p. 677) .	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 <code>g2.8xlarge</code> instance type		The new <code>g2.8xlarge</code> 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. 202).</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. 426).</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. 194).</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. 638).</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. 328).</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 T2 Instances (p. 168).</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. 737).</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. 819).</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. 898) .	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. 202) .	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 Controlling Access to Amazon EC2 Resources (p. 545) .	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. 372) .	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. 208) .	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. 355) .	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. 264) .	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. 264) .	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. 664) .	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. 548) .	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. 799) .	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. 887) .	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. 812) .	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. 139) .	11 March 2013
Launch instances into a default VPC	2013-02-01	Your AWS account is capable of launching instances into either the EC2-Classic or EC2-VPC platform, or only into the EC2-VPC platform, on a region-by-region basis. If you can launch instances only into EC2-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. For more information, see Supported Platforms (p. 637) .	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. 799) .	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 Monitoring Volumes with CloudWatch (p. 754) . Also added new status checks for Provisioned IOPS SSD volumes. For more information, see Monitoring Volumes with Status Checks (p. 759) .	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. 735) .	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"> • Place bids for Spot instances using Auto Scaling launch configurations, and set up a schedule for placing bids 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
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

Feature	API Version	Description	Release Date
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. 683) .	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. 474) .	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
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

Feature	API Version	Description	Release Date
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. 157) .	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 Controlling Access to Amazon EC2 Resources (p. 545) .	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. 157) .	26 April 2011
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. 345) .	27 March 2011

Feature	API Version	Description	Release Date
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. 240) .	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. For more information, see AMIs for GPU-Based Accelerated Computing Instances (p. 212) .	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. 454) and Instance Metadata Categories (p. 460) .	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 an Instance (p. 767) .	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 for an Instance (p. 422) .	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
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

Feature	API Version	Description	Release Date
Filters and Tags	2010-08-31	Added information about listing, filtering, and tagging resources. For more information, see Listing and Filtering Your Resources (p. 884) and Tagging Your Amazon EC2 Resources (p. 887) .	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 T1 Micro Instances (p. 217) .	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 Controlling Access to Amazon EC2 Resources (p. 545) .	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 Monitoring Volumes with CloudWatch (p. 754) .	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

AWS Glossary

For the latest AWS terminology, see the [AWS Glossary](#) in the *AWS General Reference*.