
Amazon Elastic Compute Cloud

User Guide for Windows Instances



Amazon Elastic Compute Cloud: User Guide for Windows 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
Basic Infrastructure	4
Amazon Machine Images and Instances	4
Regions and Availability Zones	5
Storage	6
Root Device Volume	7
Networking and Security	8
AWS Identity and Access Management	8
Differences between Windows Server and an Amazon EC2 Windows Instance	9
Designing Your Applications to Run on Amazon EC2 Windows Instances	10
Setting Up	12
Sign Up for AWS	12
Create an IAM User	12
Create a Key Pair	14
Create a Virtual Private Cloud (VPC)	16
Create a Security Group	17
Getting Started	20
Overview	20
Prerequisites	21
Step 1: Launch an Instance	21
Step 2: Connect to Your Instance	22
Step 3: Clean Up Your Instance	23
Next Steps	24
Best Practices	25
Tutorials	26
Tutorial: Deploy a WordPress Blog	26
Prerequisites	26
Installing the Microsoft Web Platform Installer	27
Installing WordPress	27
Configuring Security Keys	28
Configuring the Site Title and Administrator	29
Making Your WordPress Site Public	29
Next Steps	30
Tutorial: Installing a WAMP Server	30
Tutorial: Installing a WIMP Server	33
Prerequisites	33
Prepare Your Instance	33
Install the IIS web server	34
Install MySQL and PHP	35
Test Your Server	35
Tutorial: Increase the Availability of Your Application	36
Prerequisites	37
Scale and Load Balance Your Application	37
Test Your Load Balancer	39
Tutorial: Remotely Manage Your Instances	39
Grant Your User Account Access to Systems Manager	40
Install the SSM Agent	40
Send a Command Using the EC2 Console	40

Send a Command Using AWS Tools for Windows PowerShell	41
Send a Command Using the AWS CLI	42
Related Content	43
Tutorial: Set Up a Windows HPC Cluster	43
Prerequisites	43
Step 1: Create Security Groups	43
Step 2: Set Up Your Active Directory Domain Controller	46
Step 3: Configure Your Head Node	47
Step 4: Set Up the Compute Node	48
Step 5: Scale Your HPC Compute Nodes (Optional)	49
Amazon Machine Images	51
Creating Your Own AMI	51
Buying, Sharing, and Selling AMIs	51
Deregistering Your AMI	51
AWS Windows AMIs	51
Selecting an Initial Windows AMI	52
Keeping Your AMIs Up-to-Date	52
Virtualization Types	52
Finding a Windows AMI	53
Finding a Windows AMI Using the Amazon EC2 Console	53
Finding an AMI Using the AWS Tools for Windows PowerShell	54
Finding an AMI Using the AWS CLI	54
Shared AMIs	55
Finding Shared AMIs	55
Making an AMI Public	57
Sharing an AMI with Specific AWS Accounts	58
Using Bookmarks	61
Guidelines for Shared Windows AMIs	61
Paid AMIs	61
Selling Your AMI	62
Finding a Paid AMI	62
Purchasing a Paid AMI	64
Getting the Product Code for Your Instance	64
Using Paid Support	64
Bills for Paid and Supported AMIs	65
Managing Your AWS Marketplace Subscriptions	65
Creating a Custom Windows AMI	65
Overview of Creating an AMI	66
Creating a Windows AMI from a Running Instance	66
AMIs with Encrypted Snapshots	68
AMI Scenarios Involving Encrypted EBS Snapshots	69
Copying an AMI	71
Permissions for Copying an Instance Store-Backed AMI	71
Cross-Region AMI Copy	72
Cross-Account AMI Copy	73
Encryption and AMI Copy	74
Copying an AMI	75
Stopping a Pending AMI Copy Operation	76
Deregistering Your Windows AMI	76
AWS Windows AMIs	77
Updating Your Windows Instance	78
Upgrading or Migrating to a Newer Version of Windows Server	78
Subscribing to Windows AMI Notifications	78
Configuration Changes for AWS Windows AMIs	79
Details About AWS Windows AMI Versions	81
Changes in Windows Server 2016 AMIs	102
Docker Container Conflict on Windows Server 2016 Instances	103

Issue with the Hibernate Agent (2018.03.16 AMIs)	103
Create an AMI Using Sysprep	104
Before You Begin	105
Using Sysprep with the EC2Config Service	105
Run Sysprep with the EC2Config Service	108
Troubleshooting Sysprep with EC2Config	109
Instances	110
Instance Types	110
Available Instance Types	111
Hardware Specifications	112
Nitro-based Instances	113
Networking and Storage Features	113
Instance Limits	116
Burstable Performance Instances	116
General Purpose Instances	143
Compute Optimized Instances	148
Memory Optimized Instances	152
Storage Optimized Instances	159
Accelerated Computing Instances	163
T1 Micro Instances	170
Changing the Instance Type	182
Instance Purchasing Options	185
Determining the Instance Lifecycle	185
Reserved Instances	186
Scheduled Instances	221
Spot Instances	225
Dedicated Hosts	283
Dedicated Instances	297
On-Demand Capacity Reservations	302
Instance Lifecycle	309
Instance Launch	311
Instance Stop and Start (Amazon EBS-Backed Instances Only)	311
Instance Hibernate (Amazon EBS-Backed Instances Only)	311
Instance Reboot	312
Instance Retirement	312
Instance Termination	312
Differences Between Reboot, Stop, Hibernate, and Terminate	313
Launch	314
Connect	359
Stop and Start	362
Hibernate	365
Reboot	365
Retire	366
Terminate	368
Recover	373
Configure Instances	373
EC2Launch	374
EC2Config Service	383
PV Drivers	409
AWS NVMe Drivers	424
Optimizing CPU Options	425
Setting the Time	435
Setting the Password	439
Adding Windows Components	440
Configuring a Secondary Private IPv4 Address	443
Running Commands at Launch	447
Instance Metadata and User Data	452

Upgrade Windows Instances	463
Performing a Server Migration	464
Performing an In-Place Upgrade	464
Migrating to Latest Generation Instance Types	468
Troubleshooting an Upgrade	473
Identify Instances	473
Inspecting the Instance Identity Document	473
Inspecting the System UUID	473
Elastic Graphics	475
Elastic Graphics Basics	475
Pricing for Elastic Graphics	477
Elastic Graphics Limitations	477
Working with Elastic Graphics	477
Configuring Your Security Groups	477
Launching an Instance with an Elastic Graphics accelerator	478
Installing the Required Software for Elastic Graphics	479
Verifying Elastic Graphics Functionality on Your Instance	479
Viewing Elastic Graphics Information	481
Submitting Feedback	482
Using CloudWatch Metrics to Monitor Elastic Graphics	482
Elastic Graphics Metrics	482
Elastic Graphics Dimensions	483
Viewing CloudWatch Metrics for Elastic Graphics	483
Creating CloudWatch Alarms to Monitor Elastic Graphics	483
Troubleshooting	484
Investigating Application Performance Issues	484
Resolving Unhealthy Status Issues	486
Monitoring	487
Automated and Manual Monitoring	488
Automated Monitoring Tools	488
Manual Monitoring Tools	489
Best Practices for Monitoring	489
Monitoring the Status of Your Instances	490
Instance Status Checks	490
Scheduled Events	494
Monitoring Your Instances Using CloudWatch	498
Enable Detailed Monitoring	499
List Available Metrics	500
Get Statistics for Metrics	509
Graph Metrics	516
Create an Alarm	516
Create Alarms That Stop, Terminate, Reboot, or Recover an Instance	517
Automating Amazon EC2 with CloudWatch Events	526
Sending Data to CloudWatch	526
Methods to Send Instance Metrics to CloudWatch	527
Preliminary Tasks for Configuring Integration with CloudWatch	528
Configure Instances for CloudWatch	536
Logging API Calls with AWS CloudTrail	540
Amazon EC2 and Amazon EBS Information in CloudTrail	541
Understanding Amazon EC2 and Amazon EBS Log File Entries	541
Network and Security	543
Key Pairs	543
Creating a Key Pair Using Amazon EC2	544
Importing Your Own Public Key to Amazon EC2	544
Retrieving the Public Key for Your Key Pair on Linux	546
Retrieving the Public Key for Your Key Pair on Windows	546
Retrieving the Public Key for Your Key Pair From Your Instance	546

Verifying Your Key Pair's Fingerprint	547
Deleting Your Key Pair	547
Connecting to Your Windows Instance if You Lose Your Private Key	548
Security Groups	548
Security Group Rules	549
Default Security Groups	551
Custom Security Groups	552
Working with Security Groups	552
Security Group Rules Reference	556
Controlling Access	562
Network Access to Your Instance	562
Amazon EC2 Permission Attributes	562
IAM and Amazon EC2	563
IAM Policies	564
IAM Roles	633
Network Access	640
Instance IP Addressing	643
Private IPv4 Addresses and Internal DNS Hostnames	643
Public IPv4 Addresses and External DNS Hostnames	644
Elastic IP Addresses (IPv4)	645
Amazon DNS Server	645
IPv6 Addresses	645
Working with IP Addresses for Your Instance	646
Multiple IP Addresses	650
Bring Your Own IP Addresses	657
Requirements	657
Prepare to Bring Your Address Range to Your AWS Account	658
Provision the Address Range for use with AWS	659
Advertise the Address Range through AWS	659
Deprovision the Address Range	659
Elastic IP Addresses	660
Elastic IP Address Basics	660
Working with Elastic IP Addresses	661
Using Reverse DNS for Email Applications	665
Elastic IP Address Limit	665
Network Interfaces	665
Network Interface Basics	666
IP Addresses Per Network Interface Per Instance Type	667
Scenarios for Network Interfaces	673
Best Practices for Configuring Network Interfaces	674
Working with Network Interfaces	675
Requester-Managed Network Interfaces	683
Enhanced Networking	684
Enhanced Networking Types	684
Enabling Enhanced Networking on Your Instance	684
Enhanced Networking: ENA	685
Enhanced Networking: Intel 82599 VF	688
Operating System Optimizations	691
Placement Groups	692
Cluster Placement Groups	693
Spread Placement Groups	693
Placement Group Rules and Limitations	693
Creating a Placement Group	694
Launching Instances in a Placement Group	695
Changing the Placement Group for an Instance	695
Deleting a Placement Group	696
Network MTU	697

Jumbo Frames (9001 MTU)	697
Path MTU Discovery	698
Check the Path MTU Between Two Hosts	698
Check and Set the MTU on Your Windows Instance	699
Troubleshooting	700
Virtual Private Clouds	700
Amazon VPC Documentation	700
EC2-Classic	701
Detecting Supported Platforms	701
Instance Types Available in EC2-Classic	702
Differences Between Instances in EC2-Classic and a VPC	703
Sharing and Accessing Resources Between EC2-Classic and a VPC	707
ClassicLink	708
Migrating from EC2-Classic to a VPC	721
Storage	731
Amazon EBS	732
Features of Amazon EBS	733
EBS Volumes	734
EBS Snapshots	786
EBS Optimization	807
EBS Encryption	815
EBS Volumes and NVMe	819
EBS Performance	820
EBS CloudWatch Events	836
Instance Store	843
Instance Store Lifetime	844
Instance Store Volumes	845
Add Instance Store Volumes	848
SSD Instance Store Volumes	850
Amazon EFS	851
Amazon S3	851
Amazon S3 and Amazon EC2	852
Instance Volume Limits	853
Linux-Specific Volume Limits	853
Windows-Specific Volume Limits	853
Instance Type Limits	854
Bandwidth versus Capacity	854
Device Naming	854
Available Device Names	854
Device Name Considerations	855
Block Device Mapping	855
Block Device Mapping Concepts	856
AMI Block Device Mapping	858
Instance Block Device Mapping	860
Mapping Disks to Volumes	864
Listing the Disks Using Windows Disk Management	865
Listing the Disks Using Windows PowerShell	866
Disk Device to Device Name Mapping	868
Using Public Data Sets	870
Public Data Set Concepts	870
Finding Public Data Sets	871
Creating a Public Data Set Volume from a Snapshot	871
Attaching and Mounting the Public Data Set Volume	872
Deploy Storage Spaces Direct	872
Step 1: Launch and Domain Join Instances	874
Step 2: Install and Configure Instance Prerequisites	876
Step 3: Create Failover Cluster	877

Step 4: Enable S2D	878
Step 5: Provision Storage	878
Step 6: Review the S2D Resources	879
Step 7: Clean Up	880
Additional Resources	880
Resources and Tags	881
Resource Locations	881
Resource IDs	882
Working with Longer IDs	883
Controlling Access to Longer ID Settings	886
Listing and Filtering Your Resources	887
Advanced Search	887
Listing Resources Using the Console	888
Filtering Resources Using the Console	889
Listing and Filtering Using the CLI and API	890
Tagging Your Resources	890
Tag Basics	891
Tagging Your Resources	892
Tag Restrictions	894
Tagging Your Resources for Billing	894
Working with Tags Using the Console	895
Working with Tags Using the CLI or API	898
Service Limits	899
Viewing Your Current Limits	900
Requesting a Limit Increase	901
Limits on Email Sent Using Port 25	901
Usage Reports	901
AWS Systems Manager for Microsoft System Center VMM	902
Features	902
Limitations	784
Requirements	903
Getting Started	903
Setting Up	903
Sign Up for AWS	903
Set Up Access for Users	904
Deploy the Add-In	906
Provide Your AWS Credentials	906
Managing EC2 Instances	907
Creating an EC2 Instance	907
Viewing Your Instances	909
Connecting to Your Instance	909
Rebooting Your Instance	910
Stopping Your Instance	910
Starting Your Instance	910
Terminating Your Instance	910
Importing Your VM	911
Prerequisites	911
Importing Your Virtual Machine	911
Checking the Import Task Status	912
Backing Up Your Imported Instance	913
Troubleshooting	913
Error: Add-in cannot be installed	913
Installation Errors	913
Checking the Log File	914
Errors Importing a VM	914
Uninstalling the Add-In	914
AWS Management Pack	916

Overview of AWS Management Pack for System Center 2012	916
Overview of AWS Management Pack for System Center 2007 R2	918
Downloading	919
System Center 2012	919
System Center 2007 R2	919
Deploying	920
Step 1: Installing the AWS Management Pack	920
Step 2: Configuring the Watcher Node	922
Step 3: Create an AWS Run As Account	922
Step 4: Run the Add Monitoring Wizard	925
Step 5: Configure Ports and Endpoints	929
Using	929
Views	930
Discoveries	944
Monitors	945
Rules	946
Events	946
Health Model	947
Customizing the AWS Management Pack	948
Upgrading	949
System Center 2012	949
System Center 2007 R2	950
Uninstalling	950
System Center 2012	950
System Center 2007 R2	951
Troubleshooting	951
Errors 4101 and 4105	951
Error 4513	951
Event 623	952
Events 2023 and 2120	952
Event 6024	952
General Troubleshooting for System Center 2012 — Operations Manager	952
General Troubleshooting for System Center 2007 R2	953
EC2Rescue for Windows Server	954
Using EC2Rescue for Windows Server GUI	954
Video Walkthrough	956
Analyzing an Offline Instance	956
Collecting Data from an Active Instance	957
Using EC2Rescue for Windows Server with the Command Line	957
Collect Action	957
Rescue Action	959
Restore Action	961
Using EC2Rescue for Windows Server with Systems Manager Run Command	962
Examples	963
Troubleshooting	965
Troubleshooting Launch Issues	965
Instance Limit Exceeded	965
Insufficient Instance Capacity	966
Instance Terminates Immediately	966
Troubleshoot an Unreachable Instance	967
How to Take a Screenshot of an Unreachable Instance	968
Common Screenshots	968
Resetting a Lost or Expired Windows Administrator Password	974
Reset Using EC2Config	975
Reset Using EC2Launch	979
Common Issues	982
EBS volumes don't initialize on Windows Server 2016 AMIs	982

Boot an EC2 Windows Instance into Directory Services Restore Mode (DSRM)	983
Unable to remotely log on to an instance with a user account that is not an Administrator	984
High CPU usage shortly after Windows starts	984
No console output	985
Remote Desktop can't connect to the remote computer	986
.....	986
Error Using macOS RDP Client	988
RDP displays a black screen instead of the desktop	988
Instance loses network connectivity or scheduled tasks don't run when expected	989
Windows Server 2012 R2 not available on the network	989
Troubleshooting Remote Desktop Connection issues using AWS Systems Manager	989
AWSSupport-TroubleshootRDP	989
AWSSupport-ExecuteEC2Rescue	992
Common Messages	992
>Password is not available"	992
>Password not available yet"	993
=Cannot retrieve Windows password"	993
="Waiting for the metadata service"	993
="Unable to activate Windows"	996
="Windows is not genuine (0x80070005)"	997
="No Terminal Server License Servers available to provide a license"	998
Document History	999
AWS Glossary	1020

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](#).

Amazon EC2 enables you to run any compatible Windows-based solution on our high-performance, reliable, cost-effective, cloud computing platform. For more information, see [Windows Server on AWS](#).

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 \(p. 12\)](#)
- [Getting Started with Amazon EC2 Windows Instances \(p. 20\)](#)

Basics

- [Amazon EC2 Basic Infrastructure for Windows \(p. 4\)](#)
- [Instance Types \(p. 110\)](#)
- [Tags \(p. 890\)](#)

Networking and Security

- [Amazon EC2 Key Pairs and Windows Instances \(p. 543\)](#)
- [Security Groups \(p. 548\)](#)
- [Elastic IP Addresses \(p. 660\)](#)
- [Amazon EC2 and Amazon VPC \(p. 700\)](#)

Storage

- [Amazon EBS \(p. 732\)](#)
- [Instance Store \(p. 843\)](#)

Working with Windows Instances

- [Remote Management \(Run Command\)](#)
- [Differences between Windows Server and an Amazon EC2 Windows Instance \(p. 9\)](#)
- [Designing Your Applications to Run on Amazon EC2 Windows Instances \(p. 10\)](#)
- [Getting Started with AWS: Hosting a .NET Web App](#)

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 hour, 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](#).

Amazon EC2 Basic Infrastructure for Windows

As you get started with Amazon EC2, you'll benefit from understanding the components of its basic infrastructure and how they compare or contrast with your own data centers.

Concepts

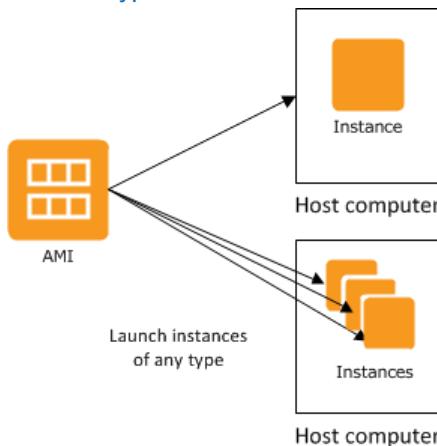
- [Amazon Machine Images and Instances \(p. 4\)](#)
- [Regions and Availability Zones \(p. 5\)](#)
- [Storage \(p. 6\)](#)
- [Root Device Volume \(p. 7\)](#)
- [Networking and Security \(p. 8\)](#)
- [AWS Identity and Access Management \(p. 8\)](#)
- [Differences between Windows Server and an Amazon EC2 Windows Instance \(p. 9\)](#)
- [Designing Your Applications to Run on Amazon EC2 Windows Instances \(p. 10\)](#)

Amazon Machine Images and Instances

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 *instances*, which are copies of the AMI running as virtual servers in the cloud.

Amazon publishes many 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 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.

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 facilities. Select an instance type based on the amount of memory and computing power that you need for the applications 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](#). You can also launch multiple instances from an AMI, as shown in the following figure.



Your Windows 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.

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.

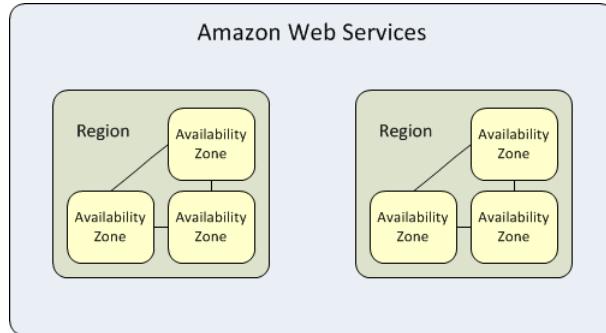
Bare Metal Instances

Amazon now offers EC2 bare metal instances which are available with the EC2 Storage Optimized [Amazon EC2 I3 instance family](#) in the form of i3.metal instances. Bare metal instances allow you to run applications that benefit from deep performance analysis tools, specialized workloads that require direct access to bare metal infrastructure, legacy workloads not supported in virtual environments, licensing-restricted Tier 1 business critical applications, as well as virtualization-secured containers.

Regions and Availability Zones

Amazon has data centers in different areas of the world (for example, North America, Europe, and Asia). Correspondingly, Amazon EC2 is available to use in different *regions*. By launching instances in separate regions, you can design your application to be closer to specific customers or to meet legal or other requirements. Prices for Amazon EC2 usage vary by region (for more information about pricing by region, see [Amazon EC2 Pricing](#)).

Each region contains multiple distinct locations called *Availability Zones*. Each Availability Zone is engineered to be isolated from failures in other Availability Zones, and to provide inexpensive, low-latency network connectivity to other zones in the same region. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location.



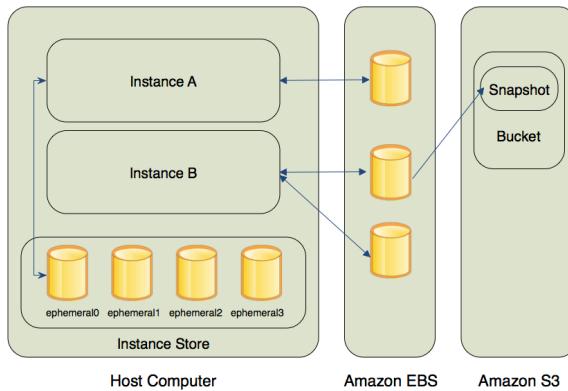
For more information about the available regions and Availability Zones, see [Using Regions and Availability Zones](#) in the *Amazon EC2 User Guide for Linux Instances*.

Storage

When using Amazon EC2, you may have data that you need to store. Amazon EC2 offers the following storage options:

- [Amazon Elastic Block Store \(Amazon EBS\)](#)
- [Amazon EC2 Instance Store \(p. 843\)](#)
- [Amazon Simple Storage Service \(Amazon S3\)](#)

The following figure shows the relationship between these types of storage.

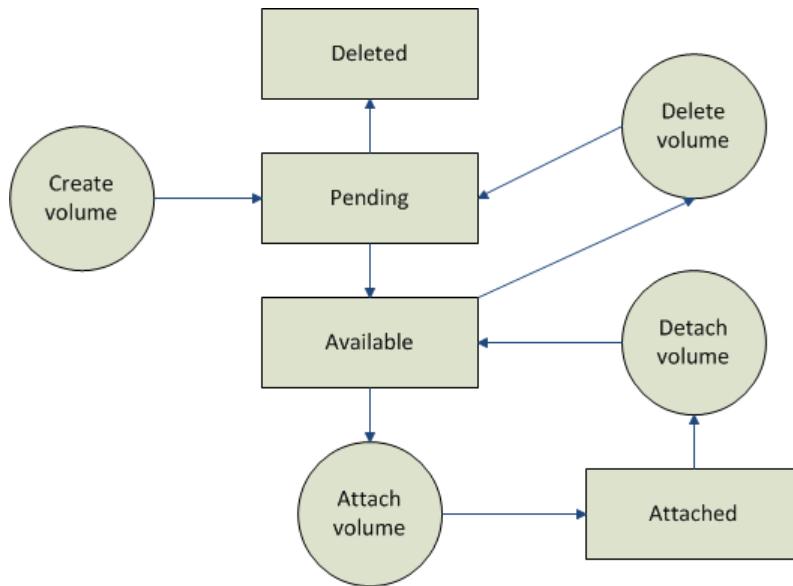


Amazon EBS Volumes

Amazon EBS volumes are the recommended storage option for the majority of use cases. Amazon EBS provides your instances with persistent, block-level storage. Amazon EBS volumes are essentially hard disks that you can attach to a running instance.

Amazon EBS is especially suited for applications that require a database, a file system, or access to raw block-level storage.

As illustrated in the previous figure, you can attach multiple volumes to an instance. Also, 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 a new Amazon EBS volume from a snapshot, and attach it to another instance. You can also detach a volume from an instance and attach it to a different instance. The following figure illustrates the life cycle of an EBS volume.



For more information about Amazon EBS volumes, see [Amazon Elastic Block Store \(p. 732\)](#).

Instance Store

All instance types, with the exception of Micro instances, offer *instance store*, which provides your instances with temporary, block-level storage. This is storage that is physically attached to the host computer. The data on an instance store volume doesn't persist when the associated instance is stopped or terminated. For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 843\)](#).

Instance store is an option for inexpensive temporary storage. You can use instance store volumes if you don't require data persistence.

Amazon S3

Amazon S3 is storage for the Internet. It provides a simple web service interface that enables you to store and retrieve any amount of data from anywhere on the web. For more information about Amazon S3, see the [Amazon S3 product page](#).

Root Device Volume

When you launch an instance, the *root device volume* contains the image used to boot the instance. When you launch a Windows instance, a root EBS volume is created from an EBS snapshot and attached to the instance.

By default, the root volume is deleted when the instance terminates (the `DeleteOnTermination` attribute is `true`). Using the console, you can change `DeleteOnTermination` 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, click **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose the AMI to use and click **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 click **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**.

To change the root device volume of an instance to persist 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)

Networking and Security

By default, an instance is assigned public IPv4 address only if it's launched into a default VPC. An instance that's launched into a nondefault VPC must be specifically assigned a public IPv4 address at launch, or you must modify your subnet's default public IPv4 addressing behavior.

Instances can fail or terminate for reasons outside of your control. If one fails and you launch a replacement instance, the replacement has a different public IPv4 address than the original. However, if your application needs a static IPv4 address, Amazon EC2 offers *Elastic IP addresses*. For more information, see [Amazon EC2 Instance IP Addressing \(p. 643\)](#).

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. For more information, see [Amazon EC2 Security Groups for Windows Instances \(p. 548\)](#).

AWS Identity and Access Management

AWS Identity and Access Management (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.

For more information about IAM, see the following:

- [Creating an IAM Group and Users \(p. 563\)](#)
- [IAM Policies for Amazon EC2 \(p. 564\)](#)
- [IAM Roles for Amazon EC2 \(p. 633\)](#)
- [Identity and Access Management \(IAM\)](#)
- [IAM User Guide](#)

Differences between Windows Server and an Amazon EC2 Windows Instance

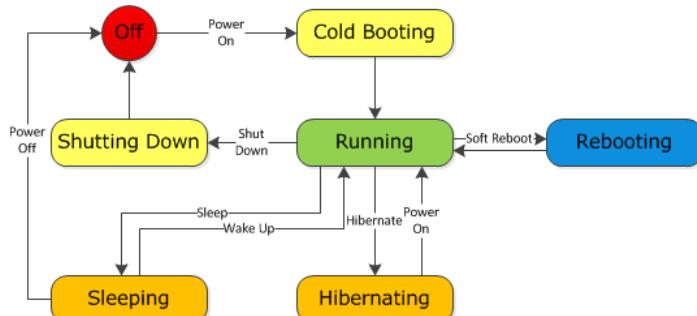
After you launch your Amazon EC2 Windows instance, it behaves like a traditional server running Windows Server. For example, both Windows Server and an Amazon EC2 instance can be used to run your web applications, conduct batch processing, or manage applications requiring large-scale computations. However, there are important differences between the server hardware model and the cloud computing model. The way an Amazon EC2 instance runs is not the same as the way a traditional server running Windows Server runs.

Before you begin launching Amazon EC2 Windows instances, you should be aware that the architecture of applications running on cloud servers can differ significantly from the architecture for traditional application models running on your hardware. Implementing applications on cloud servers requires a shift in your design process.

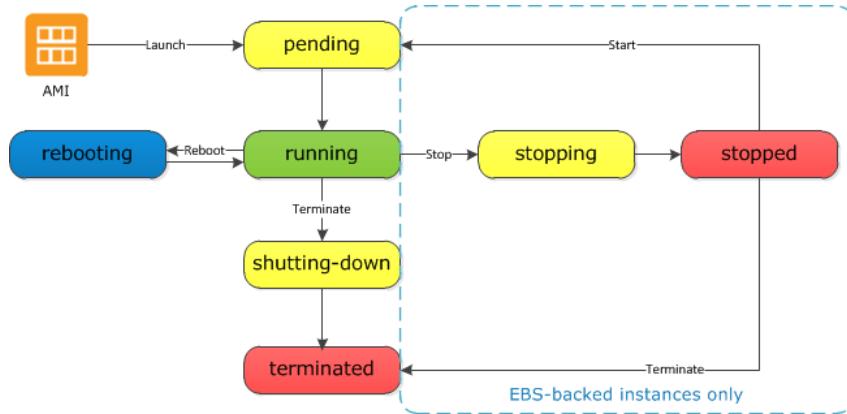
The following table describes some key differences between Windows Server and an Amazon EC2 Windows instance.

Windows Server	Amazon EC2 Windows Instance
Resources and capacity are physically limited.	Resources and capacity are scalable.
You pay for the infrastructure, even if you don't use it.	You pay for the usage of the infrastructure. We stop charging you for the instance as soon as you stop or terminate it.
Occupies physical space and must be maintained on a regular basis.	Doesn't occupy physical space and does not require regular maintenance.
Starts with push of the power button (known as <i>cold booting</i>).	Starts with the launch of the instance.
You can keep the server running until it is time to shut it down, or put it in a sleep or hibernation state (during which the server is powered down).	You can keep the server running, or stop and restart it (during which the instance is moved to a new host computer).
When you shut down the server, all resources remain intact and in the state they were in when you switched it off. Information you stored on the hard drives persists and can be accessed whenever it's needed. You can restore the server to the running state by powering it on.	When you terminate the instance, its infrastructure is no longer available to you. You can't connect to or restart an instance after you've terminated it. However, you can create an image from your instance while it's running, and launch new instances from the image at any time.

A traditional server running Windows Server goes through the states shown in the following diagram.



An Amazon EC2 Windows instance is similar to the traditional Windows Server, as you can see by comparing the following diagram with the previous diagram for Windows Server. After you launch an instance, it briefly goes into the pending state while registration takes place, then it goes into the running state. The instance remains active until you stop or terminate it. You can't restart an instance after you terminate it. You can create a backup image of your instance while it's running, and launch a new instance from that backup image.



Designing Your Applications to Run on Amazon EC2 Windows Instances

It is important that you consider the differences mentioned in the previous section when you design your applications to run on Amazon EC2 Windows instances.

Applications built for Amazon EC2 use the underlying computing infrastructure on an as-needed basis. They draw on necessary resources (such as storage and computing) on demand in order to perform a job, and relinquish the resources when done. In addition, they often dispose of themselves after the job is done. While in operation, the application scales up and down elastically based on resource requirements. An application running on an Amazon EC2 instance can terminate and recreate the various components at will in case of infrastructure failures.

When designing your Windows applications to run on Amazon EC2, you can plan for rapid deployment and rapid reduction of compute and storage resources, based on your changing needs.

When you run an Amazon EC2 Windows instance, you don't need to provision the exact system package of hardware, software, and storage, the way you do with Windows Server. Instead, you can focus on using a variety of cloud resources to improve the scalability and overall performance of your Windows application.

With Amazon EC2, designing for failure and outages is an integral and crucial part of the architecture. As with any scalable and redundant system, architecture of your system should account for computing, network, and storage failures. You have to build mechanisms in your applications that can handle different kinds of failures. The key is to build a modular system with individual components that are not tightly coupled, can interact asynchronously, and treat one another as black boxes that are independently scalable. Thus, if one of your components fails or is busy, you can launch more instances of that component without breaking your current system.

Another key element to designing for failure is to distribute your application geographically. Replicating your application across geographically distributed regions improves high availability in your system.

Amazon EC2 infrastructure is programmable and you can use scripts to automate the deployment process, to install and configure software and applications, and to bootstrap your virtual servers.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Designing Your Applications to Run
on Amazon EC2 Windows Instances

You should implement security in every layer of your application architecture running on an Amazon EC2 Windows instance. If you are concerned about storing sensitive and confidential data within your Amazon EC2 environment, you should encrypt the data before uploading it.

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 \(p. 12\)](#)
2. [Create an IAM User \(p. 12\)](#)
3. [Create a Key Pair \(p. 14\)](#)
4. [Create a Virtual Private Cloud \(VPC\) \(p. 16\)](#)
5. [Create a Security Group \(p. 17\)](#)

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

If you previously signed in to the AWS Management Console using AWS account root user credentials, choose **Sign in to a different account**. If you previously signed in to the console using IAM credentials, choose **Sign-in using root account credentials**. 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 verification code 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 \(p. 563\)](#).

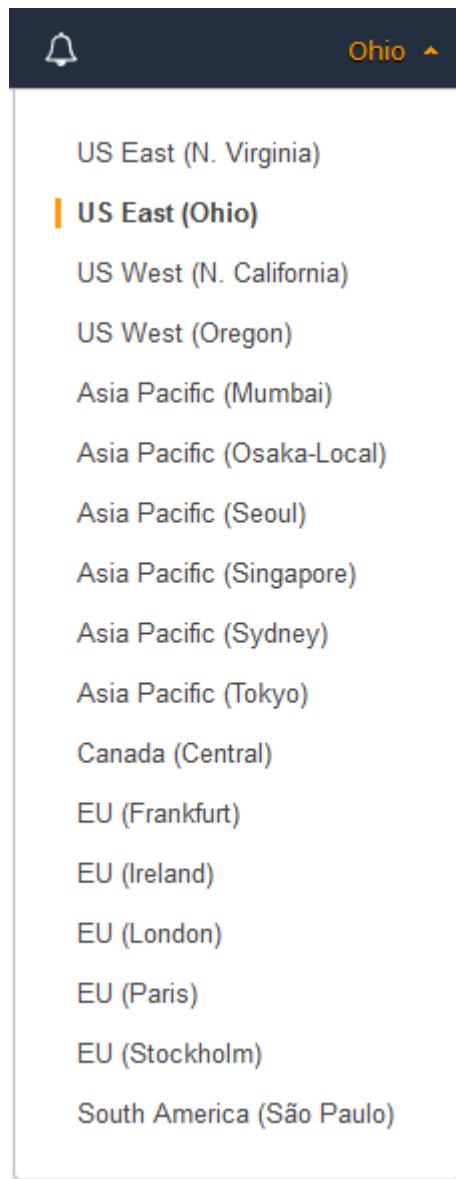
Create a Key Pair

AWS uses public-key cryptography to secure the login information for your instance. You specify the name of the key pair when you launch your instance, then provide the private key to obtain the administrator password for your Windows instance so you can log in using RDP.

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 \(p. 5\)](#).

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.

For more information, see [Amazon EC2 Key Pairs and Windows Instances \(p. 543\)](#).

Create a Virtual Private Cloud (VPC)

Amazon VPC enables you to launch AWS resources into a virtual network that you've defined, known as a *virtual private cloud* (VPC). The newer EC2 instance types require that you launch your instances in a VPC. If you have a default VPC, you can skip this section and move to the next task, [Create a Security Group \(p. 17\)](#). To determine whether you have a default VPC, open the Amazon EC2 console and look for **Default VPC** under **Account Attributes** on the dashboard. If you do not have a default VPC listed on the dashboard, you can create a nondefault VPC using the steps below.

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 **Launch VPC Wizard**.
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 VPCs, see 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 RDP. 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 \(p. 5\)](#).

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.

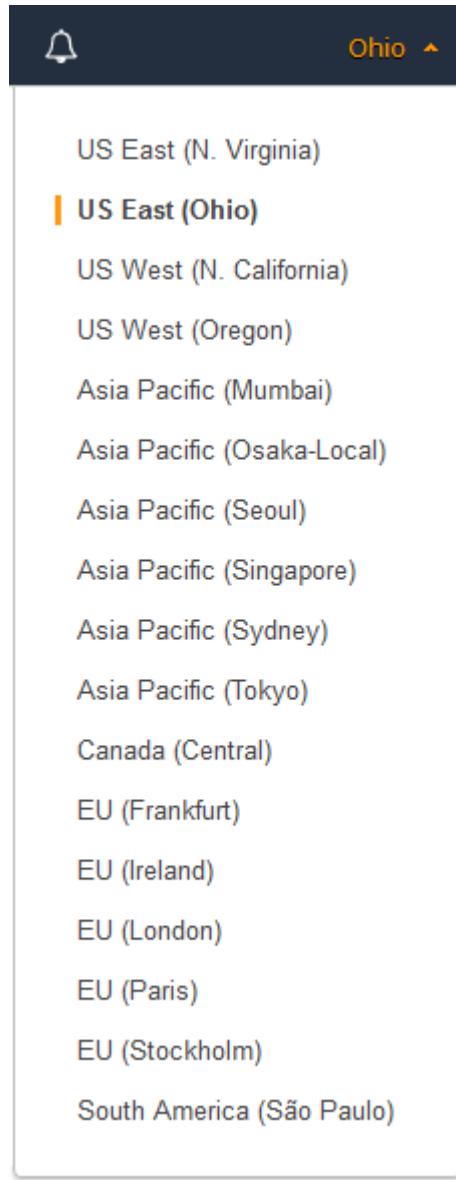
To create a security group with least privilege

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 (*).
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 **RDP** 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 RDP 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 Windows Instances \(p. 548\)](#).

Getting Started with Amazon EC2 Windows Instances

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

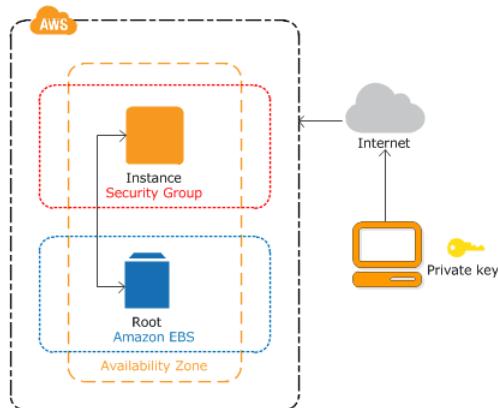
When you sign up for AWS, you can get started with Amazon EC2 using the [AWS Free Tier](#). If you created your AWS account less than 12 months ago, and have not already exceeded the free tier benefits for Amazon EC2, it will not cost you anything to complete this tutorial, because we help you select 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 \(p. 20\)](#)
- [Prerequisites \(p. 21\)](#)
- [Step 1: Launch an Instance \(p. 21\)](#)
- [Step 2: Connect to Your Instance \(p. 22\)](#)
- [Step 3: Clean Up Your Instance \(p. 23\)](#)
- [Next Steps \(p. 24\)](#)

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 \(p. 21\)](#)
2. [Connect to Your Instance \(p. 22\)](#)
3. [Clean Up Your Instance \(p. 23\)](#)

Related Tutorials

- If you'd prefer to launch a Linux instance, see this tutorial in the [Amazon EC2 User Guide for Linux Instances: Getting Started with Amazon EC2 Linux 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 \(p. 12\)](#).

Step 1: Launch an Instance

You can launch a Windows 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 the AMI for Windows Server 2016 Base or Windows Server 2012 R2 Base. 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.
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

To connect to a Windows instance, you must retrieve the initial administrator password and then specify this password when you connect to your instance using Remote Desktop.

The name of the administrator account depends on the language of the operating system. For example, for English, it's Administrator, for French it's Administrateur, and for Portuguese it's Administrador. For more information, see [Localized Names for Administrator Account in Windows](#) in the Microsoft TechNet Wiki.

If you've joined your instance to a domain, you can connect to your instance using domain credentials you've defined in AWS Directory Service. On the Remote Desktop login screen, instead of using the local computer name and the generated password, use the fully-qualified user name for the administrator (for example, `corp.example.com\Admin`) and the password for this account.

The license for the Windows Server operating system (OS) allows two simultaneous remote connections for administrative purposes. The license for Windows Server is included in the price of your Windows instance. If you need more than two simultaneous remote connections, you must purchase a Remote Desktop Services (RDS) license. If you attempt a third connection, an error occurs. For more information, see [Configure the Number of Simultaneous Remote Connections Allowed for a Connection](#).

To connect to your Windows instance using an RDP client

1. In the Amazon EC2 console, select the instance, and then choose **Connect**.
2. In the **Connect To Your Instance** dialog box, choose **Get Password** (it will take a few minutes after the instance is launched before the password is available).
3. Choose **Browse** and navigate to the private key file you created when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file into the **Contents** field.
4. Choose **Decrypt Password**. The console displays the default administrator password for the instance in the **Connect To Your Instance** dialog box, replacing the link to **Get Password** shown previously with the actual password.
5. Record the default administrator password, or copy it to the clipboard. You need this password to connect to the instance.
6. Choose **Download Remote Desktop File**. Your browser prompts you to either open or save the .rdp file. Either option is fine. When you have finished, you can choose **Close** to dismiss the **Connect To Your Instance** dialog box.

- If you opened the .rdp file, you'll see the **Remote Desktop Connection** dialog box.
 - If you saved the .rdp file, navigate to your downloads directory, and open the .rdp file to display the dialog box.
7. You may get a warning that the publisher of the remote connection is unknown. You can continue to connect to your instance.
 8. When prompted, log in to the instance, using the administrator account for the operating system and the password that you recorded or copied previously. If your **Remote Desktop Connection** already has an administrator account set up, you might have to choose the **Use another account** option and type the user name and password manually.

Note

Sometimes copying and pasting content can corrupt data. If you encounter a "Password Failed" error when you log in, try typing in the password manually.

9. Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. Use the following steps to verify the identity of the remote computer, or simply choose **Yes** or **Continue** to continue if you trust the certificate.
 - a. If you are using **Remote Desktop Connection** from a Windows PC, choose **View certificate**. If you are using **Microsoft Remote Desktop** on a Mac, choose **Show Certificate**.
 - b. Choose the **Details** tab, and scroll down to the **Thumbprint** entry on a Windows PC, or the **SHA1 Fingerprints** entry on a Mac. This is the unique identifier for the remote computer's security certificate.
 - c. In the Amazon EC2 console, select the instance, choose **Actions**, and then choose **Get System Log**.
 - d. In the system log output, look for an entry labeled **RDPMESSAGE-THUMBPRINT**. If this value matches the thumbprint or fingerprint of the certificate, you have verified the identity of the remote computer.
 - e. If you are using **Remote Desktop Connection** from a Windows PC, return to the **Certificate** dialog box and choose **OK**. If you are using **Microsoft Remote Desktop** on a Mac, return to the **Verify Certificate** and choose **Continue**.
 - f. [Windows] Choose **Yes** in the **Remote Desktop Connection** window to connect to your instance.

[Mac OS] Log in as prompted, using the default administrator account and the default administrator password that you recorded or copied previously. Note that you might need to switch spaces to see the login screen. For more information about spaces, see support.apple.com/en-us/HT204100.

- g. If you receive an error while attempting to connect to your instance, see [Remote Desktop can't connect to the remote computer \(p. 986\)](#).

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 \(p. 24\)](#).

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 \(p. 39\)](#) 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 \(p. 749\)](#) and [Attaching an Amazon EBS Volume to an Instance \(p. 752\)](#).
- Install the WAMP or WIMP stack. For more information, see [Tutorial: Installing a WAMP Server on an Amazon EC2 Instance Running Windows Server \(p. 30\)](#) and [Tutorial: Installing a WIMP Server on an Amazon EC2 Instance Running Windows Server \(p. 33\)](#).

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 \(p. 549\)](#).
- Regularly patch, update, and secure the operating system and applications on your instance. For more information about updating Amazon Linux 2 or the Amazon Linux AMI, see [Managing Software on Your Linux Instance](#) in the *Amazon EC2 User Guide for Linux Instances*. For more information about updating your Windows instance, see [Updating Your Windows Instance](#).

Storage

- 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 \(p. 371\)](#).
- 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 \(p. 452\)](#) and [Tagging Your Amazon EC2 Resources \(p. 890\)](#).
- 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 \(p. 899\)](#).

Backup and Recovery

- Regularly back up your EBS volumes using [Amazon EBS snapshots \(p. 786\)](#), and create an [Amazon Machine Image \(AMI\) \(p. 51\)](#) 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 \(p. 643\)](#).
- Monitor and respond to events. For more information, see [Monitoring Amazon EC2 \(p. 487\)](#).
- 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 \(p. 665\)](#). 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 Windows Server

The following tutorials show you how to perform common tasks using EC2 instances running Windows Server.

Tutorials

- [Tutorial: Deploying a WordPress Blog on Your Amazon EC2 Instance Running Windows Server \(p. 26\)](#)
- [Tutorial: Installing a WAMP Server on an Amazon EC2 Instance Running Windows Server \(p. 30\)](#)
- [Tutorial: Installing a WIMP Server on an Amazon EC2 Instance Running Windows Server \(p. 33\)](#)
- [Tutorial: Increase the Availability of Your Application on Amazon EC2 \(p. 36\)](#)
- [Tutorial: Remotely Manage Your Amazon EC2 Instances \(p. 39\)](#)
- [Tutorial: Setting Up a Windows HPC Cluster on Amazon EC2 \(p. 43\)](#)

Tutorial: Deploying a WordPress Blog on Your Amazon EC2 Instance Running Windows Server

This tutorial will help you install and deploy a WordPress blog on an Amazon EC2 instance running Windows Server.

If you'd prefer to host your WordPress blog on a Linux instance, see [Tutorial: Hosting a WordPress Blog with Amazon EC2](#) in the *Amazon EC2 User Guide for Linux 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*.

Prerequisites

Before you get started, be sure that you do the following:

- Launch an Amazon EC2 instance from a Windows Server AMI. For information, see [Getting Started with Amazon EC2 Windows Instances \(p. 20\)](#).
- Use the AWS free usage tier (if eligible) to launch and use the free Windows t2.micro instance for 12 months. You can use the AWS free usage tier for launching new applications, testing existing applications, or simply gaining hands-on experience with AWS. For more information about eligibility and the highlights, see the [AWS Free Usage Tier](#) product page.

Important

If you've launched a regular instance and use it to deploy the WordPress website, you will incur the standard Amazon EC2 usage fees for the instance until you terminate it. For more information about Amazon EC2 usage rates, go to the [Amazon EC2 product page](#).

- Ensure that the security group in which you're launching your instance has ports 80 (HTTP), 443 (HTTPS), and 3389 (RDP) open for inbound traffic. Ports 80 and 443 allow computers outside of the instance to connect with HTTP and HTTPS. If these ports are not open, the WordPress site can't be accessed from outside the instance. Port 3389 allows you to connect to the instance with Remote Desktop Protocol.
- Connect to your instance.

Installing the Microsoft Web Platform Installer

You can use the Microsoft Web Platform Installer to install and configure WordPress on your server. This tool simplifies deployment of Web applications and Web sites to IIS servers. For more information, see [Microsoft Web Platform Installer](#).

To install Microsoft Web Platform Installer

1. Verify that you've met the conditions in [Prerequisites \(p. 26\)](#).
2. Connect to your instance.
3. Disable Internet Explorer Enhanced Security Configuration so that you can download and install required software from the web.
 - a. Open Server Manager.
 - On Windows Server 2008 R2, under **Server Summary**, in the **Security Information** section, click **Configure IE ESC**.
 - On Windows Server 2012 R2, click **Local Server** in the left pane. In the **Properties** pane, locate **IE Enhanced Security Configuration**. Click **On**.
 - b. Under **Administrators**, click **Off**, and then click **OK**.
 - c. Close Server Manager.
 - d. Make a note to re-enable Internet Explorer Enhanced Security Configuration when you have finished installing software from the web.
4. Download and install the latest version of the [Microsoft Web Platform Installer](#).

Installing WordPress

Now you'll use the Web Platform Installer to deploy WordPress on your server.

To install WordPress

1. [Download](#) and install Visual C++ Redistributable for Visual Studio 2012 Update 4 or later.

Important

Even if your operating system is a 64-bit operating system, you must install the 32-bit version as PHP will not run in 64 bit.

Open the **Web Platform Installer** and click **Applications**.

2. Select **WordPress**, click **Add**, and then click **Install**.
3. On the **Prerequisites** page, select **MySQL** for the database to use. Enter the desired administrator password for your MySQL database in the **Password** and **Re-type Password** boxes, and then click **Continue**.

For more information about creating a secure password, see <https://identitysafe.norton.com/password-generator/>. Do not reuse an existing password, and make sure to store this password in a safe place.

4. Click **I Accept** for the list of third-party application software, Microsoft products (including the IIS web server), and components. After the Web Platform Installer finishes installing the software, you are prompted to configure your new site.
5. On the **Configure** page, clear the default application name in the **'WordPress' application name:** box and leave it blank, then leave the default information in the other boxes and click **Continue**.
6. Click **Yes** to accept that the contents of the folder will be overwritten.

Configuring Security Keys

WordPress allows you to generate and enter unique authentication keys and salts for your site. 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.

For more information about security keys, see http://codex.wordpress.org/Editing_wp-config.php#Security_Keys.

To configure security keys

1. 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 the installation wizard. The following steps will show you how to modify these values in Notepad to work with a Windows installation.
2. Copy all of the text in that page to your clipboard. It should look similar to the example below.

Note

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

```
define('AUTH_KEY',         '3#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 ?60P$eJT@;
+(ndLg');
define('LOGGED_IN_KEY',    'ju}qwre3V*+8f_zOWF?{LlGsQ]Ye@2Jh^,8x>)Y |;(^[Iw]Pi+LG#A4R?
7N`YB3');
define('NONCE_KEY',        'P(g62HeZxEes/LnI^i=H,[XwK9I&[2s : ?ON)VJM%?;v2v]v+;
+^9eXuahg@::Cj');
define('AUTH_SALT',         'C$Dp24Hj[JK:?:ql`sRVA:{:7yShy(9A@5wg+`JJVb1fk%_-Bx*M4(qc[Qg
%JT!h');
define('SECURE_AUTH_SALT', 'd!uRu#)+q#{f$Z?Z9uFPG.${+S{n~1M&%@-gL>U>NV<zpd-@2-Es7Q1O-
bp28EKv');
define('LOGGED_IN_SALT',   'j{00P*owZf)kVD+FVLn~~ >. | Y%Ug4#I^*Lv9QeZ^&XmK/e(76miC+&W&
+^OP/');
define('NONCE_SALT',       '-97r*V/cgxLmp?Zy4zUU4r99QO_rGs2LTd%P; |
_e1tS)8_B/, .6[=UK<J_y9?JWG');
```

3. Open a Notepad window by clicking **Start, All Programs, Accessories**, and then **Notepad**.
4. Paste the copied text into the Notepad window.
5. Windows WordPress installations do not accept the dollar sign (\$) in key and salt values, so they need to be replaced with another character (such as S). In the Notepad window, click **Edit**, then click **Replace**.
6. In the **Find what** box, type **\$**.
7. In the **Replace with** box, type **S**.
8. Click **Replace All** to replace all of the dollar signs with S characters.
9. Close the **Replace** window.
10. Paste the modified key and salt values from the Notepad window into their corresponding boxes in the installation wizard. For example, the AUTH_KEY value in the Notepad window should be pasted into the **Authentication Key** box in the wizard.

Do not include the single quotes or other text surrounding the values, just the actual value as in the example shown below.

The modified AUTH_KEY line from the Notepad window:

```
define('AUTH_KEY',         '3#USS+[RXN8:b^-L_0(WU_+ c+WFKI~c]o]-bHw+)/
Aj[wTwSiZ<Qb[mghEXcRh-');
```

Paste this text into the **Authentication Key** box of the wizard:

```
3#USS+[RXN8:b^-L_0(WU_+ c+WFKI-c]o]-bHw+)/Aj[wTwsIZ<Qb[mghEXcRh-
```

11. Click **Continue** and **Finish** to complete the Web Platform Installer wizard.

Configuring the Site Title and Administrator

When you complete the Web Platform Installer wizard, a browser window opens to your WordPress installation at <http://localhost/wp-admin/install.php>. On this page, you configure the title for your site and an administrative user to moderate your blog.

To complete the installation

1. On the WordPress **Welcome** page, enter the following information and click **Install WordPress**.

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 this will be more difficult to exploit than the default user name, admin.
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.
Privacy	Check to allow search engines to index your site.

2. Click **Log In**.
3. On the **Log In** page, enter your user name for **Username** and the site password you entered previously for **Password**.

Making Your WordPress Site Public

Now that you can see your WordPress blog on your local host, you can publish this website as the default site on your instance so that other people can see it. The next procedure walks you through the process of modifying your WordPress settings to point to the public DNS name of your instance instead of your local host.

To configure the default settings for your WordPress site

1. Open the WordPress dashboard by opening a browser on your instance and going to <http://localhost/wp-admin>. If prompted for your credentials, enter your user name for the **Username** and your site password for **Password**.
2. In the **Dashboard** pane, click **Settings**.

3. On the **General Settings** page, enter the following information and click **Save Changes**.
 - **WordPress address (URL)**—The public DNS address of your instance. For example, your URL may look something like `http://ec2-203-0-113-25.compute-1.amazonaws.com`. You can get the public DNS for your instance using the Amazon EC2 console (select the instance and check the **Public DNS** column; if this column is hidden, click the **Show/Hide** icon and select **Public DNS**).
 - **Site address (URL)**—The same public DNS address of your instance that you set in **WordPress address (URL)**.
4. To see your new site, open a browser on a computer other than the instance hosting WordPress and type the public DNS address of your instance in the web address field. Your WordPress site appears.

Congratulations! You have just deployed a WordPress site on a Windows instance.

Next Steps

If you no longer need this instance, you can remove it to avoid incurring charges. For more information, see [Clean Up Your Instance \(p. 23\)](#).

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 Windows \(p. 773\)](#).
- Move your MySQL database to [Amazon RDS](#) to take advantage of the service's ability to scale automatically.
- Migrate to a larger instance type. For more information, see [Changing the Instance Type \(p. 182\)](#).
- Add additional instances. For more information, see [Tutorial: Increase the Availability of Your Application on Amazon EC2 \(p. 36\)](#).

For information about WordPress, see the WordPress Codex help documentation at <http://codex.wordpress.org/>. For more information about troubleshooting your installation, see http://codex.wordpress.org/Installing_WordPress#Common_Installation_Problems. For information about making your WordPress blog more secure, see http://codex.wordpress.org/Hardening_WordPress. For information about keeping your WordPress blog up-to-date, see http://codex.wordpress.org/Updating_WordPress.

Tutorial: Installing a WAMP Server on an Amazon EC2 Instance Running Windows Server

This tutorial shows you how to install an Apache web server with PHP and MySQL on an EC2 instance running Windows Server. This software configuration is sometimes called a WAMP server or WAMP stack (Windows, Apache, MySQL, PHP). For information about how to create a similar server on Linux, see [Tutorial: Installing a LAMP Web Server](#) in the *Amazon EC2 User Guide for Linux Instances*.

A WAMP stack is designed for easy installation to help developers get up and running quickly. It is not designed for production environments for the following reasons:

- The default configurations do not meet security requirements for most production environments.
- Upgrading and patching the different software components on a single production server would affect server availability.

- The WAMP one-click installers do not place files in standard locations, which can make it difficult to locate important configuration files.

You can, however, create a WAMP stack on an EC2 instance to prototype a web project in a controlled test environment. For example, you can host a static website or deploy a dynamic PHP application that reads and writes information to a database.

There are many third-party solutions that you can use to install a WAMP stack; this tutorial uses the Bitnami WAMP stack. For more information, see [Review: WAMP stacks for Web developers](#).

Prerequisites

- Provision a Windows Server 2008 R2 or 2012 R2 base instance. You must configure the base instance with a public domain name system (DNS) name that is reachable from the Internet. For more information, see [Getting Started with Amazon EC2 Windows Instances \(p. 20\)](#).
- Verify that the security group for your instance has the following ports open:
 - Port 80 (HTTP inbound and outbound) - Allows computers outside of the instance to connect by using HTTP.
 - Port 443 (HTTPS inbound and outbound) - Allows computers outside of the instance to connect by using HTTPS.
 - Port 3389 (RDP inbound only) - Allows you to connect to the instance using Remote Desktop Protocol (RDP). As a security best practice, restrict RDP access to a range of IP addresses in your organization.

To install a WAMP server

1. Connect to your instance using Microsoft Remote Desktop. For more information, see [Connecting to Your Windows Instance \(p. 359\)](#).
2. Disable Internet Explorer Enhanced Security Configuration so that you can download and install required software from the web.
 - a. From the instance, open Server Manager.
 - b. [Windows Server 2008 R2] Under **Server Summary, Security Information**, click **Configure IE ESC**.

[Windows Server 2012 R2] Click **Local Server** in the left pane. In the **Properties** pane, locate **IE Enhanced Security Configuration**. Click **On**.

 - c. Under **Administrators**, click **Off**, and then click **OK**.
 - d. Close Server Manager.
 - e. Make a note to re-enable Internet Explorer Enhanced Security Configuration when you have finished installing software from the web.
3. Install software updates to ensure that the instance has the latest security updates and bug fixes.
 - **EC2Config** - [Download](#) and install the latest version of the EC2Config service. For more information, see [Installing the Latest Version of EC2Config \(p. 384\)](#).
 - **Windows Update** - Run Windows Update to ensure that the latest security and software updates are installed on the instance. In Control Panel, click **System and Security**. In the **Windows Update** section, click **Check for updates**.
4. Download and install the WAMP stack. For the purposes of this tutorial, we suggest that you download and install [this WAMP stack](#). You can, however, download and install [other Bitnami WAMP stacks](#). Regardless of which stack you install, the Bitnami site prompts you to either create a free Bitnami account or log in by using a social media account. After you log in, run the Bitnami setup wizard.

5. After setup completes, verify that the Apache web server is configured properly and running by browsing to a test page. Open a web browser on a different computer and enter either the public DNS address of the WAMP server or the public IP address. The public DNS address for your instance is listed on the Amazon EC2 console in the **Public DNS** column. If this column is hidden, click the **Show/Hide** icon and select **Public DNS**.

Important

If you do not see the Bitnami test page, use Windows Firewall with Advanced Security to create a custom rule that allows the HTTP protocol through port 80 and the HTTPS protocol through port 443. For more information, see [Network Security](#) on Microsoft TechNet. Also verify that the security group for your instances contains a rule to allow connections on HTTP (port 80). For more information, see [Adding Rules to a Security Group \(p. 553\)](#).

6. Test your WAMP server by viewing a PHP file from the web. You must be logged onto the instance as an administrator to perform the following steps.

- a. Create a file named `phpinfo.php` containing the code below and place this file in the Apache root directory. By default, the path is: `C:\Bitnami\wampstack-<version_number>\apache2\htdocs`.

```
<?php phpinfo(); ?>
```

- b. In a web browser, enter the URL of the file 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`.
 - c. Verify that the PHP information page is displayed. If the page does not display, verify that you entered the correct public DNS address. Also verify that Windows folder options are configured to show known file extensions. By default, folder options hide known file extensions. If you created the file in Notepad and saved it in the root directory your `phpinfo.php` file might incorrectly be saved as `phpinfo.php.txt`.
 - d. As a security best practice, delete the `phpinfo.php` file when you finish testing the WAMP server.
7. Enhance MySQL security by disabling default features and by setting a root password. The `mysql_secure_installation` Perl script can perform these tasks for you. To run the script, you must install Perl.
 - a. Download and install Perl from the [Perl Programming Language](#) website.
 - b. In the `C:\Bitnami\wampstack-<version_number>\mysql\bin` directory, double-click `mysql_secure_installation`.
 - c. When prompted, enter the MySQL root account password that you entered when you ran the Bitnami WAMP stack installer, and then press Enter.
 - d. Type `n` to skip changing the password.
 - e. Type `y` to remove the anonymous user accounts.
 - f. Type `y` to disable remote root login.
 - g. Type `y` to remove the test database.
 - h. Type `y` to reload the privilege tables and save your changes.

If you successfully completed the steps in this tutorial, then your WAMP server is functioning properly. To continue testing, you can add more content to the `C:\Bitnami\wampstack-<version_number>\apache2\htdocs` folder and view the content by using the public DNS address for your instance.

Important

As a best practice, stop the MySQL server if you do not plan to use it right away. You can restart the server when you need it again.

Tutorial: Installing a WIMP Server on an Amazon EC2 Instance Running Windows Server

This tutorial shows you how to install a Microsoft Internet Information Services (IIS) web server with PHP and MySQL on an EC2 instance running Windows Server. This software configuration is sometimes called a WIMP server or WIMP stack (Windows, IIS, MySQL, PHP).

A WIMP stack is designed for easy installation to help developers get up and running quickly. It is *not* designed for production environments for the following reasons:

- The default configurations do not meet security requirements for most production environments.
- Upgrading and patching the different software components on a single production server would affect server availability.
- The WAMP one-click installers do not place files in standard locations, which can make it difficult to locate important configuration files.

You can, however, create a WIMP stack on an EC2 instance to prototype a web project in a controlled test environment. For example, you can host a static website or deploy a dynamic PHP application that reads and writes information to a database.

Prerequisites

- Provision a Windows Server 2008 R2 or 2012 R2 base instance. You must configure the base instance with a public domain name system (DNS) name that is reachable from the Internet. For more information, see [Getting Started with Amazon EC2 Windows Instances \(p. 20\)](#).
- Verify that the security group for your instance has the following ports open:
 - Port 80 (HTTP inbound and outbound) - Allows computers outside of the instance to connect by using HTTP.
 - Port 443 (HTTPS inbound and outbound) - Allows computers outside of the instance to connect by using HTTPS.
 - Port 3389 (RDP inbound only) - Allows you to connect to the instance using Remote Desktop Protocol (RDP). As a security best practice, restrict RDP access to a range of IP addresses in your organization.
- Read the best practices for installing PHP on the [Microsoft web platform](#).

Prepare Your Instance

To prepare your instance

1. Connect to your instance using Microsoft Remote Desktop. For more information, see [Connecting to Your Windows Instance \(p. 359\)](#).
2. Disable Internet Explorer Enhanced Security Configuration so that you can download and install required software from the web.

Note

Make a note to re-enable Internet Explorer Enhanced Security Configuration when you have finished installing software from the web.

3. Install software updates to ensure that the instance has the latest security updates and bug fixes.

- a. **EC2Config** - [Download](#) and install the latest version of the EC2Config service. For more information about how to install this service, see [Installing the Latest Version of EC2Config \(p. 384\)](#).
- b. **Windows Update** - Run Windows Update to ensure that the latest security and software updates are installed on the instance. In Control Panel, click **System and Security**. In the **Windows Update** section, click **Check for updates**.

Install the IIS web server

IIS is a feature of Windows Server and is installed by using Server Manager. The procedure you'll use depends on the version of Windows Server your instance is running.

Install IIS on Windows Server 2012

1. In **Server Manager** click **Add roles and features**.
2. On the **Before you begin** page, click **Next**.
3. On the **Select installation type** page, select **Role-based or feature-based installation**, and then click **Next**.
4. On the **Select destination server** page, select your instance from the server pool, and then click **Next**.
5. On the **Select server roles** page, select **Web Server (IIS)**, click **Add features**, and then click **Next**.
6. On the **Select features** page, retain the default features and expand **.NET Framework 4.5 Features**, select **ASP.NET 4.5**, and then click **Next**.
7. On the **Web Server Role (IIS)** page, click **Next**.
8. On the **Select role services** page, retain the default services and select **Application Development**.
9. Expand **Application Development**, and then select the following features. When selecting these features, if you are prompted, click **Add features**:
 - a. .NET Extensibility 3.5
 - b. .NET Extensibility 4.5
 - c. Application Initialization
 - d. ASP.NET 3.5
 - e. ASP.NET 4.5
 - f. CGI
10. Click **Next**.
11. On the **Confirm installation selections** page, select **Restart the destination server automatically if required**. When prompted for confirmation, click **Yes**.
12. Click **Install**, and then after the installation is complete, click **Close**.
13. Run Windows update again.

Install IIS on Windows Server 2008

1. In **Server Manager**, click **Roles**.
2. Click **Add Roles**.
3. On the **Before You Begin** page, click **Next**.
4. On the **Select Server Roles** page, click **Web Server (IIS)**.
5. On the **Select Role Services** page under **Application Development**, click **ASP.NET**.

- a. When prompted, click **Add Required Role Services**.
- b. Click **CGI**.
- c. Click **Next**.
6. On the **Confirm Installation Selections**, click **Install**.
7. Run Windows update again.

To verify that the web server is running

After setup completes, verify that the IIS web server is configured properly and running by going to the IIS welcome page. Open a web browser on a different computer and enter either the public DNS address of the WIMP server or the public IP address. The public DNS address for your instance is listed on the Amazon EC2 console in the **Public DNS** column. If this column is hidden, click the **Show/Hide** icon and select **Public DNS**.

Important

If you do not see the Bitnami test page, use Windows Firewall with Advanced Security to create a custom rule that allows the HTTP protocol through port 80 and the HTTPS protocol through port 443. For more information, see [Network Security](#) on Microsoft TechNet. Also verify that the security group for your instances contains a rule to allow connections on HTTP (port 80). For more information, see [Adding Rules to a Security Group \(p. 553\)](#).

Install MySQL and PHP

You can download and install MySQL and PHP using the Microsoft Web Platform Installer.

To install MySQL and PHP

1. In your Windows Server instance, download and install the latest version of the [Microsoft Web Platform Installer 5.0](#).
2. In the Microsoft Web Platform Installer, click the **Products** tab.
3. Select **MySQL Windows 5.5** and click **Add**.
4. Select **PHP 5.6.0** and click **Add**.
5. Click **Install**.
6. On the **Prerequisites** page, enter a password for the MySQL default database administrator account, and then click **Continue**.
7. When the installation is complete, click **Finish**, and then click **Exit** to close the Web Platform Installer.

Test Your Server

Test your server by viewing a PHP file from the web. You must be logged onto the instance as an administrator to perform the following steps.

To test your WIMP server

1. Download and install the [Visual C++ Redistributable for Visual Studio 2012 Update 4 x86 package](#). Even if your server is a 64-bit server, you must install the x86 package.
2. Create a file named `phpinfo.php` that contains the following code and place this file in the IIS root directory. By default, the path is: `C:\inetpub\wwwroot`.

```
<?php phpinfo(); ?>
```

3. In a web browser, enter the URL of the file you just created. This URL is the public DNS address of your instance followed by a forward slash and the file name, as in the following example: `http://my.public.dns.amazonaws.com/phpinfo.php`.
4. Verify that the PHP information page is displayed. If the page does not display, verify that you entered the correct public DNS address. Also verify that Windows folder options are configured to show known file extensions. By default, folder options hide known file extensions. If you created the file in Notepad and saved it in the root directory your `phpinfo.php` file might incorrectly be saved as `phpinfo.php.txt`.
5. As a security best practice, delete the `phpinfo.php` file when you finish testing the WAMP server.
6. Enhance MySQL security by disabling default features and by setting a root password. The `mysql_secure_installation` Perl script can perform these tasks for you. To run the script, you must install Perl.
 - a. Download and install Perl from the [Perl Programming Language](#) website.
 - b. In the `C:\Program Files\MySQL\MySQL Server 5.5\bin` directory, double-click `mysql_secure_installation`.
 - c. When prompted, enter the current root password and press Enter.
 - d. Type `n` to skip changing the password.
 - e. Type `y` to remove the anonymous user accounts.
 - f. Type `y` to disable remote root login.
 - g. Type `y` to remove the test database.
 - h. Type `y` to reload the privilege tables and save your changes.

You should now have a fully functional WIMP web server. If you add content to the IIS document root at `C:\inetpub\wwwroot`, you can view that content at the public DNS address for your instance.

Important

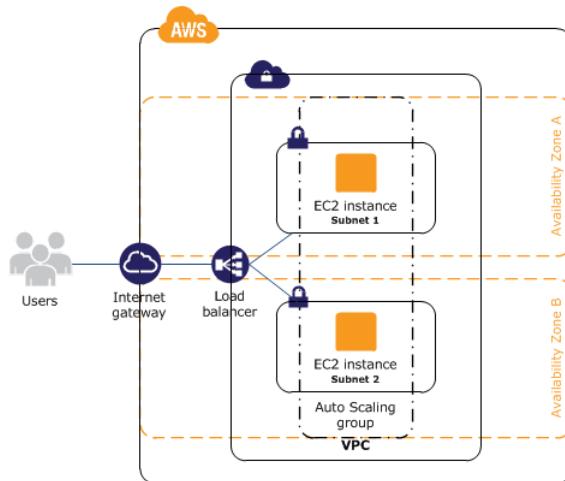
As a best practice, stop the MySQL server if you do not plan to use it right away. You can restart the server when you need it again.

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 \(p. 37\)](#)
- [Scale and Load Balance Your Application \(p. 37\)](#)
- [Test Your Load Balancer \(p. 39\)](#)

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 \(VPC\) \(p. 16\)](#).
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: Installing a WAMP Server on an Amazon EC2 Instance Running Windows Server \(p. 30\)](#) or [Tutorial: Installing a WIMP Server on an Amazon EC2 Instance Running Windows Server \(p. 33\)](#).
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 a Custom Windows AMI \(p. 65\)](#).
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 \(p. 635\)](#).

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 \(p. 37\)](#).
 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 \(p. 37\)](#).
 - 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 RDP 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 \(p. 638\)](#).

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*.

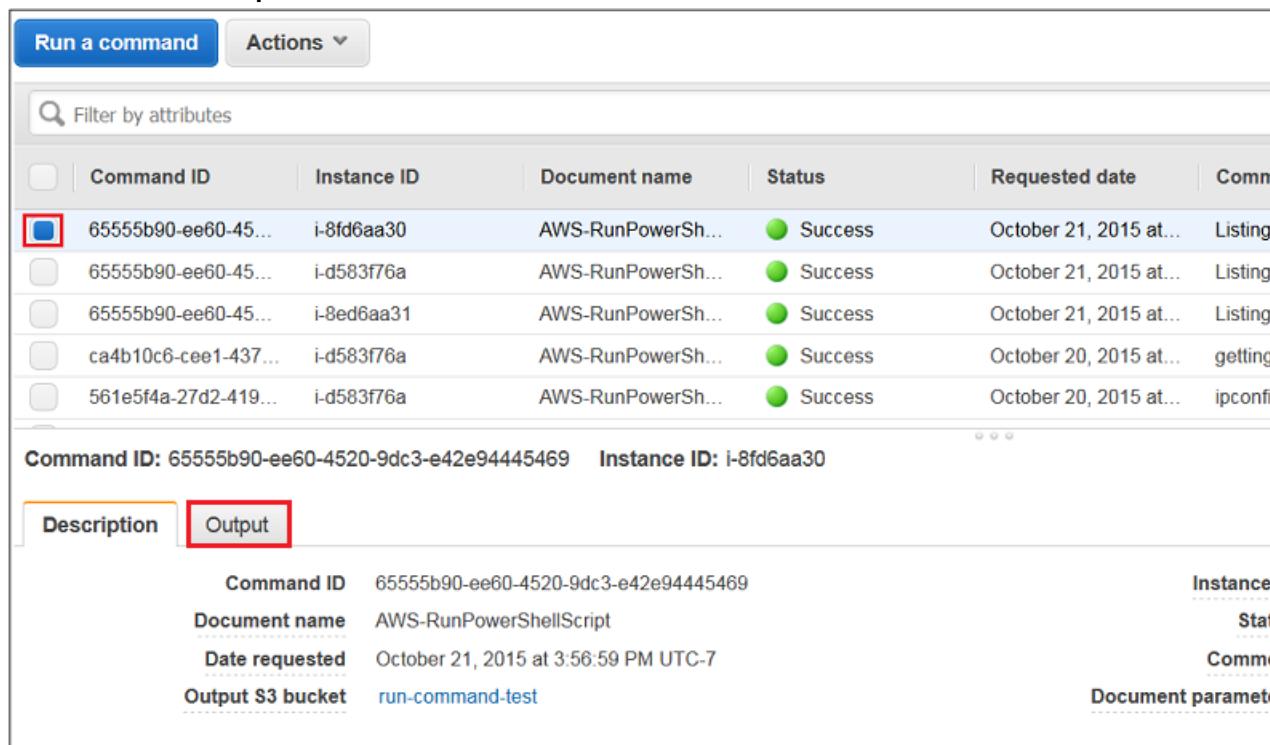
Send a Command Using the EC2 Console

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 tabs for "Run a command" and "Actions". Below that is a search bar labeled "Filter by attributes". A table lists several command executions:

	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"/>	561ef5f4a-27d2-419...	i-d583f76a	AWS-RunPowerSh...	Success	October 20, 2015 at...	ipconfi

Below the table, the command ID and instance ID are displayed: **Command ID: 65555b90-ee60-4520-9dc3-e42e94445469** **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 param...

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](#)

Tutorial: Setting Up a Windows HPC Cluster on Amazon EC2

You can launch a scalable Windows High Performance Computing (HPC) cluster using Amazon EC2 instances. A Windows HPC cluster requires an Active Directory domain controller, a DNS server, a head node, and one or more compute nodes.

To set up a Windows HPC cluster on Amazon EC2, complete the following tasks:

- [Step 2: Set Up Your Active Directory Domain Controller \(p. 46\)](#)
- [Step 3: Configure Your Head Node \(p. 47\)](#)
- [Step 4: Set Up the Compute Node \(p. 48\)](#)
- [Step 5: Scale Your HPC Compute Nodes \(Optional\) \(p. 49\)](#)

For more information about high performance computing, see [High Performance Computing \(HPC\) on AWS](#).

Prerequisites

You must launch your instances in a VPC. You can use the default VPC or create a nondefault VPC. For more information, see [Getting Started](#) in the *Amazon VPC User Guide*.

Step 1: Create Security Groups

Use the Tools for Windows PowerShell to create security groups for the domain controller, domain members, and the HPC cluster.

To create the security groups

1. Use the [New-EC2SecurityGroup](#) cmdlet to create the security group for the domain controller. Note the ID of the security group in the output.

```
PS C:\> New-EC2SecurityGroup -VpcId vpc-id -GroupName "SG - Domain Controller" -  
Description "Active Directory Domain Controller"
```

2. Use the [New-EC2SecurityGroup](#) cmdlet to create the security group for the domain members. Note the ID of the security group in the output.

```
PS C:\> New-EC2SecurityGroup -VpcId vpc-id -GroupName "SG - Domain Member" -Description  
"Active Directory Domain Member"
```

3. Use the [New-EC2SecurityGroup](#) cmdlet to create the security group for the HPC cluster. Note the ID of the security group in the output.

```
PS C:\> New-EC2SecurityGroup -VpcId vpc-id -GroupName "SG - Windows HPC Cluster" -  
Description "Windows HPC Cluster Nodes"
```

To add rules to the security groups

1. Create the following rules to add to the domain controller security group. Replace the placeholder security group ID with the ID of the domain member security group and the placeholder CIDR block with the CIDR block of your network.

```
PS C:\> $sg_dm = New-Object Amazon.EC2.Model.UserIdGroupPair  
PS C:\> $sg_dm.GroupId = "sg-12345678  
PS C:\> $r1 = @{ IpProtocol="UDP"; FromPort="123"; ToPort="123"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r2 = @{ IpProtocol="TCP"; FromPort="135"; ToPort="135"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r3 = @{ IpProtocol="UDP"; FromPort="138"; ToPort="138"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r4 = @{ IpProtocol="TCP"; FromPort="49152"; ToPort="65535"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r5 = @{ IpProtocol="TCP"; FromPort="389"; ToPort="389"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r6 = @{ IpProtocol="UDP"; FromPort="389"; ToPort="389"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r7 = @{ IpProtocol="TCP"; FromPort="636"; ToPort="636"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r8 = @{ IpProtocol="TCP"; FromPort="3268"; ToPort="3269"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r9 = @{ IpProtocol="TCP"; FromPort="53"; ToPort="53"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r10 = @{ IpProtocol="UDP"; FromPort="53"; ToPort="53"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r11 = @{ IpProtocol="TCP"; FromPort="88"; ToPort="88"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r12 = @{ IpProtocol="UDP"; FromPort="88"; ToPort="88"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r13 = @{ IpProtocol="TCP"; FromPort="445"; ToPort="445"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r14 = @{ IpProtocol="UDP"; FromPort="445"; ToPort="445"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r15 = @{ IpProtocol="ICMP"; FromPort="-1"; ToPort="-1"; UserIdGroupPairs=  
$sg_dm }  
PS C:\> $r16 = @{ IpProtocol="UDP"; FromPort="53"; ToPort="53";  
IpRanges="203.0.113.25/32" }  
PS C:\> $r17 = @{ IpProtocol="TCP"; FromPort="3389"; ToPort="3389";  
IpRanges="203.0.113.25/32" }
```

2. Use the [Grant-EC2SecurityGroupIngress](#) cmdlet to add the rules to the domain controller security group.

```
PS C:\> Grant-EC2SecurityGroupIngress -GroupId sg-1a2b3c4d -IpPermission @(`$r1, `$r2, `$r3, `$r4, `$r5, `$r6, `$r7, `$r8, `$r9, `$r10, `$r11, `$r12, `$r13, `$r14, `$r15, `$r16, `$r17)
```

For more information about these security group rules, see the following Microsoft article: [How to configure a firewall for domains and trusts](#).

3. Create the following rules to add to the domain member security group. Replace the placeholder security group ID with the ID of the domain controller security group.

```
PS C:\> $sg_dc = New-Object Amazon.EC2.Model.UserIdGroupPair
PS C:\> $sg_dc.GroupId = "sg-1a2b3c4d
PS C:\> $r1 = @{ IpProtocol="TCP"; FromPort="49152"; ToPort="65535"; UserIdGroupPairs=$sg_dc }
PS C:\> $r2 = @{ IpProtocol="UDP"; FromPort="49152"; ToPort="65535"; UserIdGroupPairs=$sg_dc }
PS C:\> $r3 = @{ IpProtocol="TCP"; FromPort="53"; ToPort="53"; UserIdGroupPairs=$sg_dc }
PS C:\> $r4 = @{ IpProtocol="UDP"; FromPort="53"; ToPort="53"; UserIdGroupPairs=$sg_dc }
```

4. Use the [Grant-EC2SecurityGroupIngress](#) cmdlet to add the rules to the domain member security group.

```
PS C:\> Grant-EC2SecurityGroupIngress -GroupId sg-12345678 -IpPermission @(`$r1, `$r2, `$r3, `$r4)
```

5. Create the following rules to add to the HPC cluster security group. Replace the placeholder security group ID with the ID of the HPC cluster security group and the placeholder CIDR block with the CIDR block of your network.

```
$sg_hpc = New-Object Amazon.EC2.Model.UserIdGroupPair
PS C:\> $sg_hpc.GroupId = "sg-87654321
PS C:\> $r1 = @{ IpProtocol="TCP"; FromPort="80"; ToPort="80"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r2 = @{ IpProtocol="TCP"; FromPort="443"; ToPort="443"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r3 = @{ IpProtocol="TCP"; FromPort="1856"; ToPort="1856"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r4 = @{ IpProtocol="TCP"; FromPort="5800"; ToPort="5800"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r5 = @{ IpProtocol="TCP"; FromPort="5801"; ToPort="5801"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r6 = @{ IpProtocol="TCP"; FromPort="5969"; ToPort="5969"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r7 = @{ IpProtocol="TCP"; FromPort="5970"; ToPort="5970"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r8 = @{ IpProtocol="TCP"; FromPort="5974"; ToPort="5974"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r9 = @{ IpProtocol="TCP"; FromPort="5999"; ToPort="5999"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r10 = @{ IpProtocol="TCP"; FromPort="6729"; ToPort="6730"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r11 = @{ IpProtocol="TCP"; FromPort="7997"; ToPort="7997"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r12 = @{ IpProtocol="TCP"; FromPort="8677"; ToPort="8677"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r13 = @{ IpProtocol="TCP"; FromPort="9087"; ToPort="9087"; UserIdGroupPairs=$sg_hpc }
```

```
PS C:\> $r14 = @{ IpProtocol="TCP"; FromPort="9090"; ToPort="9092"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r15 = @{ IpProtocol="TCP"; FromPort="9100"; ToPort="9163"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r16 = @{ IpProtocol="TCP"; FromPort="9200"; ToPort="9263"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r17 = @{ IpProtocol="TCP"; FromPort="9794"; ToPort="9794"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r18 = @{ IpProtocol="TCP"; FromPort="9892"; ToPort="9893"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r19 = @{ IpProtocol="UDP"; FromPort="9893"; ToPort="9893"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r20 = @{ IpProtocol="TCP"; FromPort="6498"; ToPort="6498"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r21 = @{ IpProtocol="TCP"; FromPort="7998"; ToPort="7998"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r22 = @{ IpProtocol="TCP"; FromPort="8050"; ToPort="8050"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r23 = @{ IpProtocol="TCP"; FromPort="5051"; ToPort="5051"; UserIdGroupPairs=$sg_hpc }
PS C:\> $r24 = @{ IpProtocol="TCP"; FromPort="3389"; ToPort="3389"; IpRanges="203.0.113.25/32" }
```

6. Use the [Grant-EC2SecurityGroupIngress](#) cmdlet to add the rules to the HPC cluster security group.

```
PS C:\> Grant-EC2SecurityGroupIngress -GroupId sg-87654321 -IpPermission @(`$r1, `$r2, `$r3, `$r4, `$r5, `$r6, `$r7, `$r8, `$r9, `$r10, `$r11, `$r12, `$r13, `$r14, `$r15, `$r16, `$r17, `$r18, `$r19, `$r20, `$r21, `$r22, `$r23, `$r24)
```

For more information about these security group rules, see the following Microsoft article: [HPC Cluster Networking: Windows Firewall configuration](#).

7. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
8. In the navigation pane, choose **Security Groups**. Verify that the all three security groups appear in the list and have the required rules.

Step 2: Set Up Your Active Directory Domain Controller

The Active Directory domain controller provides authentication and centralized resource management of the HPC environment and is required for the installation. To set up your Active Directory, launch an instance to serve as the domain controller for your HPC cluster and configure it.

To launch a domain controller for your HPC cluster

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the console dashboard, choose **Launch Instance**.
3. On the **Choose an AMI** page, select an AMI for Windows Server, and choose **Select**.
4. On the next page of the wizard, select an instance type, then choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, select your VPC from **Network** and a subnet from **Subnet**. On the next page of the wizard, you can specify additional storage for your instance.
6. On the **Add Tags** page, enter **Domain Controller** as the value for the Name tag for the instance, and then choose **Next: Configure Security Group**.
7. On the **Configure Security Group** page, choose **Select an existing security group**, choose the SG – **Domain Controller** security group, and then choose **Review and Launch**.

8. Choose **Launch**.
9. In the navigation pane, choose **Elastic IPs**.
10. Choose **Allocate new address**. Choose **Allocate**. Choose **Close**.
11. Select the Elastic IP address you created, and choose **Actions**, **Associate address**. For **Instance**, choose the domain controller instance. Choose **Associate**.

Connect to the instance you created, and configure the server as a domain controller for the HPC cluster.

To configure your instance as a domain controller

1. Connect to your Domain Controller instance. For more information, see [Connecting to Your Windows Instance \(p. 359\)](#).
2. Open **Server Manager**, and add the **Active Directory Domain Services** role.
3. Promote the server to a domain controller using Server Manager or by running **DCPromo.exe**.
4. Create a new domain in a new forest.
5. Type **hpc.local** as the fully qualified domain name (FQDN).
6. Select **Forest Functional Level as Windows Server 2008 R2**.
7. Ensure that the **DNS Server** option is selected, and then choose **Next**.
8. Select **Yes, the computer will use an IP address automatically assigned by a DHCP server (not recommended)**.
9. When prompted, choose **Yes** to continue.
10. Complete the wizard and then select **Reboot on Completion**.
11. Connect to the instance as **hpc.local\administrator**.
12. Create a domain user **hpc.local\hpcuser**.

Step 3: Configure Your Head Node

An HPC client connects to the head node. The head node facilitates the scheduled jobs. You configure your head node by launching an instance, installing the HPC Pack, and configuring the cluster.

Launch an instance and then configure it as a member of the **hpc.local** domain and with the necessary user accounts.

To configure an instance as your head node

1. Launch an instance and name it **HPC-Head**. When you launch the instance, select both of these security groups: SG - Windows HPC Cluster and SG - Domain Member.
2. Connect to the instance and get the existing DNS server address using the following command:

```
IPConfig /all
```

3. Update the TCP/IPv4 properties of the **HPC-Head** NIC to include the Elastic IP address for the **Domain Controller** instance as the primary DNS, and then add the additional DNS IP address from the previous step.
4. Join the machine to the **hpc.local** domain using the credentials for **hpc.local\administrator** (the domain administrator account).
5. Add **hpc.local\hpcuser** as the local administrator. When prompted for credentials, use **hpc.local\administrator**, and then restart the instance.
6. Connect to **HPC-Head** as **hpc.local\hpcuser**.

To install the HPC Pack

1. Connect to your **HPC-Head** instance using the `hpc.local\hpcuser` account.
2. Using **Server Manager**, turn off Internet Explorer Enhanced Security Configuration (IE ESC) for Administrators.
 - a. In **Server Manager**, under **Security Information**, choose **Configure IE ESC**.
 - b. Turn off IE ESC for administrators.
3. Install the HPC Pack on **HPC-Head**.
 - a. Download the HPC Pack to **HPC-Head** from the [Microsoft Download Center](#). Choose the HPC Pack for the version of Windows Server on **HPC-Head**.
 - b. Extract the files to a folder, open the folder, and double-click `setup.exe`.
 - c. On the Installation page, select **Create a new HPC cluster by creating a head node**, and then choose **Next**.
 - d. Accept the default settings to install all the databases on the Head Node, and then choose **Next**.
 - e. Complete the wizard.

To configure your HPC cluster on the head node

1. Start **HPC Cluster Manager**.
2. In the **Deployment To-Do List**, select **Configure your network**.
 - a. In the wizard, select the default option (5), and then choose **Next**.
 - b. Complete the wizard accepting default values on all screens, and choose how you want to update the server and participate in customer feedback.
 - c. Choose **Configure**.
3. Select **Provide Network Credentials**, then provide the `hpc.local\hpcuser` credentials.
4. Select **Configure the naming of new nodes**, and then choose **OK**.
5. Select **Create a node template**.
 - a. Select the **Compute node template**, and then choose **Next**.
 - b. Select **Without operating system**, and then continue with the defaults.
 - c. Choose **Create**.

Step 4: Set Up the Compute Node

You set up the compute node by launching an instance, installing the HPC Pack, and adding the node to your cluster.

First, launch an instance, and then configure it as a member of the `hpc.local` domain with the necessary user accounts.

To configure an instance for your compute node

1. Launch an instance and name it **HPC-Compute**. When you launch the instance, select the following security groups: **SG - Windows HPC Cluster** and **SG - Domain Member**.
2. Log in to the instance and get the existing DNS server address from **HPC-Compute** using the following command:

```
IPConfig /all
```

3. Update the TCP/IPv4 properties of the HPC-Compute NIC to include the Elastic IP address of the Domain Controller instance as the primary DNS. Then add the additional DNS IP address from the previous step.
4. Join the machine to the hpc.local domain using the credentials for hpc.local\administrator (the domain administrator account).
5. Add hpc.local\hpcuser as the local administrator. When prompted for credentials, use hpc.local\administrator, and then restart.
6. Connect to HPC-Compute as hpc.local\hpcuser.

To install the HPC Pack on the compute node

1. Connect to your HPC-Compute instance using the hpc.local\hpcuser account.
2. Using **Server Manager**, turn off Internet Explorer Enhanced Security Configuration (IE ESC) for Administrators.
 - a. In **Server Manager**, under **Security Information**, choose **Configure IE ESC**.
 - b. Turn off IE ESC for administrators.
3. Install the HPC Pack on HPC-Compute.
 - a. Download the HPC Pack to HPC-Compute from the [Microsoft Download Center](#). Choose the HPC Pack for the version of Windows Server on HPC-Compute.
 - b. Extract the files to a folder, open the folder, and double-click **setup.exe**.
 - c. On the **Installation** page, select **Join an existing HPC cluster by creating a new compute node**, and then choose **Next**.
 - d. Specify the fully-qualified name of the HPC-Head instance, and then choose the defaults.
 - e. Complete the wizard.

To complete your cluster configuration, from the head node, add the compute node to your cluster.

To add the compute node to your cluster

1. Connect to the HPC-Head instance as hpc.local\hpcuser.
2. Open **HPC Cluster Manager**.
3. Select **Node Management**.
4. If the compute node displays in the **Unapproved** bucket, right-click the node that is listed and select **Add Node**.
 - a. Select **Add compute nodes or broker nodes that have already been configured**.
 - b. Select the check box next to the node and choose **Add**.
5. Right-click the node and choose **Bring Online**.

Step 5: Scale Your HPC Compute Nodes (Optional)

To scale your compute nodes

1. Connect to the HPC-Compute instance as hpc.local\hpcuser.
2. Delete any files you downloaded locally from the HP Pack installation package. (You have already run setup and created these files on your image so they do not need to be cloned for an AMI.)
3. From C:\Program Files\Amazon\Ec2ConfigService open the file sysprep2008.xml.
4. At the bottom of <settings pass="specialize">, add the following section. Make sure to replace *hpc.local*, *password*, and *hpcuser* to match your environment.

```
<component name="Microsoft-Windows-UnattendedJoin" processorArchitecture="amd64"  
publicToken="31bf3856ad364e35"  
language="neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/  
WMICore/2002/State"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
    <Identification>  
        <UnsecureJoin>false</UnsecureJoin>  
        <Credentials>  
            <Domain>hpc.local</Domain>  
            <Password>password</Password>  
            <Username>hpcuser</Username>  
        </Credentials>  
        <JoinDomain>hpc.local</JoinDomain>  
    </Identification>  
</component>
```

5. Save sysprep2008.xml.
6. Choose **Start, All Programs, EC2ConfigService Settings**.
 - a. Choose the **General** tab, and clear the **Set Computer Name** check box.
 - b. Choose the **Bundle** tab, and then choose **Run Sysprep and Shutdown Now**.
7. Open the Amazon EC2 console.
8. In the navigation pane, choose **Instances**.
9. Wait for the instance status to show **stopped**.
10. Select the instance, choose **Actions, Image, Create Image**.
11. Specify an image name and image description, and then choose **Create Image** to create an AMI from the instance.
12. Start the original HPC-Compute instance that was shut down.
13. Connect to the head node using the hpc.local\hpcuser account.
14. From **HPC Cluster Manager**, delete the old node that now appears in an error state.
15. In the Amazon EC2 console, in the navigation pane, choose **AMIs**.
16. Use the AMI you created to add additional nodes to the cluster.

You can launch additional compute nodes from the AMI that you created. These nodes are automatically joined to the domain, but you must add them to the cluster as already configured nodes in **HPC Cluster Manager** using the head node and then bring them online.

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

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.

For more information, see [Creating a Custom Windows AMI \(p. 65\)](#).

To help categorize and manage your AMIs, you can assign custom *tags* to them. For more information, see [Tagging Your Amazon EC2 Resources \(p. 890\)](#).

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 \(p. 55\)](#).

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 \(p. 61\)](#).

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 Windows AMI \(p. 76\)](#).

AWS Windows AMIs

AWS provides a set of publicly available AMIs that contain software configurations specific to the Windows platform. Using these AMIs, you can quickly start building and deploying your applications

using Amazon EC2. First choose the AMI that meets your specific requirements, and then launch an instance using that AMI. You retrieve the password for the administrator account and then log in to the instance using Remote Desktop Connection, just as you would with any other Windows server.

Some Windows AMIs include an edition of Microsoft SQL Server (SQL Enterprise Edition, SQL Server Standard, SQL Server Express, or SQL Server Web). Launching an instance from a Windows AMI with Microsoft SQL Server enables you to run the instance as a database server. Alternatively, you can launch an instance from any Windows AMI and then install the database software that you need on the instance.

Microsoft no longer supports Windows Server 2003 (see [Microsoft Windows Server 2003 End-of-Support](#)). We recommend that you launch new EC2 instances using a supported version of Windows Server. If you have existing EC2 instances that are running an unsupported version of Windows Server, we recommend that you upgrade those instances to a supported version of Windows Server. For more information, see [Upgrading an Amazon EC2 Windows Instance to a Newer Version of Windows Server \(p. 463\)](#).

Selecting an Initial Windows AMI

To view the Windows AMIs provided by AWS, you can use the Amazon EC2 console or [AWS Marketplace](#). For more information, see [Finding a Windows AMI \(p. 53\)](#).

You can also create an AMI from your own Windows computer. For more information, see the following services:

- [AWS Server Migration Service](#)
- [VM Import/Export](#)

Keeping Your AMIs Up-to-Date

AWS provides updated, fully-patched Windows AMIs within five business days of Microsoft's patch Tuesday (the second Tuesday of each month). For more information, see [Details About AWS Windows AMI Versions \(p. 81\)](#).

The AWS Windows AMIs contain the latest security updates available at the time they were created. For more information, see [Updating Your Windows Instance \(p. 78\)](#).

Virtualization Types

AMIs 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 for better performance. Windows AMIs are 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.

HVM guests can take advantage of hardware extensions that provide fast access to the underlying hardware on the host system. 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.

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 Windows instances can get performance advantages in storage and network I/O by using them. With these PV on HVM drivers, HVM guests can get the same performance as paravirtual guests, or better.

Finding a Windows 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 provider (for example, Amazon Web Services)
- Additional software (for example, SQL server)

If you need to find a Linux AMI, see [Finding a Linux AMI](#) in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- [Finding a Windows AMI Using the Amazon EC2 Console \(p. 53\)](#)
- [Finding an AMI Using the AWS Tools for Windows PowerShell \(p. 54\)](#)
- [Finding an AMI Using the AWS CLI \(p. 54\)](#)

Finding a Windows AMI Using the Amazon EC2 Console

You can find Windows 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 Windows 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 the **Quick Start** tab, select from one of the commonly used AMIs in the list. If you don't see the AMI that you need, select the **AWS Marketplace** or **Community AMIs** tab to find additional AMIs.

To find a Windows 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 Windows 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. 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 \(p. 317\)](#). If you're not ready to launch the instance now, make note of the AMI ID for later.

Finding an AMI Using the AWS Tools for Windows PowerShell

You can use cmdlets for Amazon EC2 or AWS Systems Manager to list only the Windows 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 [Launch an Instance Using Windows PowerShell](#) in the *AWS Tools for Windows PowerShell User Guide*.

Amazon EC2

For information and examples, see [Find an AMI Using Windows PowerShell](#) in the *AWS Tools for Windows PowerShell User Guide*.

Systems Manager Parameter Store

For information and examples, see [Query for the Latest Windows AMI Using Systems Manager Parameter Store](#).

Finding an AMI Using the AWS CLI

You can use AWS CLI commands for Amazon EC2 or AWS Systems Manager to list only the Windows 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*.

Amazon EC2

The `describe-images` command supports filtering parameters. For example, use the `--owners` parameter to display public AMIs owned by Amazon.

```
aws ec2 describe-images --owners self amazon
```

You can add the following filter to the previous command to display only Windows AMIs:

```
--filters "Name=platform,Values=windows"
```

Important

Omitting the `--owners` flag from the `describe-images` command will return all images for which you have launch permissions, regardless of ownership.

Systems Manager Parameter Store

For information and examples, see [Query for the Latest Windows AMI Using Systems Manager Parameter Store](#).

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 Amazon EBS-Backed Windows 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 \(p. 55\)](#)
- [Making an AMI Public \(p. 57\)](#)
- [Sharing an AMI with Specific AWS Accounts \(p. 58\)](#)
- [Using Bookmarks \(p. 61\)](#)
- [Guidelines for Shared Windows AMIs \(p. 61\)](#)

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 (Tools for Windows PowerShell)

Use the `Get-EC2Image` command (Tools for Windows PowerShell) 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.

```
PS C:\> Get-EC2Image -ExecutableUser 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.

```
PS C:\> Get-EC2Image -ExecutableUser 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.

```
PS C:\> Get-EC2Image -Owner amazon
```

Example: List AMIs owned by an account

The following command lists the AMIs owned by the specified AWS account.

```
PS C:\> Get-EC2Image -Owner 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.

```
-Filter @{ Name="root-device-type"; Values="ebs" }
```

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"
```

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 \(p. 58\)](#)). 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 \(p. 71\)](#).

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 (Tools for Windows PowerShell)

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 [Edit-EC2ImageAttribute](#) command as follows to add the `all` group to the `launchPermission` list for the specified AMI.

```
PS C:\> Edit-EC2ImageAttribute -ImageId ami-12345678 -Attribute launchPermission -  
OperationType add -UserGroup all
```

2. To verify the launch permissions of the AMI, use the following [Get-EC2ImageAttribute](#) command.

```
PS C:\> Get-EC2ImageAttribute -ImageId 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.

```
PS C:\> Edit-EC2ImageAttribute -ImageId ami-12345678 -Attribute launchPermission -  
OperationType remove -UserGroup all
```

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 \(p. 71\)](#).

There is no limit to the number of AWS accounts with which an AMI can be shared.

Note

You cannot directly share an AMI that contains a snapshot of an encrypted volume. You can share your encrypted snapshots with other AWS accounts. This enables the other account to copy the snapshots to other regions, re-encrypt the snapshots, and create AMIs using the encrypted snapshots. For more information, see [Sharing an Amazon EBS Snapshot \(p. 796\)](#).

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 \(p. 55\)](#).

Sharing an AMI (Tools for Windows PowerShell)

Use the [Edit-EC2ImageAttribute](#) command (Tools for Windows PowerShell) 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.

```
PS C:\> Edit-EC2ImageAttribute -ImageId ami-12345678 -Attribute launchPermission -OperationType add -UserId "123456789012"
```

The following command grants create volume permission for a snapshot.

```
PS C:\> Edit-EC2SnapshotAttribute -SnapshotId snap-1234567890abcdef0 -Attribute CreateVolumePermission -OperationType add -UserId 123456789012
```

To remove launch permissions for an account

The following command removes launch permissions for the specified AMI from the specified AWS account:

```
PS C:\> Edit-EC2ImageAttribute -ImageId ami-12345678 -Attribute launchPermission -OperationType remove -UserId "123456789012"
```

The following command removes create volume permission for a snapshot.

```
PS C:\> Edit-EC2SnapshotAttribute -SnapshotId snap-1234567890abcdef0 -Attribute CreateVolumePermission -OperationType remove -UserId 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.

```
PS C:\> Reset-EC2ImageAttribute -ImageId ami-12345678 -Attribute launchPermission
```

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 Windows AMIs

Use the following guidelines to reduce the attack surface and improve the reliability of the AMIs you create.

- No list of security guidelines can be exhaustive. Build your shared AMIs carefully and take time to consider where you might expose sensitive data.
- Develop a repeatable process for building, updating, and republishing AMIs.
- Build AMIs using the most up-to-date operating systems, packages, and software.
- [Download](#) and install the latest version of the EC2Config service. For more information about installing this service, see [Installing the Latest Version of EC2Config \(p. 384\)](#).
- Verify that Ec2SetPassword, Ec2WindowsActivate and Ec2HandleUserData are enabled.
- Verify that no guest accounts or Remote Desktop user accounts are present.
- Disable or remove unnecessary services and programs to reduce the attack surface of your AMI.
- Remove instance credentials, such as your key pair, from the AMI (if you saved them on the AMI). Store the credentials in a safe location.
- Ensure that the administrator password and passwords on any other accounts are set to an appropriate value for sharing. These passwords are available for anyone who launches your shared AMI.
- Test your AMI before you share it.

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 \(p. 62\)](#)
- [Finding a Paid AMI \(p. 62\)](#)
- [Purchasing a Paid AMI \(p. 64\)](#)
- [Getting the Product Code for Your Instance \(p. 64\)](#)
- [Using Paid Support \(p. 64\)](#)
- [Bills for Paid and Supported AMIs \(p. 65\)](#)
- [Managing Your AWS Marketplace Subscriptions \(p. 65\)](#)

Selling Your AMI

You can sell your AMI using AWS Marketplace. AWS Marketplace offers an organized shopping experience. Additionally, AWS Marketplace also supports AWS features such as Amazon EBS-backed AMIs, Reserved Instances, and Spot Instances.

For information about how to sell your AMI on AWS Marketplace, see [Selling on AWS Marketplace](#).

Finding a Paid AMI

There are several ways that you can find AMIs that are available for you to purchase. For example, you can use [AWS Marketplace](#), the Amazon EC2 console, or the command line. Alternatively, a developer might let you know about a paid AMI themselves.

Finding a Paid AMI Using the Console

To find a paid AMI using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs**.
3. Choose **Public images** for the first filter.
4. In the Search bar, choose **Owner**, then **AWS Marketplace**.
5. If you know the product code, choose **Product Code**, then type the product code.

Finding a Paid AMI Using AWS Marketplace

To find a paid AMI using AWS Marketplace

1. Open [AWS Marketplace](#).
2. Enter the name of the operating system in the search box, and click **Go**.
3. To scope the results further, use one of the categories or filters.
4. Each product is labeled with its product type: either **AMI** or **Software as a Service**.

Finding a Paid AMI Using the Tools for Windows PowerShell

You can find a paid AMI using the following [Get-EC2Image](#) command.

```
PS C:\> Get-EC2Image -Owner aws-marketplace
```

The output for a paid AMI includes the product code.

ProductCodeId	ProductCodeType
<i>product_code</i>	marketplace

If you know the product code, you can filter the results by product code. This example returns the most recent AMI with the specified product code.

```
PS C:\> (Get-EC2Image -Owner aws-marketplace -Filter @{"Name"="product-code"; "Value"="product_code"} | sort CreationDate -Descending | Select-Object -First 1).ImageId
```

Finding a Paid AMI Using the AWS CLI

You can find a paid AMI using the following [describe-images](#) command (AWS CLI).

```
aws ec2 describe-images --owners aws-marketplace
```

This command returns numerous details that describe each AMI, including the product code for a paid AMI. The output from `describe-images` includes an entry for the product code like the following:

```
"ProductCodes": [  
    {  
        "ProductCodeId": "product_code",  
        "ProductCodeType": "marketplace"  
    }  
,
```

If you know the product code, you can filter the results by product code. This example returns the most recent AMI with the specified product code.

```
aws ec2 describe-images --owners aws-marketplace \  
--filters "Name=product-code,Values=product_code"  
--query "sort_by(Images, &CreationDate)[-1].[ImageId]"
```

Purchasing a Paid AMI

You must sign up for (purchase) a paid AMI before you can launch an instance using the AMI.

Typically a seller of a paid AMI presents you with information about the AMI, including its price and a link where you can buy it. When you click the link, you're first asked to log into AWS, and then you can purchase the AMI.

Purchasing a Paid AMI Using the Console

You can purchase a paid AMI by using the Amazon EC2 launch wizard. For more information, see [Launching an AWS Marketplace Instance \(p. 332\)](#).

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 \(p. 332\)](#).

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 \(p. 452\)](#).

To retrieve a product code, use the following command:

```
PS C:\> Invoke-RestMethod -uri 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 a Custom Windows AMI

To create a Windows AMI, you launch an instance from an existing Windows AMI, customize the instance, and create a new AMI from the instance.

To create a custom Linux AMI, use the procedure for the type of volume for the instance. For more information, see [Creating an Amazon EBS-Backed Linux AMI](#) or [Creating an Instance Store-Backed Linux AMI](#) in the *Amazon EC2 User Guide for Linux Instances*.

Overview of Creating an AMI

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, ensure data integrity by stopping the instance before you create an AMI and then create the image. We automatically register the AMI for you.

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 Windows AMI \(p. 76\)](#). If any volumes attached to the instance are encrypted, the new AMI only launches successfully on instance types that support Amazon EBS encryption. For more information, see [Amazon EBS Encryption \(p. 815\)](#).

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 prior to 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 \(p. 789\)](#).

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.

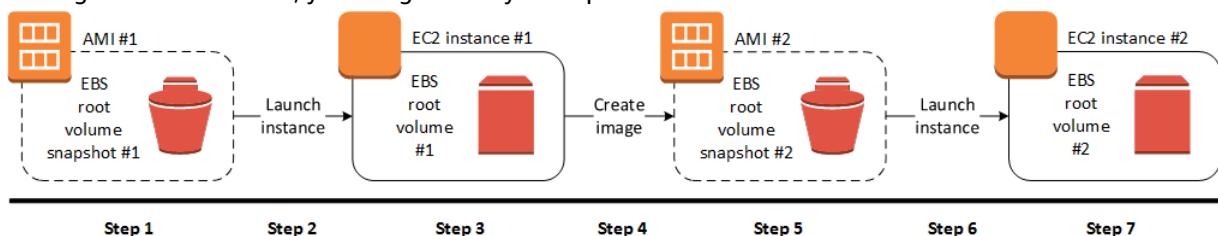
If you add instance store volumes or Amazon EBS volumes to your instance in addition to the root device volume, the block device mapping for the new AMI contains information for these volumes, and the block device mappings for instances that you launch from the new AMI automatically contain information 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 \(p. 855\)](#).

Note

When you create a new instance from a custom 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 Windows AMI from a Running Instance

You can create an AMI using the AWS Management Console or the command line. The following diagram summarizes the process for creating an 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. If you already have a running Windows instance, you can go directly to step 4.



To create an AMI 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 **Images, AMIs**.

3. Use the **Filter** options to scope the list of AMIs to the Windows AMIs that meet your needs. For example, to view the Windows AMIs provided by AWS, choose **Public images** from the drop-down list. Choose the Search bar. Choose **Owner** from the menu and choose **Amazon images**. Choose **Source** from the menu and type one of the following, depending on the version of Windows Server that you need:

- **amazon/Windows_Server-2016**
- **amazon/Windows_Server-2012**
- **amazon/Windows_Server-2008**

Add any other filters that you need. When you have chosen an AMI, select its checkbox.

4. Choose **Launch**. Accept the default values as you step through the wizard. For more information, see [Launching an Instance Using the Launch Instance Wizard \(p. 315\)](#). When the instance is ready, connect to it. For more information, see [Connecting to Your Windows Instance \(p. 359\)](#).
5. 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 EBS volumes
 - Create a new user account and add it to the Administrators group

If you are sharing your AMI, these credentials can be supplied for RDP access without disclosing your default administrator password.

- [Windows Server 2016] Configure settings using EC2Launch. To generate a random password at launch time, use the `adminPasswordType` setting. For more information, see [Configuring EC2Launch \(p. 376\)](#).
- [Windows Server 2012 R2 and earlier] Configure settings using EC2Config. To generate a random password at launch time, enable the `Ec2SetPassword` plugin; otherwise, the current administrator password is used. For more information, see [EC2Config Settings Files \(p. 390\)](#).
- [Windows Server 2008 R2] If the instance uses RedHat drivers to access Xen virtualized hardware, upgrade to Citrix drivers before you create an AMI. For more information, see [Upgrade Windows Server 2008 and 2008 R2 Instances \(Redhat to Citrix PV Upgrade\) \(p. 416\)](#).

6. In the navigation pane, choose **Instances** and select your instance. Choose **Actions**, **Image**, and **Create Image**.

Tip

If this option is disabled, your instance isn't an Amazon EBS-backed instance.

7. Specify a unique name for the image and an optional description (up to 255 characters).

By default, Amazon EC2 shuts down the instance, takes snapshots of any attached volumes, creates and registers the AMI, and then reboots the instance. Choose **No reboot** if you don't want your instance to be shut down.

Warning

If you choose **No reboot**, we can't guarantee the file system integrity of the created image.

(Optional) Modify the root volume, Amazon EBS volumes, and instance store volumes as needed. For example:

- To change the size of the root volume, locate the **Root** volume in the **Type** column, and fill in the **Size** field.
- To suppress an Amazon EBS volume specified by the block device mapping of the AMI used to launch the instance, locate the EBS volume in the list and choose **Delete**.

- To add an Amazon EBS volume, choose **Add New Volume**, **Type**, and **EBS**, and fill in the fields. When you then launch an instance from your new AMI, these additional volumes are automatically attached to the instance. Empty volumes must be formatted and mounted. Volumes based on a snapshot must be mounted.
- To suppress an instance store volume specified by the block device mapping of the AMI used to launch the instance, locate the volume in the list and choose **Delete**.
- To add an instance store volume, choose **Add New Volume**, **Type**, and **Instance Store**, and select a device name from the **Device** list. When you launch an instance from your new AMI, these additional volumes are automatically initialized and mounted. These volumes don't contain data from the instance store volumes of the running instance from which you based your AMI.

When you are finished, choose **Create Image**.

8. While your AMI is being created, you can choose **AMIs** in the navigation pane to view its status. Initially, this is **Pending**. After a few minutes, the status should change to **Available**.
(Optional) Choose **Snapshots** in the navigation pane to view the snapshot that was created for the new AMI. When you launch an instance from this AMI, we use this snapshot to create its root device volume.
9. Launch an instance from your new AMI. For more information, see [Launching an Instance Using the Launch Instance Wizard \(p. 315\)](#). The new running instance contains all of the customizations you applied in previous steps, and any additional customization you add when launching the instance, such as user data (scripts that run when the instance starts).

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)

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.

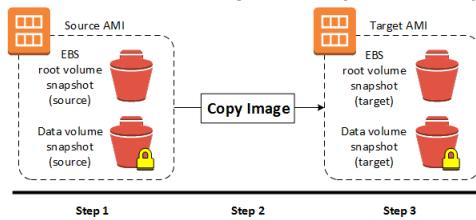
The encryption parameters of `CopyImage` are available in all regions where AWS KMS is available.

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

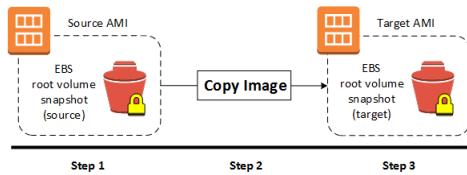
In this scenario, an EBS-backed AMI has an unencrypted root snapshot and an encrypted data snapshot, shown in step 1. The `CopyImage` action is invoked in step 2 without encryption parameters. As a result, the encryption status of each snapshot is preserved, so that the destination AMI, in step 3, is 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 \(p. 71\)](#).

Copying an AMI Backed by An Encrypted Root Snapshot

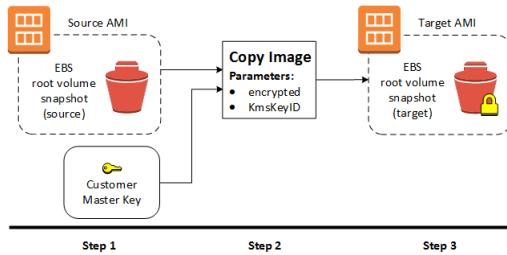
In this scenario, an Amazon EBS-backed AMI has an encrypted root snapshot, shown in step 1. The `CopyImage` action is invoked in step 2 without encryption parameters. As a result, the encryption status of the snapshot is preserved, so that the destination AMI, in step 3, is also backed by an encrypted 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 \(p. 71\)](#).

Creating an AMI with Encrypted Root Snapshot from an Unencrypted AMI

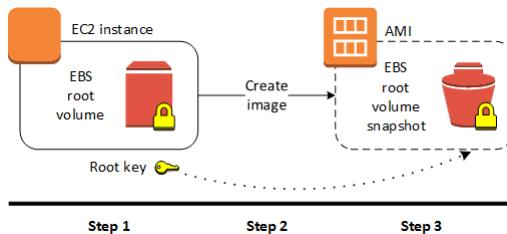
In this scenario, an Amazon EBS-backed AMI has an unencrypted root snapshot, shown in step 1, and an AMI is created with an encrypted root snapshot, shown in step 3. The `CopyImage` action in step 2 is invoked with two encryption parameters, including the choice of a CMK. As a result, the encryption 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 \(p. 71\)](#).

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.



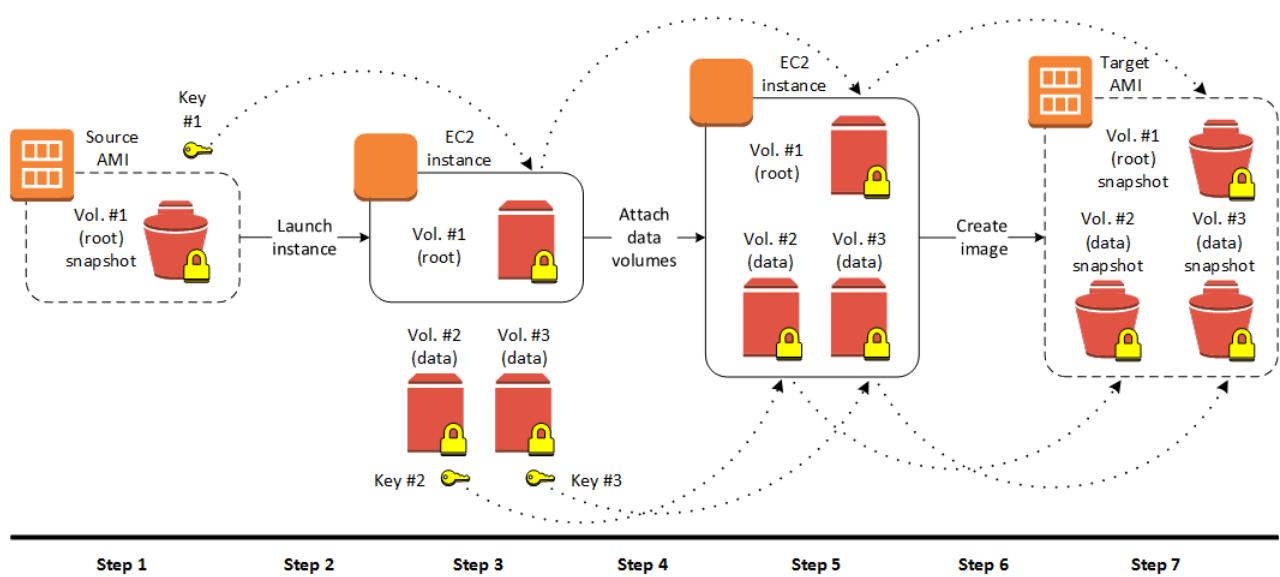
You can create an AMI from a running Amazon EC2 instance (with or without encrypted volumes) using either the Amazon EC2 console or the command line. For more information, see [Creating a Custom Windows AMI \(p. 65\)](#).

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 \(p. 314\)](#)
- [Creating a Custom Windows AMI \(p. 65\)](#).
- [Amazon EBS Volumes \(p. 734\)](#)
- [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 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*"  
            ]  
        }  
    ]  
}
```

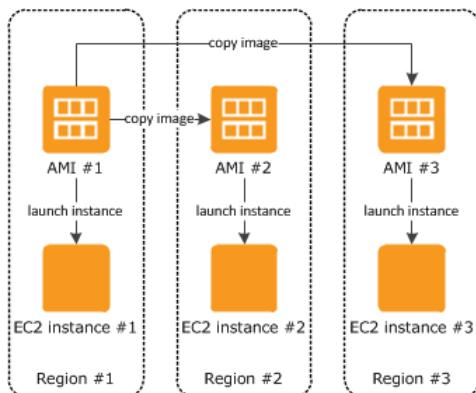
To find the Amazon Resource Name (ARN) of the AMI source bucket, open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>, in the navigation pane choose **AMIs**, and locate the bucket name in the **Source** column.

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.

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 \(p. 58\)](#).

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 a Custom Windows AMI \(p. 65\)](#).

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 \(p. 68\)](#).

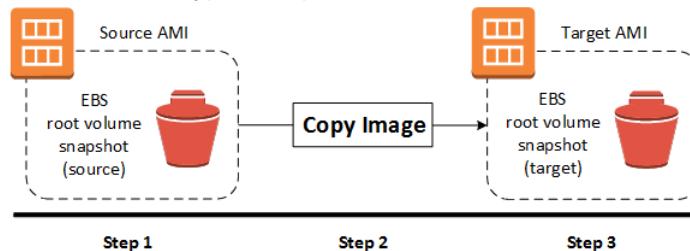
By default, the backing snapshot of an AMI is copied with its original encryption status. Copying an AMI backed by an unencrypted snapshot results in an identical target snapshot that is also unencrypted. If the source AMI is backed by an encrypted snapshot, copying it results in a target snapshot encrypted 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 \(p. 68\)](#).

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

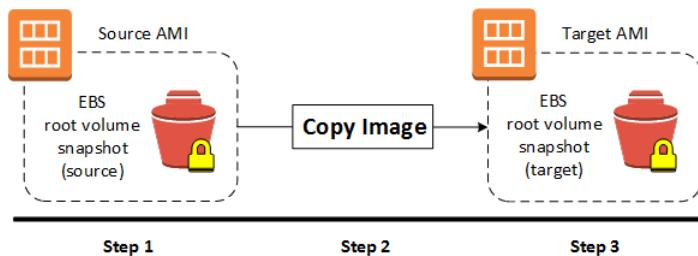
Copy an unencrypted source AMI to an unencrypted target AMI

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.



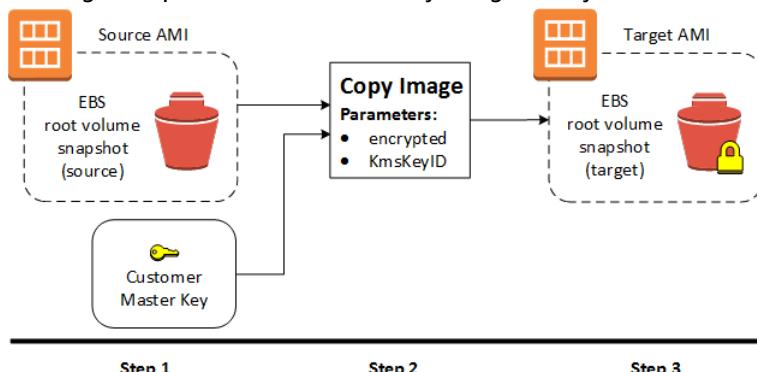
Copy an encrypted source AMI to an encrypted target AMI

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 a Custom Windows AMI \(p. 65\)](#) and [Finding a Windows AMI \(p. 53\)](#).

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

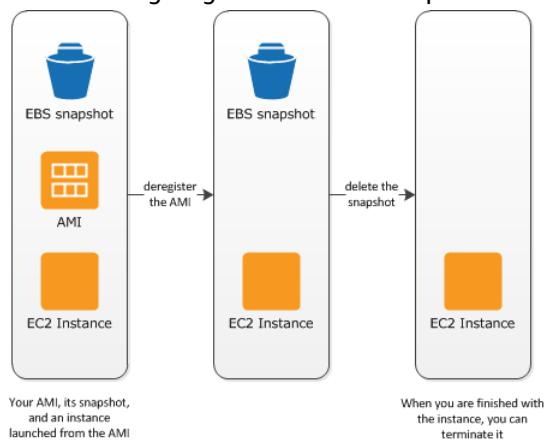
- [deregister-image](#) (AWS CLI)
- [Unregister-EC2Image](#) (AWS Tools for Windows PowerShell)

Deregistering Your Windows AMI

You can deregister a Windows 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 or any snapshots created for the EBS root volume during the AMI creation process. You'll continue to incur usage costs for these instances and storage costs for the snapshot. Therefore, you should terminate any instances that you finished with and delete any snapshots that you are finished with.

The following diagram illustrates the process for cleaning up your Windows AMI.



To clean up your Windows 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 can 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**.

Managed AWS Windows AMIs

AWS provides managed Amazon Machine Images (AMIs) that include various versions and configurations of Windows Server. In general, the AWS Windows AMIs are configured with the default settings used by the Microsoft installation media. However, there are customizations. For example, the AWS Windows AMIs come with the following software and drivers:

- EC2Config service (through Windows Server 2012 R2)
- EC2Launch (Windows Server 2016 only)
- AWS Systems Manager
- AWS CloudFormation
- AWS Tools for Windows PowerShell
- Network drivers (SRIOV, ENA, Citrix PV)
- Storage drivers (NVMe, AWS PV, Citrix PV)
- Graphics drivers (NVIDIA GPU, Elastic GPU)

- Spot Instance hibernation

For information about other customizations, see [Configuration Changes for AWS Windows AMIs \(p. 79\)](#).

Contents

- [Updating Your Windows Instance \(p. 78\)](#)
- [Upgrading or Migrating to a Newer Version of Windows Server \(p. 78\)](#)
- [Subscribing to Windows AMI Notifications \(p. 78\)](#)
- [Configuration Changes for AWS Windows AMIs \(p. 79\)](#)
- [Details About AWS Windows AMI Versions \(p. 81\)](#)
- [Changes in Windows Server 2016 AMIs \(p. 102\)](#)
- [Docker Container Conflict on Windows Server 2016 Instances \(p. 103\)](#)
- [Issue with the Hibernate Agent \(2018.03.16 AMIs\) \(p. 103\)](#)

Updating Your Windows Instance

After you launch a Windows instance, you are responsible for installing updates on it. You can manually install only the updates that interest you, or you can start from a current AWS Windows AMI and build a new Windows instance. For information about finding the current AWS Windows AMIs, see [Finding a Windows AMI \(p. 53\)](#).

For Windows instances, you can install updates to the following services or applications:

- [Microsoft Windows Server](#)
- [Microsoft SQL Server](#)
- [Windows PowerShell](#)
- [EC2Launch \(p. 375\)](#)
- [EC2Config service \(p. 384\)](#)
- [SSM Agent](#)
- [PV Drivers \(p. 414\)](#)
- [AWS Tools for Windows PowerShell](#)
- [AWS CloudFormation helper scripts](#)

You can reboot a Windows instance after installing updates. For more information, see [Reboot Your Instance \(p. 365\)](#).

Upgrading or Migrating to a Newer Version of Windows Server

For information about how to upgrade or migrate a Windows instance to a newer version of Windows Server, see [Upgrading an Amazon EC2 Windows Instance to a Newer Version of Windows Server \(p. 463\)](#).

Subscribing to Windows AMI Notifications

To be notified when new AMIs are released or when previously released AMIs are made private, subscribe for notifications using Amazon SNS.

To subscribe to Windows AMI 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 this region because the SNS notifications that you are subscribing to were created in this region.
 3. In the navigation pane, choose **Subscriptions**.
 4. Choose **Create subscription**.
 5. For the **Create subscription** dialog box, do the following:
 - a. For **Topic ARN**, copy and paste one of the following Amazon Resource Names (ARNs):
 - **arn:aws:sns:us-east-1:801119661308:ec2-windows-ami-update**
 - **arn:aws:sns:us-east-1:801119661308:ec2-windows-ami-private**
- For AWS GovCloud (US):
- arn:aws-us-gov:sns:us-gov-west-1:077303321853:ec2-windows-ami-update**
- b. For **Protocol**, choose **Email**.
 - c. For **Endpoint**, type an email address that you can use to receive the notifications.
 - d. Choose **Create subscription**.
6. You'll receive a confirmation email with the subject line **AWS Notification – Subscription Confirmation**. Open the email and choose **Confirm subscription** to complete your subscription.

Whenever Windows AMIs are released, we send notifications to the subscribers of the `ec2-windows-ami-update` topic. Whenever released Windows AMIs are made private, we send notifications to the subscribers of the `ec2-windows-ami-private` topic. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from Windows AMI 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 this region because the SNS notifications were created in this region.
3. In the navigation pane, choose **Subscriptions**.
4. Select the subscriptions and then choose **Actions, Delete subscriptions**. When prompted for confirmation, choose **Delete**.

Configuration Changes for AWS Windows AMIs

The following changes are applied to each AWS Windows AMI.

Clean and Prepare

Change	Applies to
Check for pending file renames or reboots, and reboot as needed	All AMIs
Delete .dmp files	All AMIs
Delete logs (event logs, SSM, EC2Config)	All AMIs
Delete temporary folders and files for sysprep	All AMIs

Change	Applies to
Clear recent history (Start menu, Windows Explorer, and so on)	Windows Server 2012 R2 and earlier
Perform virus scan	All AMIs
Pre-compile queued .NET assemblies (before sysprep)	All AMIs
Run Windows maintenance tools	Windows Server 2012 R2 and later
Restore default values for Internet Explorer	All AMIs
Restore default values for EC2Config	Windows Server 2012 R2 and earlier
Set EC2Launch to run at the next launch	Windows Server 2016
Reset the Windows wallpaper	All AMIs
Run sysprep	All AMIs

Install and Configure

Change	Applies to
Add links to the Amazon EC2 Windows Guide	All AMIs
Attach instance storage volumes to extended mount points	All AMIs
Install the current AWS Tools for Windows PowerShell	All AMIs
Install the current AWS CloudFormation helper scripts	All AMIs
Install the current EC2Config and SSM Agent	Windows Server 2012 R2 and earlier
Install the current EC2Launch and SSM Agent	Windows Server 2016
Install the current AWS PV, ENA, and NVMe drivers	Windows Server 2008 R2 and later
Install the current SRIOV drivers	Windows Server 2012 R2 and later
Install the current Citrix PV driver	Windows Server 2008 SP2 and earlier
Install the current EC2WinUtil driver	Windows Server 2008 R2 and later
Install PowerShell 2.0 and 3.0	Windows Server 2008 SP2 and R2
If Microsoft SQL Server is installed:	All AMIs
<ul style="list-style-type: none"> • Install service packs • Configure to start automatically • Add BUILTIN\Administrators to the SysAdmin role 	

Change	Applies to
<ul style="list-style-type: none"> Open TCP port 1433 and UDP port 1434 	
Apply the following hotfixes: <ul style="list-style-type: none"> MS15-011 KB2582281 KB2634328 KB2800213 KB2922223 KB2394911 KB2780879 	Windows Server 2008 SP2 and R2
Allow ICMP traffic through the firewall	Windows Server 2012 R2 and earlier
Enable file and printer sharing	Windows Server 2012 R2 and earlier
Disable RunOnce for Internet Explorer	All AMIs
Enable remote PowerShell	All AMIs
Configure a paging file on the system volume as follows: <ul style="list-style-type: none"> Windows Server 2016 - Managed by the system Windows Server 2012 R2 - Initial size and max size are 8 GB Windows Server 2012 and earlier - Initial size is 512 MB, max size is 8 GB 	All AMIs
Configure an additional system managed paging file on z:, if available	Windows Server 2012 R2 and earlier
Disable hibernation and delete the hibernation file	All AMIs
Set the performance options for best performance	All AMIs
Set the power setting to high performance	All AMIs
Disable the screen saver password	All AMIs
Set the RealTimeIsUniversal registry key	All AMIs
Set the timezone to UTC	All AMIs
Disable Windows updates and notifications	All AMIs
Run Windows Update and reboot until there are no pending updates	All AMIs

Details About AWS Windows AMI Versions

What to Expect in an Official AWS Windows AMI

AWS provides AMIs with a variety of configurations for all supported Windows Operating System versions from 2003 R2 to 2016. For each of these images, AWS:

- Installs all Microsoft recommended Windows security patches. We release images shortly after each monthly Microsoft patches are made available.
- Installs the latest drivers for AWS hardware, including network and disk drivers, EC2WinUtil for troubleshooting, as well as GPU drivers in selected AMIs.
- Includes AWS helper software, like [EC2 Config \(p. 383\)](#) for Server 2012 R2 and earlier, or [EC2 Launch \(p. 374\)](#) for Server 2016.
- Configures Windows Time to use the [AWS Time Service \(p. 437\)](#).
- Makes changes in all power schemes to set the display to never turn off.
- Performs minor bug fixes – generally one-line registry changes to enable or disable features that we have found to improve performance on AWS.

Other than the adjustments listed above, we keep our AMIs as close as possible to the default install. This means we default to the “stock” PowerShell or .NET framework versions, don’t install Windows Features, and generally don’t change the AMI.

How AWS Decides Which Windows AMIs to Offer

Each AMI is extensively tested prior to release to the general public. We periodically streamline our AMI offerings to simplify customer choice and to reduce costs.

- New AMI offerings are created for new OS releases. You can count on AWS releasing “Base,” “Core/Container,” and “SQL Express/Standard/Web/Enterprise” offerings in English and other widely used languages.
- New AMI offerings are created to support new platforms – for example, the Deep Learning and “NVIDIA” AMIs were created to support customers using our GPU-based instance types (P2 and P3, G2 and G3, etc.).
- Less popular AMIs are sometimes removed. If we see a particular AMI is launched only a few times in its entire lifespan, we will remove it in favor of more widely used options.

If there is an AMI variant that you would like to see, let us know by filing a ticket with Cloud Support, or by providing feedback through [one of our established channels](#).

Patches, Security Updates, and AMI IDs

AWS provides updated, fully-patched Windows AMIs within five business days of Microsoft’s patch Tuesday (the second Tuesday of each month). The new AMIs are available immediately through the [Images](#) page in the Amazon EC2 console. The new AMIs are available in the AWS Marketplace and the [Quick Start](#) tab of the launch instance wizard within a few days of their release.

To ensure that customers have the latest security updates by default, AWS keeps Windows AMIs available for three months. After releasing new Windows AMIs, AWS makes the Windows AMIs that are older than three months private within 10 days. After an AMI has been made private, if you look at an instance launched from that AMI in the console, the **AMI ID** field states, “Cannot load detail for ami-xxxxx. You may not be permitted to view it.” You can still retrieve the AMI ID using the AWS CLI or an AWS SDK.

The Windows AMIs in each release have new AMI IDs. Therefore, we recommend that you write scripts that locate the latest AWS Windows AMIs by their names, rather than by their IDs. For more information, see the following examples:

- [Get-EC2ImageByName](#) (AWS Tools for Windows PowerShell)
- [Query for the Latest Windows AMI Using Systems Manager Parameter Store](#)
- [Walkthrough: Looking Up Amazon Machine Image IDs](#) (AWS Lambda, AWS CloudFormation)

Semiannual Channel Releases

AWS provides Windows Server semiannual channel releases that combine the scale, performance, and elasticity of AWS with the new capabilities in the [Semiannual channel release versions of Windows Server](#).

The following tables summarize the changes to each release of the AWS Windows AMIs. Note that some changes apply to all AWS Windows AMIs while others apply to only a subset of these AMIs.

Contents

- [Monthly AMI Updates for 2018 \(to date\) \(p. 83\)](#)
- [Monthly AMI Updates for 2017 \(p. 87\)](#)
- [Monthly AMI Updates for 2016 \(p. 91\)](#)
- [Monthly AMI Updates for 2015 \(p. 94\)](#)
- [Monthly AMI Updates for 2014 \(p. 97\)](#)
- [Monthly AMI Updates for 2013 \(p. 98\)](#)
- [Monthly AMI Updates for 2012 \(p. 100\)](#)
- [Monthly AMI Updates for 2011 and earlier \(p. 102\)](#)

For more information about components included in these AMIs, see the following:

- [EC2Config Version History \(p. 395\)](#)
- [EC2Launch Version History \(p. 382\)](#)
- [Amazon SSM Agent Release Notes](#)
- [Amazon ENA Driver Versions \(p. 687\)](#)
- [AWS PV Driver Version History \(p. 411\)](#)

Monthly AMI Updates for 2018 (to date)

For more information about Microsoft updates, see [Description of Software Update Services and Windows Server Update Services changes in content for 2018](#).

Release	Changes
2018.11.28	All AMIs <ul style="list-style-type: none">• SSM Agent 2.3.235.0• Changes in all power schemes to set the display to never turn off
2018.11.20	Windows_Server-2016-English-Deep-Learning Windows_Server-2016-English-Deep-Learning <ul style="list-style-type: none">• TensorFlow version 1.12• MXNet version 1.3• NVIDIA version 392.05
2018.11.19	All AMIs <ul style="list-style-type: none">• Microsoft security updates current to November 19th, 2018• AWS SDK version 3.15.602.0• SSM agent 2.3.193.0

Release	Changes
	<ul style="list-style-type: none"> EC2Config version 4.9.3067 <p>Windows Server, version 1809</p> <ul style="list-style-type: none"> AMIs are now available.
2018.10.14	<p>All AMIs</p> <ul style="list-style-type: none"> Microsoft security updates current to October 9th, 2018 AWS Tools for Windows PowerShell version 3.3.365.0 CloudFormation version 1.4.31 AWS PV Driver version 8.2.4 AWS PCI Serial Driver version 1.0.0.0 (support for Windows 2008R2 and 2012 on Bare Metal instances) ENI Driver version 1.5.0 <p>Microsoft Windows Server 2016 Datacenter and Standard Editions for Nano Server</p> <p>Microsoft ended mainstream support for Windows Server 2016 Datacenter and Standard Editions for Nano Server installation options as of April 10, 2018 (see the Microsoft Support Lifecycle page for more details).</p> <p>Additional information about Nano Server lifecycle, including details on launching Nano Server as a container, can be found here: https://docs.microsoft.com/en-us/windows-server/get-started/nano-in-semi-annual-channel.</p>
2018.09.15	<p>All AMIs</p> <ul style="list-style-type: none"> Microsoft security updates current to September 12, 2018 AWS Tools for Windows PowerShell version 3.3.343 EC2Launch version 1.3.2000430 AWS NVMe Driver version 1.3.0 EC2 WinUtil Driver version 2.0.0 <p>Microsoft Windows Server 2016 Base Nano</p> <p>Access to all public versions of Windows_Server-2016-English-Nano-Base will be removed in September 2018. Additional information about Nano Server lifecycle, including details on launching Nano Server as a Container, can be found here: https://docs.microsoft.com/en-us/windows-server/get-started/nano-in-semi-annual-channel.</p>

Release	Changes
2018.08.15	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to August 14, 2018 • AWS Tools for Windows PowerShell version 3.3.335 • AMIs now default to use Amazon's NTP service at IP 169.254.169.123 for time synchronization. For more information, see Default NTP Settings for Amazon Windows AMIs (p. 437). <p>Microsoft Windows Server 2016 Base Nano</p> <p>Access to all public versions of Windows_Server-2016-English-Nano-Base will be removed in September 2018. Additional information about Nano Server lifecycle, including details on launching Nano Server as a Container, can be found here: https://docs.microsoft.com/en-us/windows-server/get-started/nano-in-semi-annual-channel.</p>
2018.07.11	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to July 10, 2018 • EC2Config version 4.9.2756 • SSM Agent 2.2.800.0
2018.06.22	<p>Windows Server 2008 R2</p> <ul style="list-style-type: none"> • Resolves an issue with the 2018.06.13 AMIs when changing an instance from a previous generation to a current generation (for example, M4 to M5).
2018.06.13	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to June 12, 2018 • EC2Config version 4.9.2688 • SSM Agent 2.2.619.0 • AWS Tools for Windows PowerShell 3.3.283.0 • AWS NVMe driver 1.2.0 • AWS PV driver 8.2.3
2018.05.09	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to May 9, 2018 • EC2Config version 4.9.2644 • SSM Agent 2.2.493.0 • AWS Tools for Windows PowerShell 3.3.270.0 <p>Windows Server, version 1709 and Windows Server, version 1803</p> <ul style="list-style-type: none"> • AMIs are now available. For more information, see Windows Server version 1709 and 1803 AMIs for Amazon EC2.

Release	Changes
2018.04.11	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to April 10, 2018 • EC2Config version 4.9.2586 • SSM Agent 2.2.392.0 • AWS Tools for Windows PowerShell 3.3.256.0 • AWS CloudFormation templates 1.4.30 • Serial INF and Intel Chipset INF configurations to support new instance types <p>SQL Server 2017</p> <ul style="list-style-type: none"> • Cumulative update 5 (CU5) <p>SQL Server 2016 SP1</p> <ul style="list-style-type: none"> • Cumulative update 8 (CU8)
2018.03.24	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to March 13, 2018 • EC2Config version 4.9.2565 • SSM Agent 2.2.355.0 • AWS Tools for Windows PowerShell 3.3.245.0 • AWS PV driver 8.2 • AWS ENA driver 1.2.3.0 • Amazon EC2 Hibernate Agent 1.0 (rollback from 2.1.0 in the 2018.03.16 AMI release) • AWS EC2WinUtilDriver 1.0.1 (for troubleshooting) <p>Windows Server 2016</p> <ul style="list-style-type: none"> • EC2Launch 1.3.2000080
2018.03.16	AWS has removed all Windows AMIs dated 2018.03.16 due to an issue with an unquoted path in the configuration for the Amazon EC2 Hibernate Agent. For more information, see Issue with the Hibernate Agent (2018.03.16 AMIs) (p. 103) .
2018.03.06	All AMIs
2018.02.23	All AMIs
	<ul style="list-style-type: none"> • AWS PV driver 8.2.1
	<ul style="list-style-type: none"> • AWS PV driver 7.4.6 (rollback from 8.2 in the 2018.02.13 AMI release)

Release	Changes
2018.02.13	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to February 13, 2018 • EC2Config version 4.9.2400 • SSM Agent 2.2.160.0 • AWS Tools for Windows PowerShell 3.3.225.1 • AWS PV driver 8.2 • AWS ENA driver 1.2.3.0 • AWS NVMe driver 1.0.0.146 • Amazon EC2 HibernateAgent 1.0.0 <p>Windows Server 2016</p> <ul style="list-style-type: none"> • EC2Launch 1.3.740
2018.01.12	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to January 9, 2018
2018.01.05	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to January 2018 • Registry settings to enable mitigations for the Spectre and Meltdown exploits • AWS Tools for Windows PowerShell 3.3.215 • EC2Config version 4.9.2262

Monthly AMI Updates for 2017

For more information about Microsoft updates, see [Description of Software Update Services and Windows Server Update Services changes in content for 2017](#).

Release	Changes
2017.12.13	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to December 12, 2017 • EC2Config version 4.9.2218 • AWS CloudFormation templates 1.4.27 • AWS NVMe driver 1.02 • SSM Agent 2.2.93.0 • AWS Tools for Windows PowerShell 3.3.201
2017.11.29	<p>All AMIs</p> <ul style="list-style-type: none"> • Removed components for Volume Shadow Copy Service (VSS) included in 2017.11.18 and 2017.11.19 due to a compatibility issue with Windows Backup.
2017.11.19	<p>All AMIs</p> <ul style="list-style-type: none"> • EC2 Hibernate Agent 1.0 (supports hibernation for Spot Instances)

Release	Changes
2017.11.18	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to November 14, 2017 • EC2Config version 4.9.2218 • SSM Agent 2.2.64.0 • AWS Tools for Windows PowerShell 3.3.182 • Elastic Network Adapter (ENA) driver 1.08 (rollback from 1.2.2 in the 2017.10.13 AMI release) • Query for the latest Windows AMI using Systems Manager Parameter Store <p>Windows Server 2016</p> <ul style="list-style-type: none"> • EC2Launch 1.3.640
2017.10.13	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to October 11, 2017 • EC2Config version 4.9.2188 • SSM Agent 2.2.30.0 • AWS CloudFormation templates 1.4.24 • Elastic Network Adapter (ENA) driver 1.2.2. (Windows Server 2008 R2 through Windows Server 2016)
2017.10.04	<p>Microsoft SQL Server</p> <p>Windows Server 2016 with Microsoft SQL Server 2017 AMIs are now public in all regions.</p> <ul style="list-style-type: none"> • Windows_Server-2016-English-Full-SQL_2017_Enterprise-2017.10.04 • Windows_Server-2016-English-Full-SQL_2017_Standard-2017.10.04 • Windows_Server-2016-English-Full-SQL_2017_Web-2017.10.04 • Windows_Server-2016-English-Full-SQL_2017_Express-2017.10.04 <p>Microsoft SQL Server 2017 supports the following features:</p> <ul style="list-style-type: none"> • Machine Learning Services with Python (ML and AI) and R language support • Automatic database tuning • Clusterless Availability Groups • Runs on Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), and Ubuntu. For more information, see the following Microsoft article: Installation guidance for SQL Server on Linux. Not supported on Amazon Linux. • Windows-Linux cross-OS migrations • Resumable online index rebuild • Improved adaptive query processing • Graph data support

Release	Changes
2017.09.13	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to September 13, 2017 • EC2Config version 4.9.2106 • SSM Agent 2.0.952.0 • AWS Tools for Windows PowerShell 3.3.143 • AWS CloudFormation templates 1.4.21
2017.08.09	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to August 9, 2017 • EC2Config version 4.9.2016 • SSM Agent 2.0.879.0 <p>Windows Server 2012 R2</p> <ul style="list-style-type: none"> • Due to an internal error, these AMIs were released with an older version of AWS Tools for Windows PowerShell, 3.3.58.0.
2017.07.13	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to July 13, 2017 • EC2Config version 4.9.1981 • SSM Agent 2.0.847.0 <p>Windows Server 2016</p> <ul style="list-style-type: none"> • Intel SRIOV Driver 2.0.210.0
2017.06.14	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to June 14, 2017 • Updates for .NET Framework 4.7 installed from Windows Update • Microsoft updates to address the "privilege not held" error using the PowerShell Stop-Computer cmdlet. For more information, see Privilege not held error on the Microsoft site. • EC2Config version 4.9.1900 • SSM Agent 2.0.805.0 • AWS Tools for Windows PowerShell 3.3.99.0 • Internet Explorer 11 for the desktop is the default, instead of the immersive Internet Explorer <p>Windows Server 2016</p> <ul style="list-style-type: none"> • EC2Launch 1.3.610
2017.05.30	The Windows_Server-2008-SP2-English-32Bit-Base-2017.05.10 AMI was updated to the Windows_Server-2008-SP2-English-32Bit-Base-2017.05.30 AMI to resolve an issue with password generation.

Release	Changes
2017.05.22	<p>The Windows_Server-2016-English-Full-Base-2017.05.10 AMI was updated to the Windows_Server-2016-English-Full-Base-2017.05.22 AMI after some log cleaning.</p>
2017.05.10	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to May 9, 2017 • AWS PV Driver v7.4.6 • AWS Tools for Windows PowerShell 3.3.83.0 <p>Windows Server 2016</p> <ul style="list-style-type: none"> • SSM Agent 2.0.767
2017.04.12	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to April 11, 2017 • AWS Tools for Windows PowerShell 3.3.71.0 • AWS CloudFormation templates 1.4.18 <p>Windows Server 2003 to Windows Server 2012</p> <ul style="list-style-type: none"> • EC2Config version 4.9.1775 • SSM Agent 2.0.761.0 <p>Windows Server 2016</p> <ul style="list-style-type: none"> • SSM Agent 2.0.730.0
2017.03.15	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to March 14, 2017 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation templates <p>Windows Server 2003 to Windows Server 2012</p> <ul style="list-style-type: none"> • EC2Config version 4.7.1631 • SSM Agent 2.0.682.0 <p>Windows Server 2016</p> <ul style="list-style-type: none"> • SSM Agent 2.0.706.0 • EC2Launch v1.3.540
2017.02.21	<p>Microsoft recently announced that they will not release monthly patches or security updates for the month of February. All February patches and security updates will be included in the March update.</p> <p>Amazon Web Services did not release updated Windows Server AMIs in February.</p>

Release	Changes
2017.01.11	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to January 10, 2017 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation templates <p>Windows Server 2003 to Windows Server 2012</p> <ul style="list-style-type: none"> • EC2Config version 4.2.1442 • SSM Agent 2.0.599.0

Monthly AMI Updates for 2016

For more information about Microsoft updates, see [Description of Software Update Services and Windows Server Update Services changes in content for 2016](#).

Release	Changes
2016.12.14	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to December 13, 2016 • Current AWS Tools for Windows PowerShell <p>Windows Server 2003 to Windows Server 2012</p> <ul style="list-style-type: none"> • Released EC2Config version 4.1.1396 • Elastic Network Adapter (ENA) driver 1.0.9.0 (Windows Server 2008 R2 only) <p>Windows Server 2016</p> <p>New AMIs available in all regions:</p> <ul style="list-style-type: none"> • Windows_Server-2016-English-Core-Base <p>Microsoft SQL Server</p> <p>All Microsoft SQL Server AMIs with the latest service pack are now public in all regions. These new AMIs replace old SQL Service Pack AMIs going forward.</p> <ul style="list-style-type: none"> • Windows_Server-2008-R2_SP1-English-64Bit-SQL_2012_SP3_edition-2016.12.14 • Windows_Server-2012-RTM-English-64Bit-SQL_2012_SP3_edition-2016.12.14 • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2014_SP2_edition-2016.12.14 • Windows_Server-2012-RTM-English-64Bit-SQL_2014_SP2_edition-2016.12.14 • Windows_Server-2012-R2_RTM-English-64Bit-SQL_2016_SP1_edition-2016.12.14

Release	Changes
	<ul style="list-style-type: none"> Windows_Server-2016-English-Full-SQL_2016_SP1_<i>edition</i>-2016.12.14 <p>SQL Server 2016 SP1 is a major release. The following features, which were previously available in Enterprise edition only, are now enabled in Standard, Web, and Express editions with SQL Server 2016 SP1:</p> <ul style="list-style-type: none"> Row-level security Dynamic Data Masking Change Data Capture Database snapshot Column store Partitioning Compression In Memory OLTP Always Encrypted
2016.11.23	<p>Windows Server 2003 to Windows Server 2012</p> <ul style="list-style-type: none"> Released EC2Config version 4.1.1378 The AMIs released this month, and going forward, use the EC2Config service to process boot-time configurations and SSM Agent to process Amazon EC2 Run Command and SSM Config requests. EC2Config no longer processes requests for Run Command and SSM Config. The latest EC2Config installer installs SSM Agent side-by-side with the EC2Config service. For more information, see EC2Config and AWS Systems Manager (p. 386).
2016.11.09	<p>All AMIs</p> <ul style="list-style-type: none"> Microsoft security updates current to November 8 2016 Released AWS PV driver, version 7.4.3.0 for Windows 2008 R2 and later Current AWS Tools for Windows PowerShell
2016.10.18	<p>All AMIs</p> <ul style="list-style-type: none"> Microsoft security updates current to October 12, 2016 Current AWS Tools for Windows PowerShell <p>Windows Server 2016</p> <ul style="list-style-type: none"> Released AMIs for Windows Server 2016. These AMIs include significant changes. For example, they don't include the EC2Config service. For more information, see Changes in Windows Server 2016 AMIs (p. 102).
2016.9.14	<p>All AMIs</p> <ul style="list-style-type: none"> Microsoft security updates current to September 13, 2016 Current AWS Tools for Windows PowerShell Renamed AMI Windows_Server-2012-RTM-Japanese-64Bit-SQL_2008_R3_SP2_Standard to Windows_Server-2012-RTM-Japanese-64Bit-SQL_2008_R2_SP3_Standard

Release	Changes
2016.8.26	All Windows Server 2008 R2 AMIs dated 2016.08.11 were updated to fix a known issue. New AMIs are dated 2016.08.25.
2016.8.11	<p>All AMIs</p> <ul style="list-style-type: none"> • Ec2Config v3.19.1153 • Microsoft security updates current to August 10, 2016 • Enabled the registry key User32 exception handler hardening feature in Internet Explorer for MS15-124 <p>Windows Server 2008 R2, Windows Server 2012 RTM, and Windows Server 2012 R2</p> <ul style="list-style-type: none"> • Elastic Network Adapter (ENA) Driver 1.0.8.0 • ENA AMI property set to enabled • AWS PV Driver for Windows Server 2008 R2 was re-released this month because of a known issue. Windows Server 2008 R2 AMI's were removed in July because of this issue.
2016.8.2	All Windows Server 2008 R2 AMIs for July were removed and rolled back to AMIs dated 2016.06.15, because of an issue discovered in the AWS PV driver. The AWS PV driver issue has been fixed. The August AMI release will include Windows Server 2008 R2 AMIs with the fixed AWS PV driver and July/August Windows updates.
2016.7.26	<p>All AMIs</p> <ul style="list-style-type: none"> • Ec2Config v3.18.1118 • 2016.07.13 AMIs were missing security patches. AMIs were re-patched. Additional processes were put in place to verify successful patch installations going forward.
2016.7.13	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to July 2016 • Current AWS Tools for Windows PowerShell • Updated AWS PV Driver 7.4.2.0 • AWS PV Driver for Windows Server 2008 R2
2016.6.16	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to June 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.17.1032 <p>Microsoft SQL Server</p> <ul style="list-style-type: none"> • Released 10 AMIs that include 64-bit versions of Microsoft SQL Server 2016. If using the Amazon EC2 console, navigate to Images, AMIs, Public Images, and type Windows_Server-2012-R2_RTM-English-64Bit-SQL_2016_Standard in the search bar. For more information, see What's New in SQL Server 2016 on MSDN.

Release	Changes
2016.5.11	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to May 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.16.930 • MS15-011 Active Directory patch installed <p>Windows Server 2012 R2</p> <ul style="list-style-type: none"> • Intel SRIOV Driver 1.0.16.1
2016.4.13	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to April 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.15.880
2016.3.9	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to March 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.14.786
2016.2.10	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to February 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.13.727
2016.1.25	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to January 2016 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.12.649
2016.1.5	<p>All AMIs</p> <ul style="list-style-type: none"> • Current AWS Tools for Windows PowerShell

Monthly AMI Updates for 2015

For more information about Microsoft updates, see [Description of Software Update Services and Windows Server Update Services changes in content for 2015](#).

Release	Changes
2015.12.15	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to December 2015 • Current AWS Tools for Windows PowerShell

Release	Changes
2015.11.11	All AMIs <ul style="list-style-type: none"> • Microsoft security updates current to November 2015 • Current AWS Tools for Windows PowerShell • EC2Config service version 3.11.521 • CFN Agent updated to latest version
2015.10.26	Corrected boot volume sizes of base AMIs to be 30GB instead of 35GB
2015.10.14	All AMIs <ul style="list-style-type: none"> • Microsoft security updates current to October 2015 • EC2Config service version 3.10.442 • Current AWS Tools for Windows PowerShell • Updated SQL Service Packs to latest versions for all SQL variants • Removed old entries in Event Logs • AMI Names have been changed to reflect the latest service pack. For example, the latest AMI with Server 2012 and SQL 2014 Standard is named "Windows_Server-2012-RTM-English-64Bit-SQL_2014_SP1_Standard-2015.10.26", not "Windows_Server-2012-RTM-English-64Bit-SQL_2014_RTM_Standard-2015.10.26".
2015.9.9	All AMIs <ul style="list-style-type: none"> • Microsoft security updates current to September 2015 • EC2Config service version 3.9.359 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts
2015.8.18	All AMIs <ul style="list-style-type: none"> • Microsoft security updates current to August 2015 • EC2Config service version 3.8.294 • Current AWS Tools for Windows PowerShell <p>Only AMIs with Windows Server 2012 and Windows Server 2012 R2</p> <ul style="list-style-type: none"> • AWS PV Driver 7.3.2
2015.7.21	All AMIs <ul style="list-style-type: none"> • Microsoft security updates current to July 2015 • EC2Config service version 3.7.308 • Current AWS Tools for Windows PowerShell • Modified AMI descriptions of SQL images for consistency

Release	Changes
2015.6.10	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to June 2015 • EC2Config service version 3.6.269 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts <p>Only AMIs with Windows Server 2012 R2</p> <ul style="list-style-type: none"> • AWS PV Driver 7.3.1
2015.5.13	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to May 2015 • EC2Config service version 3.5.228 • Current AWS Tools for Windows PowerShell
2015.04.15	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to April 2015 • EC2Config service version 3.3.174 • Current AWS Tools for Windows PowerShell
2015.03.11	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to March 2015 • EC2Config service version 3.2.97 • Current AWS Tools for Windows PowerShell <p>Only AMIs with Windows Server 2012 R2</p> <ul style="list-style-type: none"> • AWS PV Driver 7.3.0
2015.02.11	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to February 2015 • EC2Config service version 3.0.54 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts
2015.01.14	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to January 2015 • EC2Config service version 2.3.313 • Current AWS Tools for Windows PowerShell • Current AWS CloudFormation helper scripts

Monthly AMI Updates for 2014

For more information about Microsoft updates, see [Description of Software Update Services and Windows Server Update Services changes in content for 2014](#).

Release	Changes
2014.12.10	All AMIs <ul style="list-style-type: none">Microsoft security updates current to December 2014EC2Config service version 2.2.12Current AWS Tools for Windows PowerShell
2014.11.19	All AMIs <ul style="list-style-type: none">Microsoft security updates current to November 2014EC2Config service version 2.2.11Current AWS Tools for Windows PowerShell
2014.10.15	All AMIs <ul style="list-style-type: none">Microsoft security updates current to October 2014EC2Config service version 2.2.10Current AWS Tools for Windows PowerShell Only AMIs with Windows Server 2012 R2 <ul style="list-style-type: none">AWS PV Driver 7.2.4.1 (resolves the issues with Plug and Play Cleanup, which is now enabled by default)
2014.09.10	All AMIs <ul style="list-style-type: none">Microsoft security updates current to September 2014EC2Config service version 2.2.8Current AWS Tools for Windows PowerShell Only AMIs with Windows Server 2012 R2 <ul style="list-style-type: none">Disable Plug and Play Cleanup (see Important information)AWS PV Driver 7.2.2.1 (resolves issues with the uninstaller)
2014.08.13	All AMIs <ul style="list-style-type: none">Microsoft security updates current to August 2014EC2Config service version 2.2.7Current AWS Tools for Windows PowerShell Only AMIs with Windows Server 2012 R2 <ul style="list-style-type: none">AWS PV Driver 7.2.2.1 (improves disk performance, resolves issues with reconnecting multiple network interfaces and lost network settings)

Release	Changes
2014.07.10	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to July 2014 EC2Config service version 2.2.5 Current AWS Tools for Windows PowerShell
2014.06.12	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to June 2014 EC2Config service version 2.2.4 Removed NVIDIA drivers (except for Windows Server 2012 R2 AMIs) Current AWS Tools for Windows PowerShell
2014.05.14	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to May 2014 EC2Config service version 2.2.2 Current AWS Tools for Windows PowerShell AWS CloudFormation helper scripts version 1.4.0
2014.04.09	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to April 2014 Current AWS Tools for Windows PowerShell Current AWS CloudFormation helper scripts
2014.03.12	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to March 2014
2014.02.12	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to February 2014 EC2Config service version 2.2.1 Current AWS Tools for Windows PowerShell KB2634328 Remove the BCDEdit useplatformclock value <p>Only AMIs with Microsoft SQL Server</p> <ul style="list-style-type: none"> Microsoft SQL Server 2012 SP1 cumulative update package 8 Microsoft SQL Server 2008 R2 cumulative update package 10

Monthly AMI Updates for 2013

For more information about Microsoft updates, see [Description of Software Update Services and Windows Server Update Services changes in content for 2013](#).

Release	Changes
2013.11.13	<p>All AMIs</p> <ul style="list-style-type: none"> Microsoft security updates current to November 2013 EC2Config service version 2.1.19 Current AWS Tools for Windows PowerShell Configure NTP to synchronize the time once a day (the default is every seven days) <p>Only AMIs with Windows Server 2012</p> <ul style="list-style-type: none"> Clean up the WinSXS folder using the following command: <code>dism /online / cleanup-image /StartComponentCleanup</code>
2013.09.11	<p>All AMIs</p> <ul style="list-style-type: none"> Microsoft security updates current to September 2013 EC2Config service version 2.1.18 Current AWS Tools for Windows PowerShell AWS CloudFormation helper scripts version 1.3.15
2013.07.10	<p>All AMIs</p> <ul style="list-style-type: none"> Microsoft security updates current to July 2013 EC2Config service version 2.1.16 Expanded the root volume to 50 GB Set the page file to 512 MB, expanding to 8 GB as needed Current AWS Tools for Windows PowerShell
2013.06.12	<p>All AMIs</p> <ul style="list-style-type: none"> Microsoft security updates current to June 2013 Current AWS Tools for Windows PowerShell <p>Only AMIs with Microsoft SQL Server</p> <ul style="list-style-type: none"> Microsoft SQL Server 2012 SP1 with cumulative update package 4
2013.05.15	<p>All AMIs</p> <ul style="list-style-type: none"> Microsoft security updates current to May 2013 EC2Config service version 2.1.15 All instance store volumes attached by default Remote PowerShell enabled by default Current AWS Tools for Windows PowerShell
2013.04.14	<p>All AMIs</p> <ul style="list-style-type: none"> Microsoft security updates current to April 2013 Current AWS Tools for Windows PowerShell AWS CloudFormation helper scripts version 1.3.14

Release	Changes
2013.03.14	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to March 2013 • EC2Config service version 2.1.14 • Citrix Agent with CPU heartbeat fix • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.3.11
2013.02.22	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to February 2013 • KB2800213 • Windows PowerShell 3.0 upgrade • EC2Config service version 2.1.13 • Citrix Agent with time fix • Citrix PV drivers dated 2011.07.19 • Current AWS Tools for Windows PowerShell • AWS CloudFormation helper scripts version 1.3.8 <p>Only AMIs with Microsoft SQL Server</p> <ul style="list-style-type: none"> • Microsoft SQL Server 2012 cumulative update package 5

Monthly AMI Updates for 2012

For more information about Microsoft updates, see [Description of Software Update Services and Windows Server Update Services changes in content for 2012](#).

Release	Changes
2012.12.12	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to December 2012 • Set the ActiveTimeBias registry value to 0 • Disable IPv6 for the network adapter • EC2Config service version 2.1.9 • Add AWS Tools for Windows PowerShell and set the policy to allow import-module
2012.11.15	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to November 2012 • EC2Config service version 2.1.7
2012.10.10	<p>All AMIs</p> <ul style="list-style-type: none"> • Microsoft security updates current to October 2012

Release	Changes
2012.08.15	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to August 2012 EC2Config service version 2.1.2 KB2545227
2012.07.11	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to July 2012
2012.06.12	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to June 2012 Set page file to 4 GB Remove installed language packs Set performance option to "Adjust for best performance" Set the screen saver to no longer display the logon screen on resume Remove previous RedHat driver versions using pnputil Remove duplicate bootloaders and set bootstatuspolicy to ignoreallfailures using bcdedit
2012.05.10	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to May 2012 EC2Config service version 2.1.0
2012.04.11	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to April 2012 KB2582281 Current version of EC2Config System time in UTC instead of GMT
2012.03.13	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to March 2012
2012.02.24	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to February 2012 Standardize AMI names and descriptions
2012.01.12	All AMIs <ul style="list-style-type: none"> Microsoft security updates current to January 2012 RedHat PV driver version 1.3.10

Monthly AMI Updates for 2011 and earlier

Release	Changes
2011.09.11	All AMIs <ul style="list-style-type: none">Microsoft security updates current to September 2011
1.04	All AMIs <ul style="list-style-type: none">Current Microsoft security updatesUpdate network driverFix issue with instances in a VPC losing connectivity when changing the time zone of the instance
1.02	All AMIs <ul style="list-style-type: none">Current Microsoft security updatesUpdate network driverAdd support for licensing activation for instances in a VPC
1.01	All AMIs <ul style="list-style-type: none">Current Microsoft security updatesFix issue with password improperly generated while waiting for network availability
1.0	All AMIs <ul style="list-style-type: none">Initial release

Changes in Windows Server 2016 AMIs

AWS provides AMIs for Windows Server 2016. These AMIs include the following high-level changes from earlier Windows AMIs:

- To accommodate the change from .NET Framework to .NET Core, the EC2Config service has been deprecated on Windows Server 2016 AMIs and replaced by EC2Launch. EC2Launch is a bundle of Windows PowerShell scripts that perform many of the tasks performed by the EC2Config service. For more information, see [Configuring a Windows Instance Using EC2Launch \(p. 374\)](#).
- On earlier versions of Windows Server AMIs, you can use the EC2Config service to join an EC2 instance to a domain and configure integration with Amazon CloudWatch. On Windows Server 2016 AMIs, the Amazon EC2 Systems Manager (SSM) agent performs these tasks. This means that you must use either Amazon EC2 Run Command or SSM Config to join an EC2 instance to a domain or configure integration with Amazon CloudWatch on Windows Server 2016 instances. For more information about configuring instances to send log data to CloudWatch, see [Sending Logs, Events, and Performance Counters to Amazon CloudWatch \(p. 526\)](#). For information about joining an EC2 instance to a domain, see [Joining a Windows Instance to an AWS Directory Service Domain](#).

Other Differences

Note these additional important differences for instances created from Windows Server 2016 AMIs.

- By default, EC2Launch does not initialize secondary EBS volumes. You can configure EC2Launch to initialize disks automatically by either scheduling the script to run or by calling EC2Launch in user data. For the procedure to initialize disks using EC2Launch, see "Initialize Drives and Drive Letter Mappings" in [Configuring EC2Launch \(p. 376\)](#).
- If you previously enabled CloudWatch integration on your instances by using a local configuration file (AWS.EC2.Windows.CloudWatch.json), you can configure the file to work with the SSM Agent on instances created from Windows Server 2016 AMIs. For more information, see [Use SSM Agent to Configure CloudWatch \(p. 538\)](#).

For more information, see [Windows Server 2016](#) on Microsoft.com.

Docker Container Conflict on Windows Server 2016 Instances

If you run the Docker service on Windows Server 2016 AMIs, the service is configured to use a different CIDR value than the default internal IP address prefix value. The default value is 172.16.0.0/12. Windows Server 2016 AMIs use 172.17.0.0/16 to avoid a conflict with the default Amazon EC2 VPC/subnet. If you don't change VPC/subnet settings for your EC2 instances, then you don't need to do anything. The conflict is essentially avoided because of the different CIDR values. If you do change VPC/subnet settings, be aware of these internal IP address prefix values and avoid creating a conflict. For more information, read the following section.

Important

If you plan to run Docker on a Windows Server 2016 instance, you must create the instance from the following Amazon Machine Image (AMI) or an AMI based on an image with `Windows_Server-2016-English-Full-Containers` in the name. Otherwise, if you use a different Windows Server 2016 AMI, instances fail to boot correctly after installing Docker and then running Sysprep.

Issue with the Hibernate Agent (2018.03.16 AMIs)

After the release of the 2018.03.16 Windows AMIs, we discovered an unquoted path in the configuration of the Amazon EC2 Hibernate Agent. The agent was included in the AMIs for Windows Server 2008 through Windows Server 2016. This issue does not impact the AMIs for Windows Server 2003.

AWS has removed the Windows AMIs dated 2018.03.16. To be notified when new Windows AMIs are available, see [Subscribing to Windows AMI Notifications \(p. 78\)](#).

To mitigate the issue, you can use one of the following procedures to add the missing quotation marks. If the agent is running, you must also restart the agent. Alternatively, you can terminate any instances that you launched from a 2018.03.16 Windows AMI and replace them with instances launched using a different AMI.

Windows PowerShell

1. On your Windows instance, open Windows Powershell.
2. Use the following command to update the configuration, adding the missing quotation marks:

```
cmd /c 'sc config EC2HibernateAgent binPath="\%ProgramFiles%\Amazon\Hibernate\EC2HibernateAgent.exe\""
```

3. Use the following command to view the updated configuration:

```
(Get-ItemProperty -Path Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EC2HibernateAgent).ImagePath
```

Verify that the response is enclosed in quotation marks, as shown in the following example:

```
"C:\Program Files\Amazon\Hibernate\EC2HibernateAgent.exe"
```

4. Use the following command to check whether Status is Running:

```
Get-Service EC2HibernateAgent
```

If the agent is running, you must restart it using the following command so that the change takes effect:

```
Restart-Service EC2HibernateAgent
```

Command Prompt

1. On your Windows instance, open a Command Prompt window.
2. Use the following command to update the configuration, adding the missing quotation marks:

```
sc config EC2HibernateAgent binPath="\"%ProgramFiles%\Amazon\Hibernate\EC2HibernateAgent.exe\""
```

3. Use the following command to view the updated configuration:

```
sc qc EC2HibernateAgent
```

Verify that the path in BINARY_PATH_NAME is enclosed in quotation marks, as shown in the following example:

```
"C:\Program Files\Amazon\Hibernate\EC2HibernateAgent.exe"
```

4. Use the following command to check whether STATE is RUNNING:

```
sc query EC2HibernateAgent
```

If the agent is running, you must restart it using the following command so that the change takes effect:

```
sc stop EC2HibernateAgent && sc start EC2HibernateAgent
```

Create a Standard Amazon Machine Image Using Sysprep

The Microsoft System Preparation (Sysprep) tool simplifies the process of duplicating a customized installation of Windows. We recommend that you use Sysprep to create a standardized Amazon Machine Image (AMI). You can then create new Amazon EC2 instances for Windows from this standardized image.

We also recommend that you run Sysprep with EC2Launch (Windows Server 2016) or the EC2Config service (prior to Windows Server 2016).

Important

Don't use Sysprep to create an instance backup. Sysprep removes system-specific information; removing this information might have unintended consequences for an instance backup.

Contents

- [Before You Begin \(p. 105\)](#)
- [Using Sysprep with the EC2Config Service \(p. 105\)](#)
- [Run Sysprep with the EC2Config Service \(p. 108\)](#)
- [Troubleshooting Sysprep with EC2Config \(p. 109\)](#)

Before You Begin

- Learn more about [Sysprep](#) on Microsoft TechNet.
- Learn which [server roles are supported for Sysprep](#).
- The procedures on this page apply to E2Config. With Windows Server 2016, see [Using Sysprep with EC2Launch \(p. 379\)](#).

Using Sysprep with the EC2Config Service

Learn the details of the different Sysprep execution phases and the tasks performed by the EC2Config service as the image is prepared.

Sysprep Phases

Sysprep runs through the following phases:

- **Generalize:** The tool removes image-specific information and configurations. For example, Sysprep removes the security identifier (SID), the computer name, the event logs, and specific drivers, to name a few. After this phase is completed, the operating system (OS) is ready to create an AMI.

Note

When you run Sysprep with the EC2Config service, the system prevents drivers from being removed because the PersistAllDeviceInstalls setting is set to true by default.

- **Specialize:** Plug and Play scans the computer and installs drivers for any detected devices. The tool generates OS requirements like the computer name and SID. Optionally, you can execute commands in this phase.
- **Out-of-Box Experience (OOBE):** The system runs an abbreviated version of Windows Setup and asks the user to enter information such as a system language, the time zone, and a registered organization. When you run Sysprep with EC2Config, the answer file automates this phase.

Sysprep Actions

Sysprep and the EC2Config service perform the following actions when preparing an image.

1. When you choose **Shutdown with Sysprep** in the **EC2 Service Properties** dialog box, the system runs the `ec2config.exe -sysprep` command.
2. The EC2Config service reads the content of the `BundleConfig.xml` file. This file is located in the following directory, by default: `C:\Program Files\Amazon\Ec2ConfigService\Settings`.

The `BundleConfig.xml` file includes the following settings. You can change these settings:

- **AutoSysprep:** Indicates whether to use Sysprep automatically. You do not need to change this value if you are running Sysprep from the EC2 Service Properties dialog box. The default value is No.

- **SetRDPCertificate:** Sets a self-signed certificate for the Remote Desktop server. This enables you to securely use the Remote Desktop Protocol (RDP) to connect to the instance. Change the value to Yes if new instances should use a certificate. This setting is not used with Windows Server 2008 or Windows Server 2012 instances because these operating systems can generate their own certificates. The default value is No.
- **SetPasswordAfterSysprep:** Sets a random password on a newly launched instance, encrypts it with the user launch key, and outputs the encrypted password to the console. Change the value to No if new instances should not be set to a random encrypted password. The default value is Yes.
- **PreSysprepRunCmd:** The location of the command to run. The command is located in the following directory, by default: C:\Program Files\Amazon\Ec2ConfigService\Scripts\BeforeSysprep.cmd

3. The system executes `BeforeSysprep.cmd`. This command creates a registry key as follows:

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 1 /f"
```

The registry key disables RDP connections until they are re-enabled. Disabling RDP connections is a necessary security measure because, during the first boot session after Sysprep has run, there is a short period of time where RDP allows connections and the Administrator password is blank.

4. The EC2Config service calls Sysprep by running the following command:

```
sysprep.exe /unattend: "C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml" /oobe /generalize /shutdown
```

Generalize Phase

- The tool removes image-specific information and configurations such as the computer name and the SID. If the instance is a member of a domain, it is removed from the domain. The `sysprep2008.xml` answer file includes the following settings which affect this phase:
 - **PersistAllDeviceInstalls:** This setting prevents Windows Setup from removing and reconfiguring devices, which speeds up the image preparation process because Amazon AMIs require certain drivers to run and re-detection of those drivers would take time.
 - **DoNotCleanUpNonPresentDevices:** This setting retains Plug and Play information for devices that are not currently present.
- Sysprep shuts down the OS as it prepares to create the AMI. The system either launches a new instance or starts the original instance.

Specialize Phase

The system generates OS specific requirements such as a computer name and a SID. The system also performs the following actions based on configurations that you specify in the `sysprep2008.xml` answer file.

- **CopyProfile:** Sysprep can be configured to delete all user profiles, including the built-in Administrator profile. This setting retains the built-in Administrator account so that any customizations you made to that account are carried over to the new image. The default value is True.

If you don't have specific user-profile customizations that you want to carry over to the new image then change this setting to False. Sysprep will remove all user profiles; this saves time and disk space.

- **TimeZone:** The time zone is set to Coordinate Universal Time (UTC) by default.
- **Synchronous command with order 1:** The system executes the following command that enables the administrator account and specifies the password requirement.

```
net user Administrator /ACTIVE:YES /LOGONPASSWORDCHG:NO /EXPIRES:NEVER /  
PASSWORDREQ:YES
```

- **Synchronous command with order 2:** The system scrambles the administrator password. This security measure is designed to prevent the instance from being accessible after Sysprep completes if you did not enable the ec2setpassword setting.

C:\Program Files\Amazon\Ec2ConfigService\ScramblePassword.exe" -u Administrator

- **Synchronous command with order 3:** The system executes the following command:

C:\Program Files\Amazon\Ec2ConfigService\Scripts\SysprepSpecializePhase.cmd

This command adds the following registry key, which re-enables RDP:

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v  
fDenyTSConnections /t REG_DWORD /d 0 /f
```

OOBE Phase

1. Using the EC2Config service answer file, the system specifies the following configurations:

- <InputLocale>en-US</InputLocale>
- <SystemLocale>en-US</SystemLocale>
- <UILanguage>en-US</UILanguage>
- <UserLocale>en-US</UserLocale>
- <HideEULAPage>true</HideEULAPage>
- <HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>
- <NetworkLocation>Other</NetworkLocation>
- <ProtectYourPC>3</ProtectYourPC>
- <BluetoothTaskbarIconEnabled>false</BluetoothTaskbarIconEnabled>
- <TimeZone>UTC</TimeZone>
- <RegisteredOrganization>Amazon.com</RegisteredOrganization>
- <RegisteredOwner>Amazon</RegisteredOwner>

Note

During the generalize and specialize phases the EC2Config service monitors the status of the OS. If EC2Config detects that the OS is in a Sysprep phase, then it publishes the following message to the system log:

EC2ConfigMonitorState: 0 Windows is being configured. SysprepState=IMAGE_STATE_UNDEPLOYABLE

2. After the OOBE phase completes, the system executes the SetupComplete.cmd from the following location: C:\Windows\Setup\Scripts\SetupComplete.cmd. In Amazon public AMIs before April 2015 this file was empty and executed nothing on the image. In public AMIs dated after April 2015, the file includes the following value: **call "C:\Program Files\Amazon\Ec2ConfigService\Scripts\PostSysprep.cmd"**.
3. The system executes the PostSysprep.cmd, which performs the following operations:
 - Sets the local Administrator password to not expire. If the password expired, Administrators might not be able to log on.
 - Sets the MSSQLServer machine name (if installed) so that the name will be in sync with the AMI.

Post Sysprep

After Sysprep completes, the EC2Config services sends the following message to the console output:

```
Windows sysprep configuration complete.  
Message: Sysprep Start  
Message: Sysprep End
```

EC2Config then performs the following actions:

1. Reads the content of the config.xml file and lists all enabled plug-ins.
2. Executes all "Before Windows is ready" plug-ins at the same time.
 - Ec2SetPassword
 - Ec2SetComputerName
 - Ec2InitializeDrives
 - Ec2EventLog
 - Ec2ConfigureRDP
 - Ec2OutputRDPCert
 - Ec2SetDriveLetter
 - Ec2WindowsActivate
 - Ec2DynamicBootVolumeSize
3. After it is finished, sends a "Windows is ready" message to the instance system logs.
4. Runs all "After Windows is ready" plug-ins at the same time.
 - AWS CloudWatch logs
 - UserData
 - Simple Systems Manager (SSM)

For more information about Windows plug-ins, see [Configuring a Windows Instance Using the EC2Config Service \(p. 383\)](#).

Run Sysprep with the EC2Config Service

Use the following procedure to create a standardized AMI using Sysprep and the EC2Config service.

1. In the Amazon EC2 console locate or [create \(p. 65\)](#) an AMI that you want to duplicate.
2. Launch and connect to your Windows instance.
3. Customize it.
4. Specify configuration settings in the EC2Config service answer file:

`C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml`
5. From the Windows **Start** menu, choose **All Programs**, and then choose **EC2ConfigService Settings**.
6. Choose the **Image** tab in the **Ec2 Service Properties** dialog box. For more information about the options and settings in the Ec2 Service Properties dialog box, see [Ec2 Service Properties \(p. 383\)](#).
7. Select an option for the Administrator password, and then select **Shutdown with Sysprep** or **Shutdown without Sysprep**. EC2Config edits the settings files based on the password option that you selected.
 - **Random:** EC2Config generates a password, encrypts it with user's key, and displays the encrypted password to the console. We disable this setting after the first launch so that this password persists if the instance is rebooted or stopped and started.
 - **Specify:** The password is stored in the Sysprep answer file in unencrypted form (clear text). When Sysprep runs next, it sets the Administrator password. If you shut down now, the password is set immediately. When the service starts again, the Administrator password is removed. It's important to remember this password, as you can't retrieve it later.

- **Keep Existing:** The existing password for the Administrator account doesn't change when Sysprep is run or EC2Config is restarted. It's important to remember this password, as you can't retrieve it later.
8. Choose **OK**.

When you are asked to confirm that you want to run Sysprep and shut down the instance, click **Yes**. You'll notice that EC2Config runs Sysprep. Next, you are logged off the instance, and the instance is shut down. If you check the **Instances** page in the Amazon EC2 console, the instance state changes from running to stopping, and then finally to stopped. At this point, it's safe to create an AMI from this instance.

You can manually invoke the Sysprep tool from the command line using the following command:

```
"%programfiles%\amazon\ec2configservice\"ec2config.exe -sysprep"
```

Note

The double quotation marks in the command are not required if your CMD shell is already in the C:\Program Files\Amazon\EC2ConfigService\ directory.

However, you must be very careful that the XML file options specified in the Ec2ConfigService\Settings folder are correct; otherwise, you might not be able to connect to the instance. For more information about the settings files, see [EC2Config Settings Files \(p. 390\)](#). For an example of configuring and then running Sysprep from the command line, see Ec2ConfigService\Scripts\InstallUpdates.ps1.

Troubleshooting Sysprep with EC2Config

If you experience problems or receive error messages during image preparations, review the following logs:

- %WINDIR%\Panther\Unattendgc
- %WINDIR%\System32\Sysprep\Panther
- "C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt"

If you receive an error message during image preparation with Sysprep, the OS might not be reachable. To review the log files, you must stop the instance, attach its root volume to another healthy instance as a secondary volume, and then review the logs mentioned earlier on the secondary volume.

If you locate errors in the Unattendgc log file, use the [Microsoft Error Lookup Tool](#) to get more details about the error. The following issue reported in the Unattendgc log file is typically the result of one or more corrupted user profiles on the instance:

```
Error [Shell Unattend] _FindLatestProfile failed (0x80070003) [gle=0x00000003]  
Error [Shell Unattend] CopyProfile failed (0x80070003) [gle=0x00000003]
```

There are two options for resolving this issue:

Option 1: Use Regedit on the instance to search for the following key. Verify that there are no profile registry keys for a deleted user:

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\ProfileList\
```

Option 2: Edit the EC2Config answer file (C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml) and change <CopyProfile>true</CopyProfile> to <CopyProfile>false</CopyProfile>. Run Sysprep again. Note that this configuration change will delete the built-in administrator user profile after Sysprep completes.

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 \(p. 12\)](#)
- [Getting Started with Amazon EC2 Windows Instances \(p. 20\)](#)
- [Instance Lifecycle \(p. 309\)](#)

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 \(p. 110\)](#).

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 \(p. 185\)](#).

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 \(p. 111\)](#)
- [Hardware Specifications \(p. 112\)](#)
- [Nitro-based Instances \(p. 113\)](#)
- [Networking and Storage Features \(p. 113\)](#)
- [Instance Limits \(p. 116\)](#)
- [Burstable Performance Instances \(p. 116\)](#)
- [General Purpose Instances \(p. 143\)](#)
- [Compute Optimized Instances \(p. 148\)](#)
- [Memory Optimized Instances \(p. 152\)](#)
- [Storage Optimized Instances \(p. 159\)](#)
- [Windows Accelerated Computing Instances \(p. 163\)](#)
- [T1 Micro Instances \(p. 170\)](#)
- [Changing the Instance Type \(p. 182\)](#)

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	m4.large m4.xlarge m4.2xlarge m4.4xlarge m4.10xlarge m4.16xlarge m5.large m5.xlarge m5.2xlarge m5.4xlarge m5.12xlarge m5.24xlarge m5a.large m5a.xlarge m5a.2xlarge m5a.4xlarge m5a.12xlarge m5a.24xlarge m5d.large m5d.xlarge m5d.2xlarge m5d.4xlarge m5d.12xlarge m5d.24xlarge t2.nano t2.micro t2.small t2.medium t2.large t2.xlarge t2.2xlarge t3.nano t3.micro t3.small t3.medium t3.large t3.xlarge t3.2xlarge
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 c5n.large c5n.xlarge c5n.2xlarge c5n.4xlarge c5n.9xlarge c5n.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 r5a.large r5a.xlarge r5a.2xlarge r5a.4xlarge r5a.12xlarge r5a.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

Instance Family	Current Generation Instance Types
	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
Accelerated computing	f1.2xlarge f1.4xlarge f1.16xlarge g3s.xlarge g3.4xlarge g3.8xlarge g3.16xlarge p2.xlarge p2.8xlarge p2.16xlarge p3.2xlarge p3.8xlarge p3.16xlarge p3dn.24xlarge
Bare metal	i3.metal u-6tb1.metal u-9tb1.metal u-12tb1.metal

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 hour, 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 \(p. 182\)](#).

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.

Nitro-based Instances

The Nitro system is a collection of AWS-built hardware and software components that enable high performance, high availability, and high security. In addition, the Nitro system provides bare metal capabilities that eliminate virtualization overhead and support workloads that require full access to host hardware.

Nitro Components

The following components are part of the Nitro system:

- Nitro hypervisor - A lightweight hypervisor that manages memory and CPU allocation and delivers performance that is indistinguishable from bare metal for most workloads.
- Nitro card
 - Local NVMe storage volumes
 - Networking hardware support
 - Management
 - Monitoring
 - Security
- Nitro security chip, integrated into the motherboard

Instance Types

The following instances are based on the Nitro system:

- C5, C5d, C5n, M5, M5a, M5d, p3dn.24xlarge, R5, R5a, R5d, T3, and z1d
- Bare metal: i3.metal, u-6tb1.metal, u-9tb1.metal, and u-12tb1.metal

Resources

For more information, see the following videos:

- [AWS re:Invent 2017: The Amazon EC2 Nitro System Architecture](#)
- [AWS re:Invent 2017: Amazon EC2 Bare Metal Instances](#)
- [The Nitro Project: Next-Generation EC2 Infrastructure](#)

Networking and Storage Features

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

Networking features

- IPv6 is supported on all current generation instance types and the C3, R3, and I2 previous generation instance types.
- To maximize the networking and bandwidth performance of your instance type, you can do the following:
 - Launch supported instance types into a cluster placement group to optimize your instances for high performance computing (HPC) applications. Instances in a common cluster placement group can benefit from high-bandwidth, low-latency networking. For more information, see [Placement Groups \(p. 692\)](#).
 - 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 Windows \(p. 684\)](#).
- 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 \(p. 697\)](#).

Storage features

- Some instance types support EBS volumes and instance store volumes, while other instance types support only EBS volumes. Some instance types that support instance store volumes use solid state drives (SSD) to deliver very high random I/O performance. Some instance types support NVMe instance store volumes. Some instance types support NVMe EBS volumes. For more information, see [Storage \(p. 731\)](#).
- 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 \(p. 807\)](#).

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

	EBS only	NVMe EBS	Instance store	Placement group	Enhanced networking
C4	Yes			Yes	Intel 82599 VF
C5	Yes	Yes		Yes	ENAs
C5d		Yes	NVMe *	Yes	ENAs
C5n	Yes	Yes		Yes	ENAs
D2			HDD	Yes	Intel 82599 VF
F1			NVMe *	Yes	ENAs
G3	Yes			Yes	ENAs

	EBS only	NVMe EBS	Instance store	Placement group	Enhanced networking
H1			HDD	Yes	ENA
I3			NVMe *	Yes	ENA
M4	Yes			Yes	m4.16xlarge: ENA All other sizes: Intel 82599 VF
M5	Yes	Yes		Yes	ENA
M5a	Yes	Yes		Yes	ENA
M5d		Yes	NVMe *	Yes	ENA
P2	Yes			Yes	ENA
P3	p3dn.24xlarge No	p3dn.24xlarge Yes	p3dn.24xlarge Yes NVMe *		ENA
	All other sizes: Yes	All other sizes: No			
R4	Yes			Yes	ENA
R5	Yes	Yes		Yes	ENA
R5a	Yes	Yes		Yes	ENA
R5d		Yes	NVMe *	Yes	ENA
T2	Yes				
T3	Yes	Yes			ENA
u-xtb1.metal	Yes	Yes			ENA
X1			SSD	Yes	ENA
X1e			SSD	Yes	ENA
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.

	Instance store	Placement group	Enhanced networking
C3	SSD	Yes	Intel 82599 VF
G2	SSD	Yes	
I2	SSD	Yes	Intel 82599 VF

	Instance store	Placement group	Enhanced networking
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 \(p. 899\)](#).

Burstable Performance Instances

Burstable performance instances, which are T3 and 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. Burstable performance instances are well suited for a wide range of general-purpose applications. Examples include microservices, low-latency interactive applications, small and medium databases, virtual desktops, development, build, and stage environments, code repositories, and product prototypes.

Burstable performance instances are the only instance types that use credits for CPU usage. For more information about instance pricing and additional hardware details, see [Amazon EC2 Pricing](#) and [Amazon EC2 Instance Types](#).

If your account is less than 12 months old, you can use a t2.micro instance for free within certain usage limits. For more information, see [AWS Free Tier](#).

Contents

- [Hardware Specifications \(p. 116\)](#)
- [Burstable Performance Instance Requirements \(p. 117\)](#)
- [Best Practices \(p. 117\)](#)
- [CPU Credits and Baseline Performance for Burstable Performance Instances \(p. 118\)](#)
- [Unlimited Mode for Burstable Performance Instances \(p. 120\)](#)
- [Standard Mode \(p. 125\)](#)
- [Working with Burstable Performance Instances \(p. 135\)](#)
- [Monitoring Your CPU Credits \(p. 139\)](#)

Hardware Specifications

The following is a summary of the hardware specifications for burstable performance instances.

Instance type	Default vCPUs	Memory (GiB)
t2.nano	1	0.5
t2.micro	1	1

Instance type	Default vCPUs	Memory (GiB)
t2.small	1	2
t2.medium	2	4
t2.large	2	8
t2.xlarge	4	16
t2.2xlarge	8	32
t3.nano	2	0.5
t3.micro	2	1
t3.small	2	2
t3.medium	2	4
t3.large	2	8
t3.xlarge	4	16
t3.2xlarge	8	32

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

Burstable Performance Instance Requirements

The following are the requirements for these instances:

- You must launch T3 instances using an AMI with ENA and NVMe drivers. For more information, see [Release Notes \(p. 147\)](#).
- These instances are available as On-Demand Instances, Reserved Instances, and Spot Instances, but not as Scheduled Instances or Dedicated Instances. They are also not supported on a Dedicated Host. For more information, see [Instance Purchasing Options \(p. 185\)](#).
- There is a limit on the total number of instances that you can launch in an AWS Region, and there are additional limits on some instance types. If you need more instances, request them using the [Amazon EC2 Instance Request Form](#).
- Ensure that the instance size you choose passes the minimum memory requirements of your operating system and applications. Operating systems with graphical user interfaces that consume significant memory and CPU resources (for example, Windows) might require a t2.micro or larger instance size for many use cases. As the memory and CPU requirements of your workload grow over time, you can scale to larger instance sizes of the same instance type, or another instance type.

Best Practices

Follow these best practices to get the maximum benefit from burstable performance instances.

- **Use a recommended AMI** – Use an AMI that provides the required drivers. For more information, see [Release Notes \(p. 147\)](#).
- **Turn on instance recovery** – Create a CloudWatch alarm that monitors an EC2 instance and automatically recovers it if it becomes impaired for any reason. For more information, see [Adding Recover Actions to Amazon CloudWatch Alarms \(p. 521\)](#).

CPU Credits and Baseline Performance for Burstable Performance Instances

Traditional Amazon EC2 instance types provide fixed performance, while burstable performance 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.

Contents

- [CPU Credits \(p. 118\)](#)
- [Baseline Performance \(p. 120\)](#)

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 burstable performance instance continuously earns (at a millisecond-level resolution) a set rate of CPU credits per hour, depending on the instance size. The accounting process for whether credits are accrued or spent also happens at a millisecond-level resolution, so you don't have to worry about overspending CPU credits; a short burst of CPU uses a small fraction of a CPU credit.

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

The following table lists the burstable performance instance types, the rate at which CPU credits are earned per hour, the maximum number of earned CPU credits that an instance can accrue, the number of vCPUs per instance, and the baseline 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	81.6	1958.4	8	17%**
t3.nano	6	144	2	5%**
t3.micro	12	288	2	10%**

Instance type	CPU credits earned per hour	Maximum earned credits that can be accrued*	vCPUs	Baseline performance per vCPU
t3.small	24	576	2	20%**
t3.medium	24	576	2	20%**
t3.large	36	864	2	30%**
t3.xlarge	96	2304	4	40%**
t3.2xlarge	192	4608	8	40%**

* The number of credits that can be accrued is equivalent to the number of credits that can be earned in a 24-hour period.

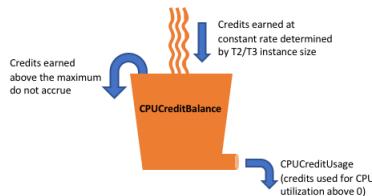
** The baseline performance in the table is per vCPU. For instance sizes that have more than one vCPU, to calculate the baseline CPU utilization for the instance, multiply the vCPU percentage by the number of vCPUs. For example, a t3.large instance has two vCPUs, which provide a baseline CPU utilization 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 t3.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 t3.nano earns six credits per hour, while a t3.small earns 24 credits per hour. The preceding table lists the credit earn rate for all instances.

CPU Credit Accrual Limit

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



The CPU credit balance limit differs for each instance size. For example, a t3.micro instance can accrue a maximum of 288 earned CPU credits in the CPU credit balance. The preceding table lists the maximum number of earned credits that each instance can accrue.

Note

T2 Standard instances also earn launch credits. 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 \(p. 125\)](#).

T3 instances do not earn launch credits. These instances launch as `unlimited` by default, and therefore can burst immediately upon start without any launch credits.

Accrued CPU Credits Life Span

CPU credits on a running instance do not expire.

For T3, the CPU credit balance persists for seven days after an instance stops and the credits are lost thereafter. If you start the instance within seven days, no credits are lost.

For T2, the CPU credit balance does not persist between instance stops and starts. If you stop a T2 instance, the instance loses all its accrued credits.

For more information, see `CPUCreditBalance` in the [CloudWatch metrics table \(p. 139\)](#).

Baseline Performance

The number of credits that an instance earns per hour can be expressed as a percentage of CPU utilization. It is known as the *baseline performance*, and sometimes just as *the baseline*. For example, a `t3.nano` instance, with two vCPUs, earns six credits per hour, resulting in a baseline performance of 5% (3/60 minutes) per vCPU. A `t3.xlarge` instance, with four vCPUs, earns 96 credits per hour, resulting in a baseline performance of 40% (24/60 minutes) per vCPU.

Unlimited Mode for Burstable Performance Instances

A burstable performance instance configured as `unlimited` can sustain high CPU performance for any period of time whenever required. The hourly 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 section for Unlimited pricing 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.

Contents

- [Unlimited Mode Concepts \(p. 120\)](#)
- [Examples: Unlimited Mode \(p. 122\)](#)

Unlimited Mode Concepts

The `unlimited` mode is a credit configuration option for burstable performance instances. It can be enabled or disabled at any time for a running or stopped instance.

- T3 instances are launched as `unlimited` by default.
- T2 instances are launched as `standard` by default.

A burstable performance instance configured as `unlimited` can burst above the baseline for as long as required. This enables you to enjoy the low 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 hourly instance price automatically covers all CPU usage spikes if the average CPU utilization of an instance configured as `unlimited` over a rolling 24-hour period is at or below the baseline. For the vast majority of general-purpose workloads, instances configured as `unlimited` 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.

How Unlimited Burstable Performance Instances Work

If a burstable performance instance configured as `unlimited` depletes its CPU credit balance, it can spend *surplus* credits to burst beyond the baseline. When its CPU utilization falls below the baseline, it uses the CPU credits that it earns to pay down the surplus credits that it spent earlier. The ability to earn CPU credits to pay down surplus credits enables Amazon EC2 to average the CPU utilization of an instance over a 24-hour period.

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 \(p. 118\)](#) in a 24-hour period (for example, a `t3.micro` instance can earn a maximum of 288 credits in a 24-hour period), it can spend surplus credits up to that maximum without being charged.

However, if CPU utilization stays above the baseline, the instance cannot earn enough credits to pay down the surplus credits that it has spent. The surplus credits that are not paid down are charged at a flat additional rate per vCPU-hour.

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

- The spent surplus credits exceed the [maximum number of credits \(p. 118\)](#) the instance can earn in a 24-hour period. Spent surplus credits above the maximum are charged at the end of the hour.
- The instance is stopped or terminated.
- The instance is switched from `unlimited` to `standard`.

Spent surplus credits are tracked by the CloudWatch metric `CPUSurplusCreditBalance`. Surplus credits that are charged are tracked by the CloudWatch metric `CPUSurplusCreditsCharged`. For more information, see [Additional CloudWatch Metrics for Burstable Performance Instances \(p. 139\)](#).

No Launch Credits for T2 Unlimited

T2 Standard instances receive [launch credits \(p. 125\)](#), but T2 Unlimited instances do not. A T2 Unlimited instance can burst beyond the baseline at any time with no additional charge, as long as its average CPU utilization is at or below the baseline over a rolling 24-hour window or its lifetime, whichever is shorter. As such, T2 Unlimited instances do not require launch credits to achieve high performance immediately after launch.

If a T2 instance is switched from `standard` to `unlimited`, any accrued launch credits are removed from the `CPUCreditBalance` before the remaining `CPUCreditBalance` is carried over.

Note

T3 instances never receive launch credits.

Enabling Unlimited Mode

T3 instances launch as `unlimited` by default. T2 instances launch as `standard` by default, but you can enable `unlimited` at launch.

You can switch from `unlimited` to `standard`, and from `standard` to `unlimited`, at any time on a running or stopped instance. For more information, see [Launching a Burstable Performance Instance as Unlimited or Standard \(p. 136\)](#) and [Modifying the Credit Specification of a Burstable Performance Instance \(p. 138\)](#).

You can check whether your burstable performance instance is configured as `unlimited` or `standard` using the Amazon EC2 console or the AWS CLI. For more information, see [Viewing the Credit Specification of a Burstable Performance Instance \(p. 138\)](#).

What Happens to Credits when Switching between Unlimited and Standard

`CPUCreditBalance` is a CloudWatch metric that tracks the number of credits accrued by an instance. `CPUSurplusCreditBalance` is a CloudWatch metric that tracks the number of surplus credits spent by an instance.

When you change an instance configured as `unlimited` to `standard`, the following occurs:

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

When a `standard` instance is switched to `unlimited`, the following occurs:

- The `CPUCreditBalance` value containing accrued earned credits is carried over.
- For T2 Standard instances, any launch credits are removed from the `CPUCreditBalance` value, and the remaining `CPUCreditBalance` value containing accrued earned credits is carried over.

Monitoring Credit Usage

To see if your instance is spending more credits than the baseline provides, you can use CloudWatch metrics to track and set up hourly alarms to be notified of credit usage. For more information, see [Monitoring Your CPU Credits \(p. 139\)](#).

Examples: Unlimited Mode

The following examples explain credit use for instances that are configured as `unlimited`.

Examples

- [Example 1: Explaining Credit Use with T3 Unlimited \(p. 122\)](#)
- [Example 2: Explaining Credit Use with T2 Unlimited \(p. 123\)](#)

Example 1: Explaining Credit Use with T3 Unlimited

In this example, you see the CPU utilization of a `t3.nano` instance launched as `unlimited`, and how it spends *earned* and *surplus* credits to sustain CPU performance.

A `t3.nano` instance earns 144 CPU credits over a rolling 24-hour period, which it can redeem for 144 minutes of vCPU use. When it depletes its CPU credit balance (represented by the CloudWatch metric `CPUCreditBalance`), it can spend *surplus* CPU credits—that it has *not yet earned*—to burst for as long as it needs. Because a `t3.nano` instance earns a maximum of 144 credits in a 24-hour period, it can spend surplus credits up to that maximum without being charged immediately. If it spends more than 144 CPU credits, it is charged for the difference at the end of the hour.

The intent of the example, illustrated by the following graph, is to show how an instance can burst using surplus credits even after it depletes its `CPUCreditBalance`. The following workflow references the numbered points on the graph:

P1 – At 0 hours on the graph, the instance is launched as `unlimited` and immediately begins to earn credits. The instance remains idle from the time it is launched—CPU utilization is 0%—and no credits are spent. All unspent credits are accrued in the credit balance. For the first 24 hours, `CPUCreditUsage` is at 0, and the `CPUCreditBalance` value reaches its maximum of 144.

P2 – For the next 12 hours, CPU utilization is at 2.5%, which is below the 5% baseline. The instance earns more credits than it spends, but the `CPUCreditBalance` value cannot exceed its maximum of 144 credits.

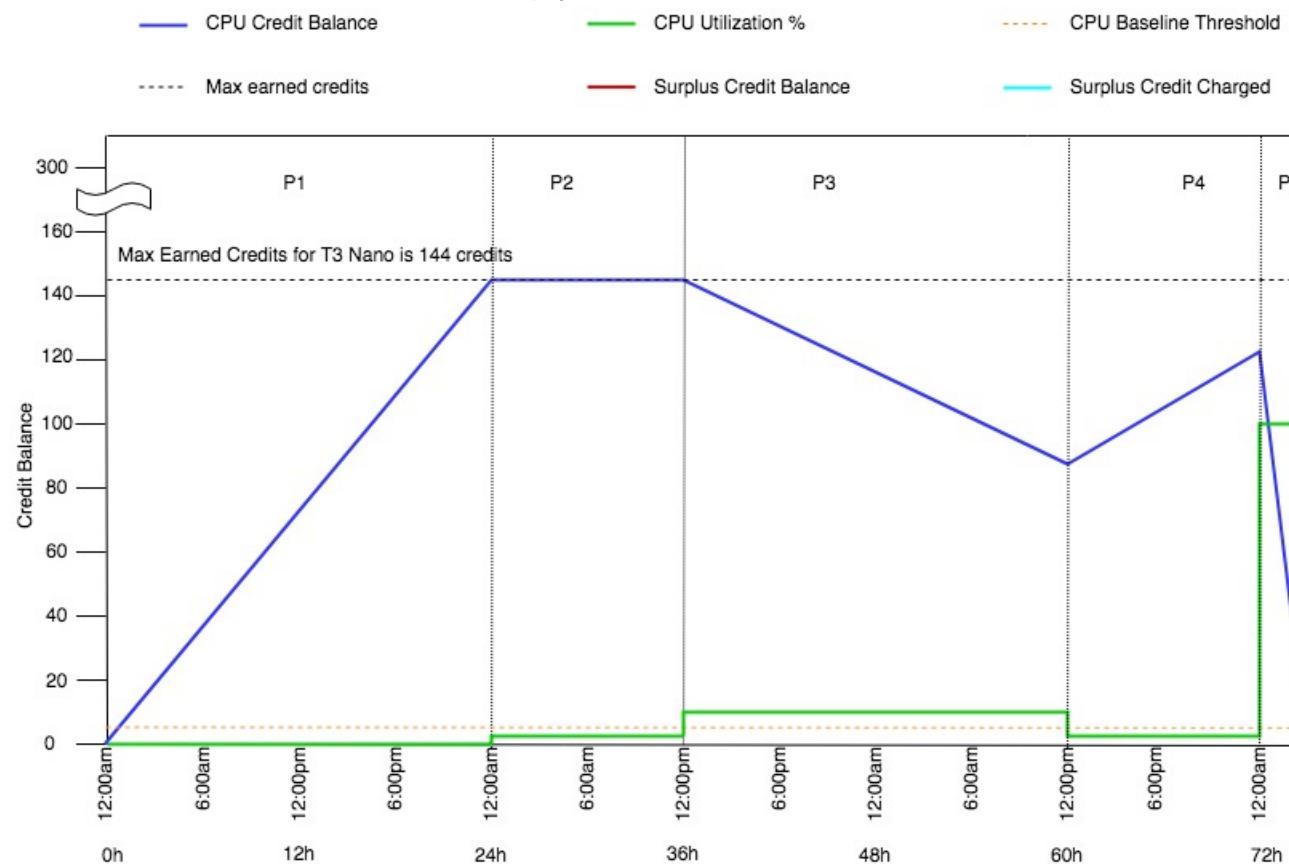
P3 – For the next 24 hours, CPU utilization is at 7% (above the baseline), which requires a spend of 57.6 credits. The instance spends more credits than it earns, and the `CPUCreditBalance` value reduces to 86.4 credits.

P4 – For the next 12 hours, CPU utilization decreases to 2.5% (below the baseline), which requires a spend of 36 credits. In the same time, the instance earns 72 credits. The instance earns more credits than it spends, and the `CPUCreditBalance` value increases to 122 credits.

P5 – For the next 5 hours, the instance bursts at 100% CPU utilization, and spends a total of 570 credits to sustain the burst. About an hour into this period, the instance depletes its entire `CPUCreditBalance` of 122 credits, and starts to spend surplus credits to sustain the high CPU performance, totaling 448 surplus credits in this period ($570 - 122 = 448$). When the `CPUSurplusCreditBalance` value reaches 144 CPU credits (the maximum a `t3.nano` instance can earn in a 24-hour period), any surplus credits spent thereafter cannot be offset by earned credits. The surplus credits spent thereafter amounts to 304 credits ($448 - 144 = 304$), which results in a small additional charge at the end of the hour for 304 credits.

P6 – For the next 13 hours, CPU utilization is at 5% (the baseline). The instance earns as many credits as it spends, with no excess to pay down the `CPUSurplusCreditBalance`. The `CPUSurplusCreditBalance` value remains at 144 credits.

P7 – For the last 24 hours in this example, the instance is idle and CPU utilization is 0%. During this time, the instance earns 144 credits, which it uses to pay down the `CPUSurplusCreditBalance`.



Example 2: Explaining Credit Use with T2 Unlimited

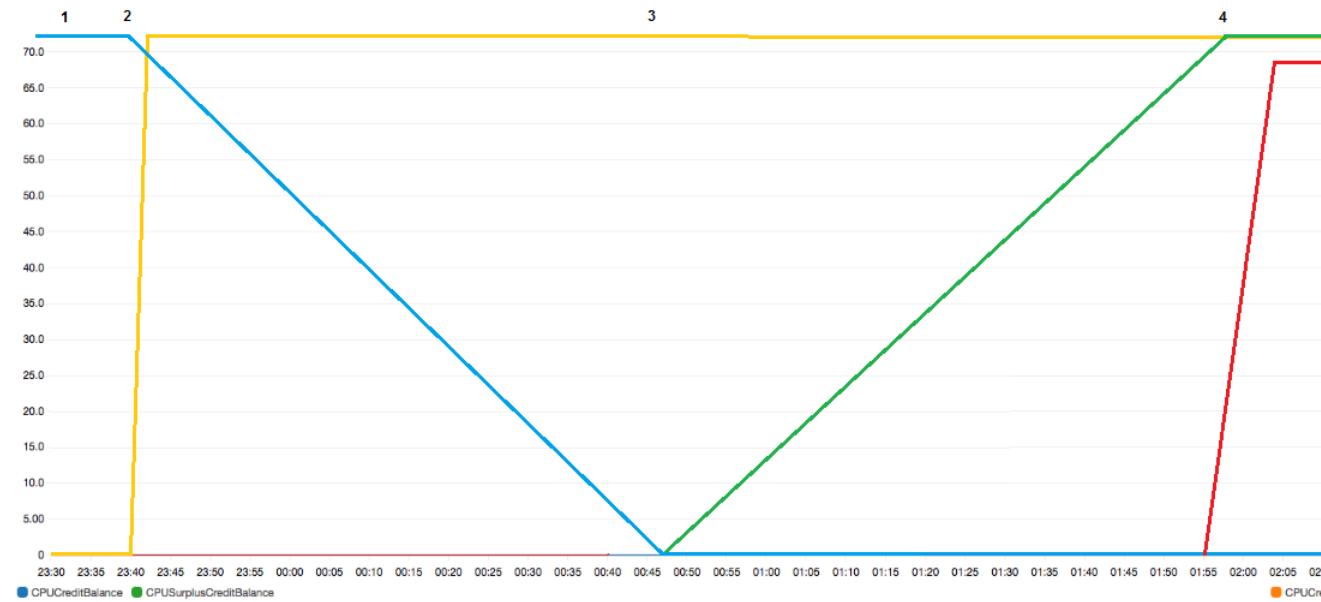
In this example, you see the CPU utilization of a `t2.nano` instance launched as `unlimited`, and how it spends *earned* and *surplus* credits to sustain CPU performance.

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` instance earns a maximum of 72 credits in a 24-hour period, it can spend

surplus credits up to that maximum without being charged immediately. If it spends more than 72 CPU credits, it is charged for the difference at the end of the hour.

The intent of the example, illustrated by the following graph, is to show how an instance can burst using surplus credits even after it depletes its `CPUCreditBalance`. You can assume that, at the start of the time line in the graph, the instance has an accrued credit balance equal to the maximum number of credits it can earn in 24 hours. The following workflow references the numbered points on the graph:

- 1 – In the first 10 minutes, `CPUCreditUsage` is at 0, and the `CPUCreditBalance` value remains at its maximum of 72.
- 2 – At 23:40, as CPU utilization increases, the instance spends CPU credits and the `CPUCreditBalance` value decreases.
- 3 – At around 00:47, the instance depletes its entire `CPUCreditBalance`, and starts to spend surplus credits to sustain high CPU performance.
- 4 – Surplus credits are spent until 01:55, when the `CPUSurplusCreditBalance` value reaches 72 CPU credits. This is equal to the maximum a t2.nano instance can earn in a 24-hour period. Any surplus credits spent thereafter cannot be offset by earned credits within the 24-hour period, which results in a small additional charge at the end of the hour.
- 5 – The instance continues to spend surplus credits until around 02:20. At this time, CPU utilization falls below the baseline, and the instance starts to earn credits at 3 credits per hour (or 0.25 credits every 5 minutes), which it uses to pay down the `CPUSurplusCreditBalance`. After the `CPUSurplusCreditBalance` value reduces to 0, the instance starts to accrue earned credits in its `CPUCreditBalance` at 0.25 credits every 5 minutes.



All metrics		Graphed metrics (4)	Graph options	
Label	Details	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 \$0.096 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 \$0.096/\text{vCPU-hour} = \0.04032 , rounded to \$0.04.

Here is the month-end bill for this T2 Unlimited instance:

Amazon Elastic Compute Cloud running Windows		
\$0.0081 per On Demand Windows t2.nano Instance Hour	720.000 Hrs	\$5.83
Amazon Elastic Compute Cloud T2 CPU Credits		
\$0.096 per vCPU-Hour of T2 CPU credits	0.420 vCPU-Hours	\$0.04

You can set billing alerts to be notified every hour of any accruing charges, and take action if required.

Standard Mode

A burstable performance instance configured as **standard** 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 for Burstable Performance Instances \(p. 118\)](#).

Contents

- [Standard Mode Concepts \(p. 125\)](#)
- [Examples: Standard Mode \(p. 127\)](#)

Standard Mode Concepts

The **standard** mode is a configuration option for burstable performance instances. It can be enabled or disabled at any time for a running or stopped instance.

Note

T3 instances are launched as **unlimited** by default. T2 instances are launched as **standard** by default.

How Standard Burstable Performance Instances Work

When a burstable performance instance configured as **standard** is in a running state, it continuously earns (at a millisecond-level resolution) a set rate of earned credits per hour. For T2 Standard, when the instance is stopped, it loses all its accrued credits, and its credit balance is reset to zero. When it is restarted, it receives a new set of launch credits, and begins to accrue earned credits. For T3 Standard, the CPU credit balance persists for seven days after the instance stops and the credits are lost thereafter. If you start the instance within seven days, no credits are lost.

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.

T3 Standard instances do not receive launch credits.

Launch Credits

T2 Standard instances get 30 launch credits per vCPU at launch or start. For example, a `t2.micro` instance has one vCPU and gets 30 launch credits, while a `t2.xlarge` instance has four vCPUs and gets

120 launch credits. Launch credits are designed to provide a good startup experience to allow instances to burst immediately after launch before they have accrued earned credits.

Launch credits are spent first, before earned credits. Unspent launch credits are accrued in the CPU credit balance, but do not count towards the CPU credit balance limit. For example, a `t2.micro` instance has a CPU credit balance limit of 144 earned credits. If it is launched and remains idle for 24 hours, its CPU credit balance reaches 174 (30 launch credits + 144 earned credits), which is over the limit. However, after the instance spends the 30 launch credits, the credit balance cannot exceed 144. For more information about the CPU credit balance limit for each instance size, see the [credit table \(p. 118\)](#).

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.

Tip

To ensure that your workloads always get the performance they need, switch to [Unlimited Mode for Burstable Performance Instances \(p. 120\)](#) or consider using a larger instance size.

Differences Between Launch Credits and Earned Credits

The following table lists the differences between launch credits and earned credits.

	Launch credits	Earned credits
Credit earn rate	T2 Standard instances get 30 launch credits per vCPU at launch or start. If a T2 instance is switched from unlimited to standard, it does not get launch credits at the time of switching.	Each T2 instance continuously earns (at a millisecond-level resolution) a set rate of CPU credits per hour, depending on the instance size. For more information about the number of CPU credits earned per instance size, see the credit table (p. 118) .
Credit earn limit	The limit for receiving launch credits is 100 launches or starts of all T2 Standard	A T2 instance cannot accrue more credits than the CPU credit balance limit. If the

	Launch credits	Earned credits
	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.	CPU credit balance has reached its limit, any credits that are earned after the limit is reached are discarded. Launch credits do not count towards the limit. For more information about the CPU credit balance limit for each T2 instance size, see the credit table (p. 118) .
Credit use	Launch credits are spent first, before earned credits.	Earned credits are spent only after all launch credits are spent.
Credit expiration	When a T2 Standard instance is running, launch credits do not expire. When a T2 Standard instance stops or is switched to T2 Unlimited, all launch credits are lost.	When a T2 instance is running, earned credits that have accrued do not expire. When the T2 instance stops, all accrued earned credits are lost.

The number of accrued launch credits and accrued earned credits is tracked by the CloudWatch metric `CPUCreditBalance`. For more information, see `CPUCreditBalance` in the [CloudWatch metrics table \(p. 139\)](#).

Examples: Standard Mode

The following examples explain credit use when instances are configured as standard.

Examples

- [Example 1: Explaining Credit Use with T3 Standard \(p. 127\)](#)
- [Example 2: Explaining Credit Use with T2 Standard \(p. 128\)](#)

Example 1: Explaining Credit Use with T3 Standard

In this example, you see how a `t3.nano` instance launched as standard earns, accrues, and spends *earned* credits. You see how the credit balance reflects the accrued *earned* credits.

Note

`T3` instances configured as standard do not receive launch credits.

A running `t3.nano` instance earns 144 credits every 24 hours. Its credit balance limit is 144 earned credits. After the limit is reached, new credits that are earned are discarded. For more information about the number of credits that can be earned and accrued, see the [credit table \(p. 118\)](#).

You might launch a `T3 Standard` instance and use it immediately. Or, you might launch a `T3 Standard` instance and leave it idle for a few days before running applications on it. Whether an instance is used or remains idle determines if credits are spent or accrued. If an instance remains idle for 24 hours from the time it is launched, the credit balance reaches its limit, which is the maximum number of earned credits that can be accrued.

This example describes an instance that remains idle for 24 hours from the time it is launched, and walks you through seven periods of time over a 96-hour period, showing the rate at which credits are earned, accrued, spent, and discarded, and the value of the credit balance at the end of each period.

The following workflow references the numbered points on the graph:

P1 – At 0 hours on the graph, the instance is launched as standard and immediately begins to earn credits. The instance remains idle from the time it is launched—CPU utilization is 0%—and no credits are

spent. All unspent credits are accrued in the credit balance. For the first 24 hours, `CPUCreditUsage` is at 0, and the `CPUCreditBalance` value reaches its maximum of 144.

P2 – For the next 12 hours, CPU utilization is at 2.5%, which is below the 5% baseline. The instance earns more credits than it spends, but the `CPUCreditBalance` value cannot exceed its maximum of 144 credits. Any credits that are earned in excess of the limit are discarded.

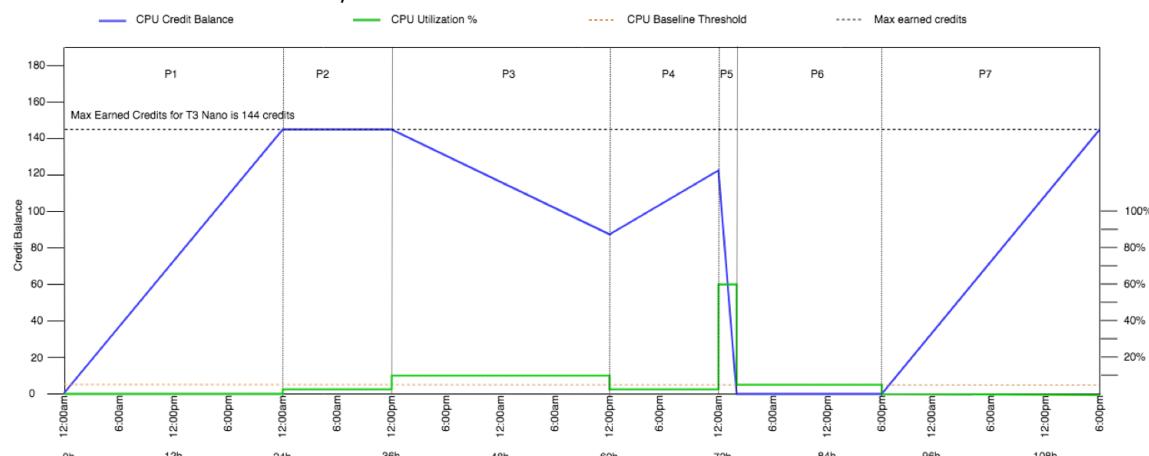
P3 – For the next 24 hours, CPU utilization is at 7% (above the baseline), which requires a spend of 57.6 credits. The instance spends more credits than it earns, and the `CPUCreditBalance` value reduces to 86.4 credits.

P4 – For the next 12 hours, CPU utilization decreases to 2.5% (below the baseline), which requires a spend of 36 credits. In the same time, the instance earns 72 credits. The instance earns more credits than it spends, and the `CPUCreditBalance` value increases to 122 credits.

P5 – For the next two hours, the instance bursts at 100% CPU utilization, and depletes its entire `CPUCreditBalance` value of 122 credits. At the end of this period, with the `CPUCreditBalance` at zero, CPU utilization is forced to drop to the baseline performance level of 5%. At the baseline, the instance earns as many credits as it spends.

P6 – For the next 14 hours, CPU utilization is at 5% (the baseline). The instance earns as many credits as it spends. The `CPUCreditBalance` value remains at 0.

P7 – For the last 24 hours in this example, the instance is idle and CPU utilization is 0%. During this time, the instance earns 144 credits, which it accrues in its `CPUCreditBalance`.



Example 2: Explaining Credit Use with T2 Standard

In this example, you see how a `t2.nano` instance launched as standard earns, accrues, and spends *launch* and *earned* credits. You see how the credit balance reflects not only accrued *earned* credits, but also accrued *launch* credits.

A `t2.nano` instance gets 30 launch credits when it is launched, and earns 72 credits every 24 hours. Its credit balance limit is 72 earned credits; launch credits do not count towards the limit. After the limit is reached, new credits that are earned are discarded. For more information about the number of credits that can be earned and accrued, see the [credit table \(p. 118\)](#). For more information about limits, see [Launch Credit Limits \(p. 126\)](#).

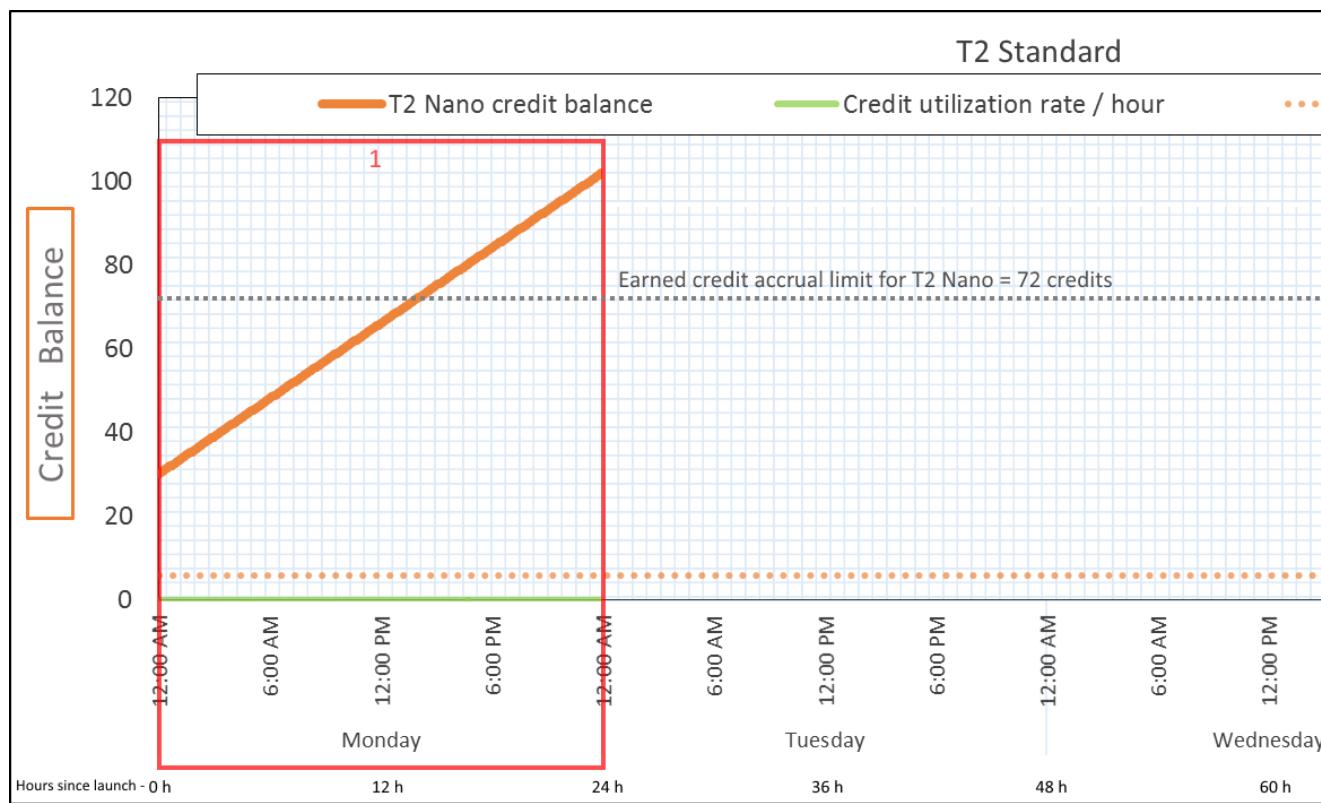
You might launch a T2 Standard instance and use it immediately. Or, you might launch a T2 Standard instance and leave it idle for a few days before running applications on it. Whether an instance is used or remains idle determines if credits are spent or accrued. If an instance remains idle for 24 hours from the time it is launched, the credit balance appears to exceed its limit because the balance reflects both accrued earned credits and accrued launch credits. However, after CPU is used, the launch credits are

spent first. Thereafter, the limit always reflects the maximum number of earned credits that can be accrued.

This example describes an instance that remains idle for 24 hours from the time it is launched, and walks you through seven periods of time over a 96-hour period, showing the rate at which credits are earned, accrued, spent, and discarded, and the value of the credit balance at the end of each period.

Period 1: 1 – 24 hours

At 0 hours on the graph, the T2 instance is launched as standard and immediately gets 30 launch credits. It earns credits while in the running state. The instance remains idle from the time it is launched—CPU utilization is 0%—and no credits are spent. All unspent credits are accrued in the credit balance. At approximately 14 hours after launch, the credit balance is 72 (30 launch credits + 42 earned credits), which is equivalent to what the instance can earn in 24 hours. At 24 hours after launch, the credit balance exceeds 72 credits because the unspent launch credits are accrued in the credit balance—the credit balance is 102 credits: 30 launch credits + 72 earned credits.



Credit Spend Rate	0 credits per 24 hours (0% CPU utilization)
Credit Earn Rate	72 credits per 24 hours
Credit Discard Rate	0 credits per 24 hours
Credit Balance	102 credits (30 launch credits + 72 earned credits)

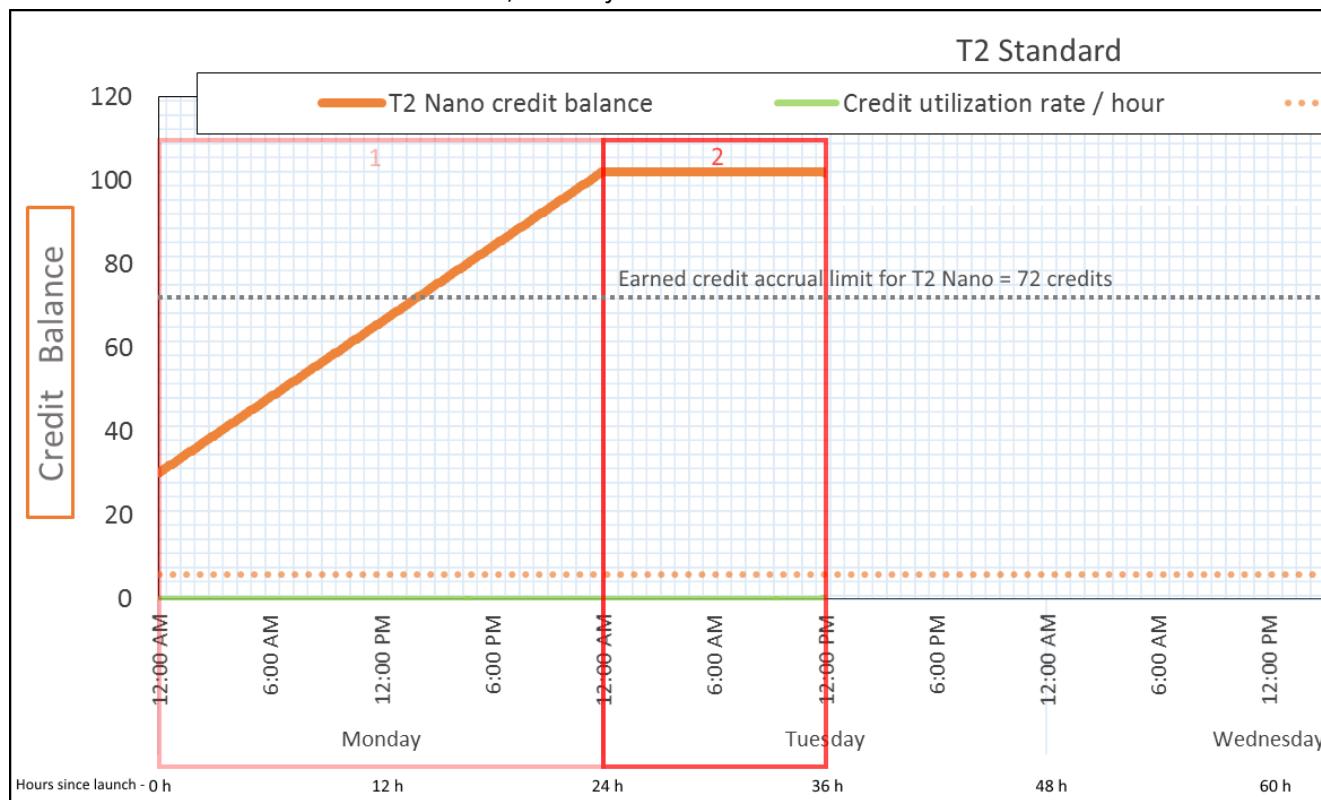
Conclusion

If there is no CPU utilization after launch, the instance accrues more credits than what it can earn in 24 hours (30 launch credits + 72 earned credits = 102 credits).

In a real-world scenario, an EC2 instance consumes a small number of credits while launching and running, which prevents the balance from reaching the maximum theoretical value in this example.

Period 2: 25 – 36 hours

For the next 12 hours, the instance continues to remain idle and earn credits, but the credit balance does not increase. It plateaus at 102 credits (30 launch credits + 72 earned credits). The credit balance has reached its limit of 72 accrued earned credits, so newly earned credits are discarded.



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

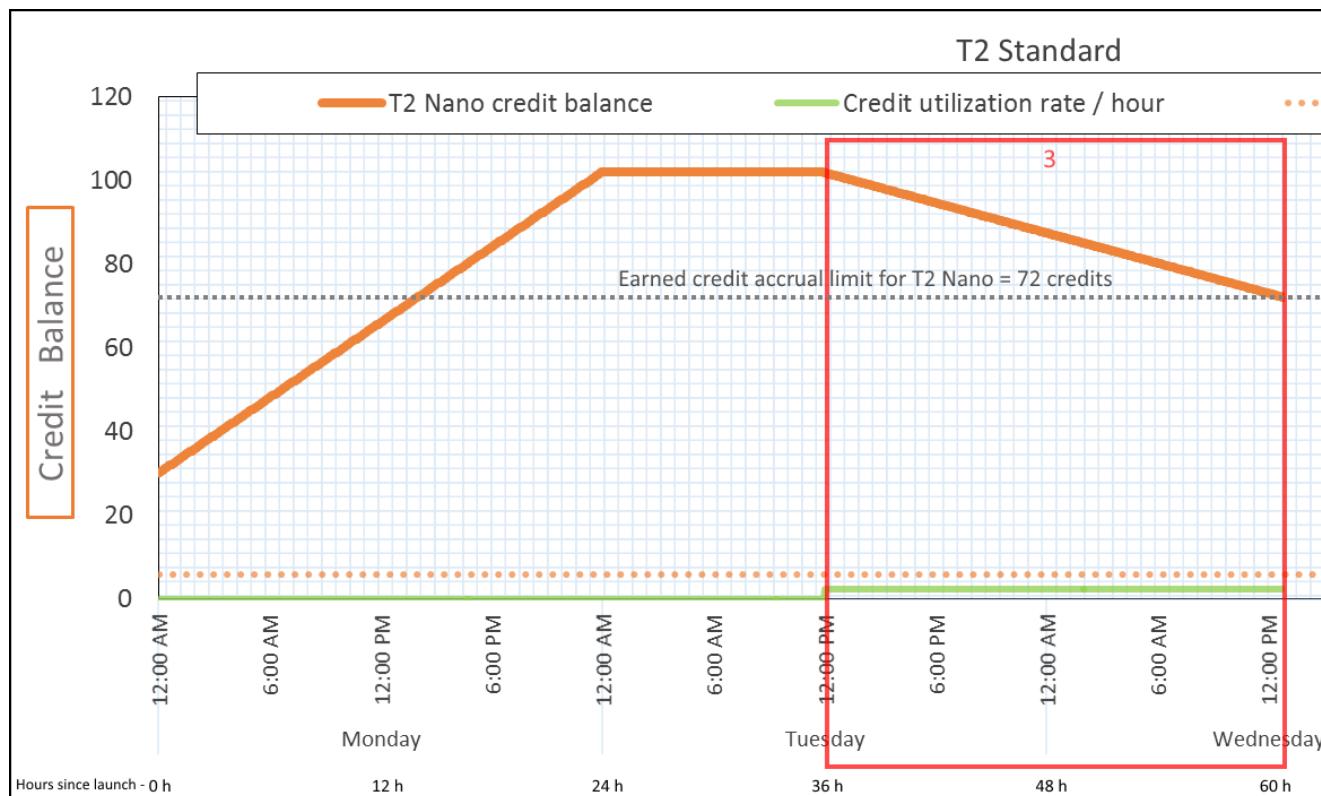
Conclusion

An instance constantly earns credits, but it cannot accrue more earned credits if the credit balance has reached its limit. After the limit is reached, newly earned credits are discarded. Launch credits do not count towards the credit balance limit. If the balance includes accrued launch credits, the balance appears to be over the limit.

Period 3: 37 – 61 hours

For the next 25 hours, the instance uses 2% CPU, which requires 30 credits. In the same period, it earns 75 credits, but the credit balance decreases. The balance decreases because the accrued *launch* credits

are spent first, while newly earned credits are discarded because the credit balance is already at its limit of 72 earned credits.



Credit Spend Rate	28.8 credits per 24 hours (1.2 credits per hour, 2% CPU utilization, 40% of credit earn rate)—30 credits over 25 hours
Credit Earn Rate	72 credits per 24 hours
Credit Discard Rate	72 credits per 24 hours (100% of credit earn rate)
Credit Balance	72 credits (30 launch credits were spent; 72 earned credits remain unspent)

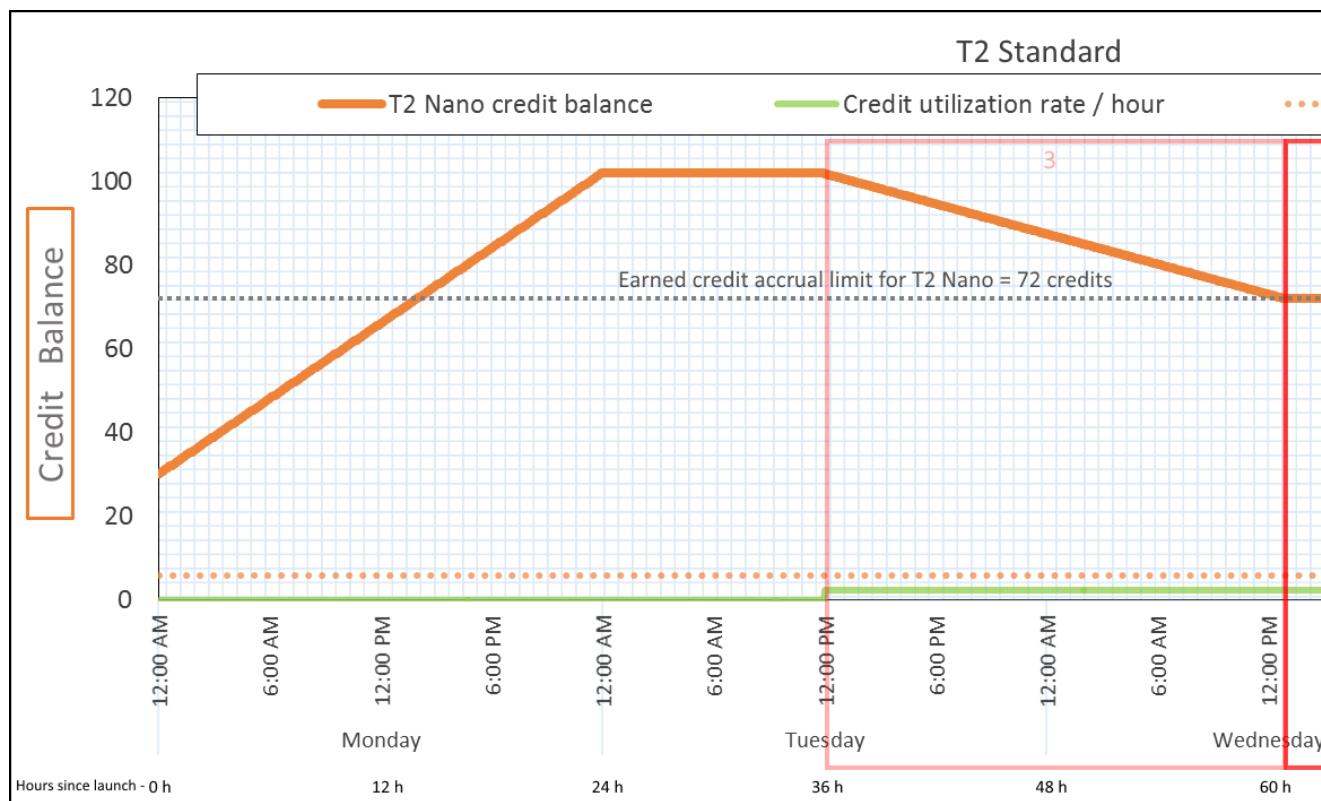
Conclusion

An instance spends launch credits first, before spending earned credits. Launch credits do not count towards the credit limit. After the launch credits are spent, the balance can never go higher than what can be earned in 24 hours. Furthermore, while an instance is running, it cannot get more launch credits.

Period 4: 62 – 72 hours

For the next 11 hours, the instance uses 2% CPU, which requires 13.2 credits. This is the same CPU utilization as in the previous period, but the balance does not decrease. It stays at 72 credits.

The balance does not decrease because the credit earn rate is higher than the credit spend rate. In the time that the instance spends 13.2 credits, it also earns 33 credits. However, the balance limit is 72 credits, so any earned credits that exceed the limit are discarded. The balance plateaus at 72 credits, which is different from the plateau of 102 credits during Period 2, because there are no accrued launch credits.



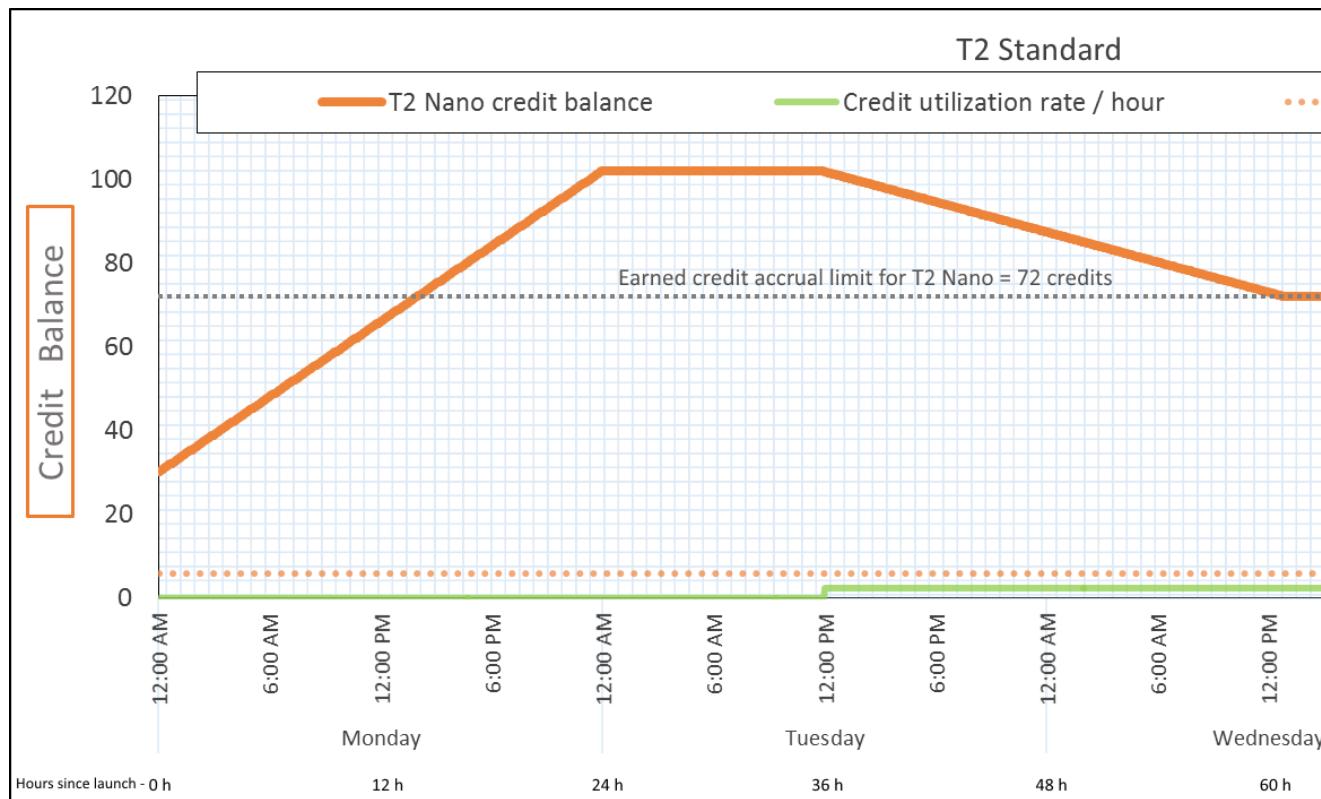
Credit Spend Rate	28.8 credits per 24 hours (1.2 credits per hour, 2% CPU utilization, 40% of credit earn rate)—13.2 credits over 11 hours
Credit Earn Rate	72 credits per 24 hours
Credit Discard Rate	43.2 credits per 24 hours (60% of credit earn rate)
Credit Balance	72 credits (0 launch credits, 72 earned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

Conclusion

After launch credits are spent, the credit balance limit is determined by the number of credits that an instance can earn in 24 hours. If the instance earns more credits than it spends, newly earned credits over the limit are discarded.

Period 5: 73 – 75 hours

For the next three hours, the instance bursts at 20% CPU utilization, which requires 36 credits. The instance earns nine credits in the same three hours, which results in a net balance decrease of 27 credits. At the end of three hours, the credit balance is 45 accrued earned credits.



Credit Spend Rate	288 credits per 24 hours (12 credits per hour, 20% CPU utilization, 400% of credit earn rate)—36 credits over 3 hours
Credit Earn Rate	72 credits per 24 hours (9 credits over 3 hours)
Credit Discard Rate	0 credits per 24 hours
Credit Balance	45 credits (previous balance (72) - spent credits (36) + earned credits (9))—balance decreases at a rate of 216 credits per 24 hours (spend rate 288/24 + earn rate 72/24 = balance decrease rate 216/24)

Conclusion

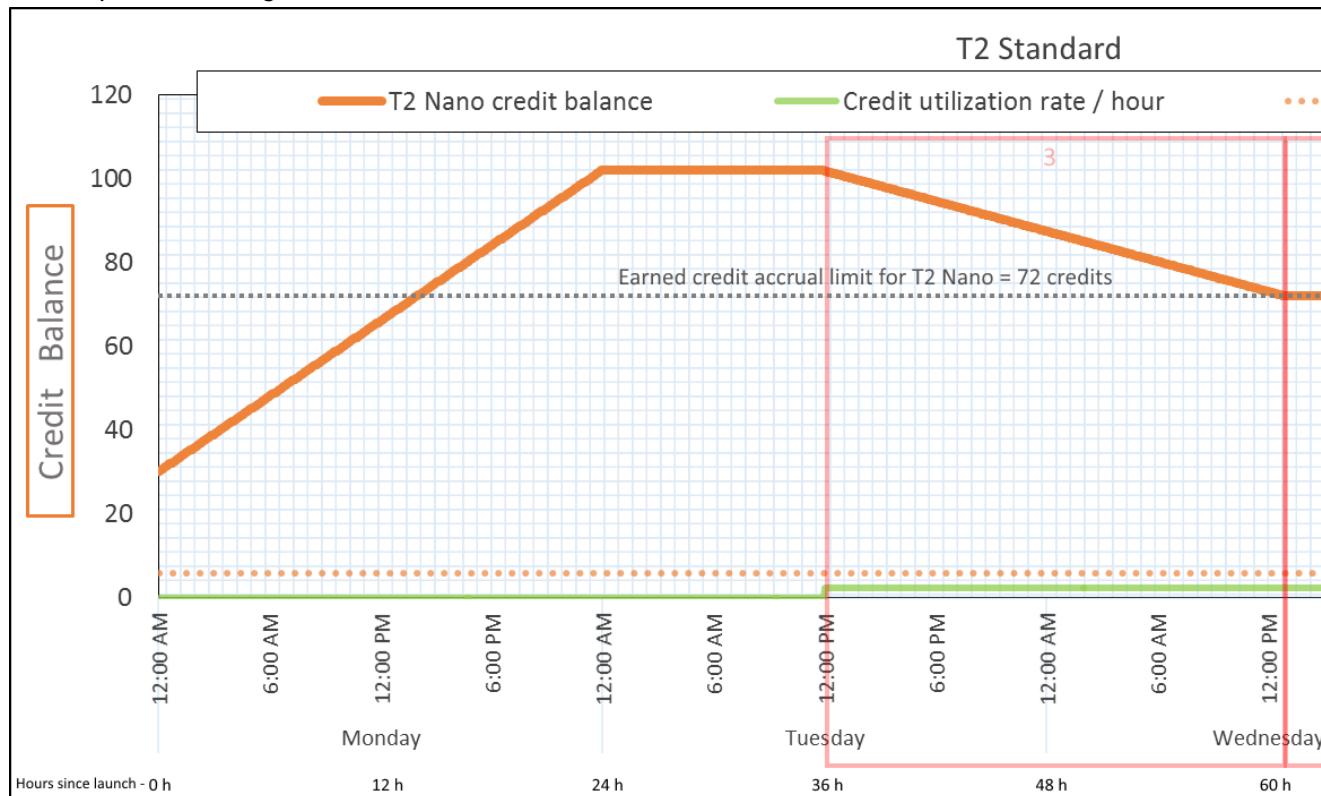
If an instance spends more credits than it earns, its credit balance decreases.

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. Any earned credits that exceeded the limit were discarded, so the balance plateaued at its maximum of 72 credits.

In this period, there are no accrued launch credits, and the number of accrued earned credits in the balance is below the limit. No earned credits are discarded. Furthermore, the instance earns more credits than it spends, resulting in an increase in the credit balance.



Credit Spend Rate	28.8 credits per 24 hours (1.2 credits per hour, 2% CPU utilization, 40% of credit earn rate)—18 credits over 15 hours
Credit Earn Rate	72 credits per 24 hours (45 credits over 15 hours)
Credit Discard Rate	0 credits per 24 hours
Credit Balance	72 credits (balance increases at a rate of 43.2 credits per 24 hours—change rate = spend rate 28.8/24 + earn rate 72/24)

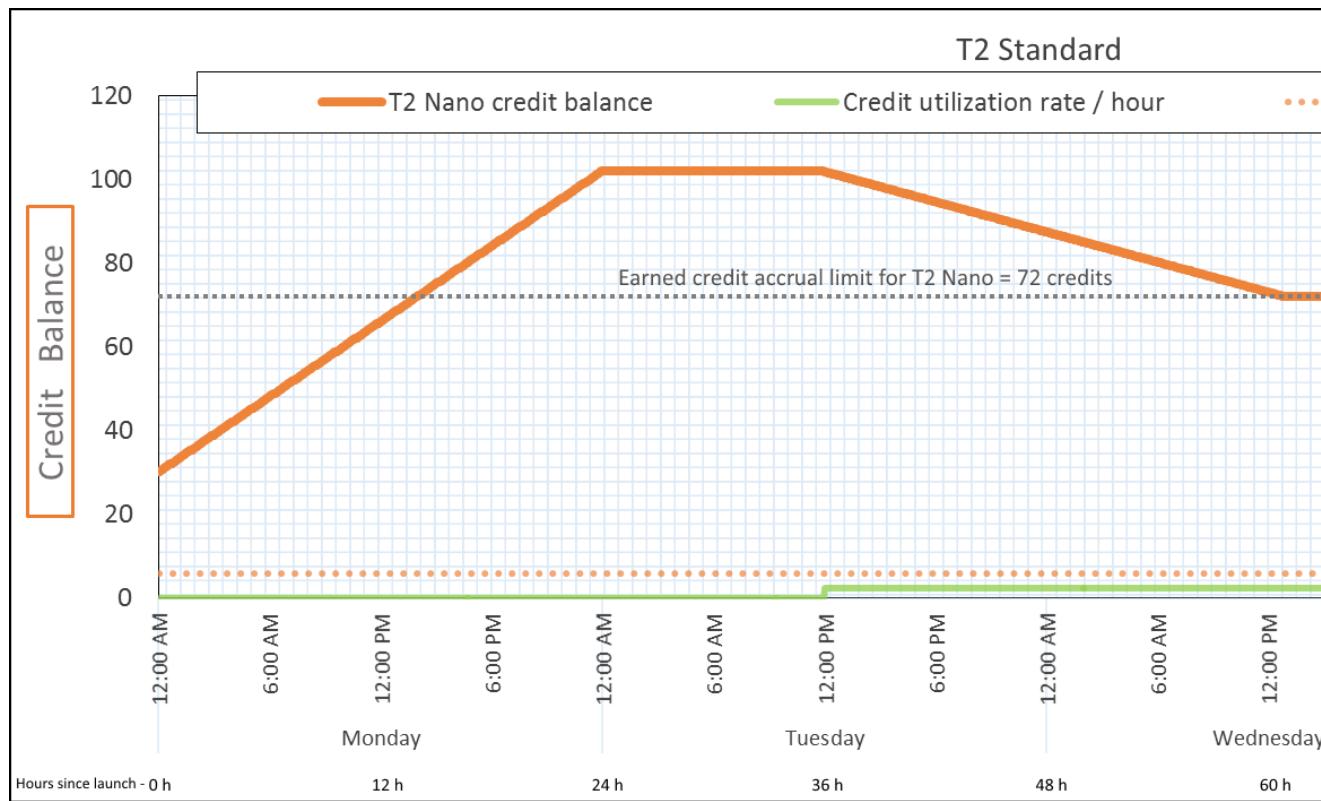
Conclusion

If an instance spends fewer credits than it earns, its credit balance increases.

Period 7: 91 – 96 hours

For the next six hours, the instance remains idle—CPU utilization is 0%—and no credits are spent. This is the same CPU utilization as in Period 2, but the balance does not plateau at 102 credits—it plateaus at 72 credits, which is the credit balance limit for the instance.

In Period 2, the credit balance included 30 accrued launch credits. The launch credits were spent in Period 3. A running instance cannot get more launch credits. After its credit balance limit is reached, any earned credits that exceed the limit are discarded.



Credit Spend Rate	0 credits per 24 hours (0% CPU utilization)
Credit Earn Rate	72 credits per 24 hours
Credit Discard Rate	72 credits per 24 hours (100% of credit earn rate)
Credit Balance	72 credits (0 launch credits, 72 earned credits)

Conclusion

An instance constantly earns credits, but cannot accrue more earned credits if the credit balance limit has been reached. After the limit is reached, newly earned credits are discarded. The credit balance limit is determined by the number of credits that an instance can earn in 24 hours. For more information about credit balance limits, see the [credit table \(p. 118\)](#).

Working with Burstable Performance Instances

The steps for launching, monitoring, and modifying these instances are similar. The key difference is the default credit specification when they launch:

- T3 instances launch as **unlimited** by default.
- T2 instances launch as **standard** by default.

Contents

- [Launching a Burstable Performance Instance as Unlimited or Standard \(p. 136\)](#)
- [Using an Auto Scaling Group to Launch a Burstable Performance Instance as Unlimited \(p. 136\)](#)

- [Viewing the Credit Specification of a Burstable Performance Instance \(p. 138\)](#)
- [Modifying the Credit Specification of a Burstable Performance Instance \(p. 138\)](#)

Launching a Burstable Performance Instance as Unlimited or Standard

T3 instances launch as **unlimited** by default. T2 instances launch as **standard** by default.

You can launch your instances as **unlimited** or **standard** using the Amazon EC2 console, an AWS SDK, a command line tool, or with an Auto Scaling group. For more information, see [Using an Auto Scaling Group to Launch a Burstable Performance Instance as Unlimited \(p. 136\)](#).

To launch a burstable performance instance as Unlimited or Standard (console)

1. Follow the [Launching an Instance Using the Launch Instance Wizard \(p. 315\)](#) procedure.
2. On the **Choose an Instance Type** page, select an instance type, and choose **Next: Configure Instance Details**.
3. Choose a credit specification. The default for T3 is **unlimited**, and for T2 it is **standard**.
 - a. To launch a T3 instance as **standard**, on the **Configure Instance Details** page, for **T2/T3 Unlimited**, clear **Enable**.
 - b. To launch a T2 instance as **unlimited**, on the **Configure Instance Details** page, for **T2/T3 Unlimited**, select **Enable**.
4. Continue as prompted by the wizard. When you've finished reviewing your options on the **Review Instance Launch** page, choose **Launch**. For more information, see [Launching an Instance Using the Launch Instance Wizard \(p. 315\)](#).

To launch a burstable performance instance as Unlimited or Standard (AWS CLI)

Use the `run-instances` command to launch your instances. Specify the credit specification using the `--credit-specification CpuCredits=` parameter. Valid credit specifications are **unlimited** and **standard**.

- For T3, if you do not include the `--credit-specification` parameter, the instance launches as **unlimited** by default.
- For T2, if you do not include the `--credit-specification` parameter, the instance launches as **standard** by default.

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t3.micro --key-name MyKeyPair --credit-specification "CpuCredits=unlimited"
```

Using an Auto Scaling Group to Launch a Burstable Performance Instance as Unlimited

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

Creating a Launch Template

You must use a *launch template* for launching instances as **unlimited** in an Auto Scaling group. A launch configuration does not support launching instances as **unlimited**.

To create a launch template that launches instances as Unlimited (console)

1. Follow the [Creating a Launch Template \(p. 323\)](#) procedure.
2. In **Launch template contents**, for **Instance type**, choose a T3 or T2 instance size.
3. To launch instances as unlimited in an Auto Scaling group, in **Advanced details**, for **T2/T3 Unlimited**, choose **Enable**.
4. When you've finished defining the launch template parameters, choose **Create launch template**. For more information, see [Creating a Launch Template \(p. 323\)](#).

To create a launch template that launches instances as Unlimited (AWS CLI)

Use the `create-launch-template` command and specify `unlimited` as the credit specification.

- For T3, if you do not include the `CreditSpecification={CpuCredits=unlimited}` value, the instance launches as unlimited by default.
- For T2, if you do not include the `CreditSpecification={CpuCredits=unlimited}` value, the instance launches as standard by default.

```
aws ec2 create-launch-template --launch-template-name MyLaunchTemplate
--version-description FirstVersion --launch-template-data
ImageId=ami-8c1be5f6,InstanceType=t3.medium,CreditSpecification={CpuCredits=unlimited}
```

Associating an Auto Scaling Group with a Launch Template

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

To create an Auto Scaling group using a launch template (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation bar at the top of the screen, select the same Region that you used when you created the launch template.
3. In the navigation pane, choose **Auto Scaling Groups**, **Create Auto Scaling group**.
4. Choose **Launch Template**, select your launch template, and then choose **Next Step**.
5. Complete the fields for the Auto Scaling group. When you've finished reviewing your configuration settings on the **Review page**, choose **Create Auto Scaling group**. For more information, see [Creating an Auto Scaling Group Using a Launch Template](#) in the *Amazon EC2 Auto Scaling User Guide*.

To create an Auto Scaling group using a launch template (AWS CLI)

Use the `create-auto-scaling-group` AWS CLI command and specify the `--launch-template` parameter.

To add a launch template to an existing Auto Scaling group (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation bar at the top of the screen, select the same Region that you used when you created the launch template.
3. In the navigation pane, choose **Auto Scaling Groups**.
4. From the Auto Scaling group list, select an Auto Scaling group, and choose **Actions**, **Edit**.
5. On the **Details** tab, for **Launch Template**, choose a launch template, and then choose **Save**.

To add a launch template to an existing Auto Scaling group (AWS CLI)

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

Viewing the Credit Specification of a Burstable Performance Instance

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

To view the credit specification of a burstable instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances** and select the instance.
3. Choose **Description** and view the **T2/T3 Unlimited** field.
 - If the value is **Enabled**, then your instance is configured as **unlimited**.
 - If the value is **Disabled**, then your instance is configured as **standard**.

To describe the credit specification of a burstable performance instance (AWS CLI)

Use the [describe-instance-credit-specifications](#) command. If you do not specify one or more instance IDs, all instances with the credit specification of **unlimited** are returned, as well as instances that were previously configured with the **unlimited** credit specification. For example, if you resize a T3 instance to an M4 instance, while it is configured as **unlimited**, Amazon EC2 returns the M4 instance.

Example

```
aws ec2 describe-instance-credit-specifications --instance-id i-1234567890abcdef0
```

The following is example output:

```
{
  "InstanceCreditSpecifications": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "CpuCredits": "unlimited"
    }
  ]
}
```

Modifying the Credit Specification of a Burstable Performance Instance

You can switch the credit specification of a running or stopped instance at any time between **unlimited** and **standard**.

To modify the credit specification of a burstable performance instance (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. To modify the credit specification for several instances at one time, select all applicable instances.
3. Choose **Actions, Instance Settings, Change T2/T3 Unlimited**.

Note

The **Change T2/T3 Unlimited** option is enabled only if you select a T3 or T2 instance.

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

To modify the credit specification of a burstable performance instance (AWS CLI)

Use the [modify-instance-credit-specification](#) command. Specify the instance and its credit specification using the --instance-credit-specification parameter. Valid credit specifications are **unlimited** and **standard**.

Example

```
aws ec2 modify-instance-credit-specification --region us-east-1 --instance-credit-specification "InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

The following is example output:

```
{  
    "SuccessfulInstanceCreditSpecifications": [  
        {  
            "InstanceId": "i- 1234567890abcdef0"  
        }  
    ],  
    "UnsuccessfulInstanceCreditSpecifications": []  
}
```

Monitoring Your CPU Credits

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

Topics

- [Additional CloudWatch Metrics for Burstable Performance Instances \(p. 139\)](#)
- [Calculating CPU Credit Usage \(p. 141\)](#)

Additional CloudWatch Metrics for Burstable Performance Instances

T3 and T2 instances have these additional CloudWatch metrics, which are updated every five minutes:

- **CPUCreditUsage** – The number of CPU credits spent during the measurement period.
- **CPUCreditBalance** – The number of CPU credits that an instance has accrued. This balance is depleted when the CPU bursts and CPU credits are spent more quickly than they are earned.
- **CPUSurplusCreditBalance** – The number of surplus CPU credits spent to sustain CPU performance when the CPUCreditBalance value is zero.
- **CPUSurplusCreditsCharged** – The number of surplus CPU credits exceeding the [maximum number of CPU credits \(p. 118\)](#) that can be earned in a 24-hour period, and thus attracting an additional charge.

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

The following table describes the CloudWatch metrics for burstable performance instances. For more information, see [List the Available CloudWatch Metrics for Your Instances \(p. 500\)](#).

Metric	Description
CPUCreditUsage	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%

Metric	Description
	<p>utilization for two minutes or two vCPUs running at 25% utilization for two minutes).</p> <p>CPU credit metrics are available at a five-minute frequency only. If you specify a period greater than five minutes, use the <code>Sum</code> statistic instead of the <code>Average</code> statistic.</p> <p>Units: Credits (vCPU-minutes)</p>
CPUCreditBalance	<p>The number of earned CPU credits that an instance has accrued since it was launched or started. For T2 Standard, the <code>CPUCreditBalance</code> also includes the number of launch credits that have been accrued.</p> <p>Credits are accrued in the credit balance after they are earned, and removed from the credit balance when they are spent. The credit balance has a maximum limit, determined by the instance size. After the limit is reached, any new credits that are earned are discarded. For T2 Standard, launch credits do not count towards the limit.</p> <p>The credits in the <code>CPUCreditBalance</code> are available for the instance to spend to burst beyond its baseline CPU utilization.</p> <p>When an instance is running, credits in the <code>CPUCreditBalance</code> do not expire. When a T3 instance stops, the <code>CPUCreditBalance</code> value persists for seven days. Thereafter, all accrued credits are lost. When a T2 instance stops, the <code>CPUCreditBalance</code> value does not persist, and all accrued credits are lost.</p> <p>CPU credit metrics are available at a five-minute frequency only.</p> <p>Units: Credits (vCPU-minutes)</p>
CPUSurplusCreditBalance	<p>The number of surplus credits that have been spent by an unlimited instance when its <code>CPUCreditBalance</code> value is zero.</p> <p>The <code>CPUSurplusCreditBalance</code> value is paid down by earned CPU credits. If the number of surplus credits exceeds the maximum number of credits that the instance can earn in a 24-hour period, the spent surplus credits above the maximum incur an additional charge.</p> <p>Units: Credits (vCPU-minutes)</p>

Metric	Description
CPUSurplusCreditsCharged	<p>The number of spent surplus credits that are not paid down by earned CPU credits, and which thus incur an additional charge.</p> <p>Spent surplus credits are charged when any of the following occurs:</p> <ul style="list-style-type: none"> • The spent surplus credits exceed the maximum number of credits that the instance can earn in a 24-hour period. Spent surplus credits above the maximum are charged at the end of the hour. • The instance is stopped or terminated. • The instance is switched from <code>unlimited</code> to <code>standard</code>. <p>Units: Credits (vCPU-minutes)</p>

Calculating CPU Credit Usage

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

Amazon EC2 sends the metrics to CloudWatch every five minutes. A reference to the *prior* value of a metric at any point in time implies the previous value of the metric, sent *five minutes ago*.

Calculating CPU Credit Usage for Standard Instances

- The CPU credit balance increases if CPU utilization is below the baseline, when the credits spent are less than the credits earned in the prior five-minute interval.
- The CPU credit balance decreases if CPU utilization is above the baseline, when the credits spent are more than the credits earned in the prior five-minute interval.

Mathematically, this is captured by the following equation:

Example

```
CPUCreditBalance = prior CPUCreditBalance + [Credits earned per hour * (5/60) - CPUCreditUsage]
```

The size of the instance determines the number of credits that the instance can earn per hour and the number of earned credits that it can accrue in the credit balance. For information about the number of credits earned per hour, and the credit balance limit for each instance size, see the [credit table \(p. 118\)](#).

Example

This example uses a `t3.nano` instance. To calculate the `CPUCreditBalance` value of the instance, use the preceding equation as follows:

- `CPUCreditBalance` – The current credit balance to calculate.
- `prior CPUCreditBalance` – The credit balance five minutes ago. In this example, the instance had accrued two credits.
- `Credits earned per hour` – A `t3.nano` instance earns six credits per hour.
- `5/60` – Represents the five-minute interval between CloudWatch metric publication. Multiply the credits earned per hour by `5/60` (five minutes) to get the number of credits that the instance earned in the past five minutes. A `t3.nano` instance earns 0.5 credits every five minutes.

- **CPUCreditUsage** – How many credits the instance spent in the past five minutes. In this example, the instance spent one credit in the past five minutes.

Using these values, you can calculate the **CPUCreditBalance** value:

Example

```
CPUCreditBalance = 2 + [0.5 - 1] = 1.5
```

Calculating CPU Credit Usage for Unlimited Instances

When a T3 or T2 instance needs to burst above the baseline, it always spends accrued credits before spending surplus credits. When it depletes its accrued CPU credit balance, it can spend surplus credits to burst 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 \(p. 118\)](#) that it can accrue.

Example

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

A negative **Adjusted balance** value indicates that the instance spent all its earned credits that it accrued and, in addition, also spent surplus credits for bursting. As a result, the **Adjusted balance** value is assigned to **CPUSurplusCreditBalance** and **CPUCreditBalance** is set to 0. Again, the instance size determines the [maximum number of credits \(p. 118\)](#) that it can accrue.

Example

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

If the surplus credits spent exceed the maximum credits that the instance can accrue, the surplus credit balance is set to the maximum, as shown in the preceding equation. The remaining surplus credits are charged as represented by the **CPUSurplusCreditsCharged** metric.

Example

```
CPUSurplusCreditsCharged = max [-Adjusted balance - max earned credit balance, 0]
```

Finally, when the instance terminates, any surplus credits tracked by the `CPUSurplusCreditBalance` are charged. If the instance is switched from `unlimited` to `standard`, any remaining `CPUSurplusCreditBalance` is also charged.

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, M5a, and M5d Instances

These instances provide an ideal cloud infrastructure, offering a balance of compute, memory, and networking resources for a broad range of applications that are deployed in the cloud. 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

For more information, see [Amazon EC2 M5 Instances](#).

T2 and T3 Instances

These instances provide a baseline level of CPU performance with the ability to burst to a higher level when required by your workload. An Unlimited instance can sustain high CPU performance for any period of time whenever required. For more information, see [Burstable Performance Instances \(p. 116\)](#). These instances are well-suited for the following applications:

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

For more information, see [Amazon EC2 T2 Instances](#) and [Amazon EC2 T3 Instances](#).

Contents

- [Hardware Specifications \(p. 143\)](#)
- [Instance Performance \(p. 145\)](#)
- [Network Performance \(p. 145\)](#)
- [SSD I/O Performance \(p. 146\)](#)
- [Instance Features \(p. 147\)](#)
- [Release Notes \(p. 147\)](#)

Hardware Specifications

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

Instance type	Default vCPUs	Memory (GiB)
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
m5a.large	2	8
m5a.xlarge	4	16
m5a.2xlarge	8	32
m5a.4xlarge	16	64
m5a.12xlarge	48	192
m5a.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
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

Instance type	Default vCPUs	Memory (GiB)
t3.nano	2	0.5
t3.micro	2	1
t3.small	2	2
t3.medium	2	4
t3.large	2	8
t3.xlarge	4	16
t3.2xlarge	8	32

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 \(p. 425\)](#).

Instance Performance

EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your instance. Some general purpose instances are EBS-optimized by default at no additional cost. For more information, see [Amazon EBS-Optimized Instances \(p. 807\)](#).

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 Windows \(p. 684\)](#).

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 sizes that use the ENA and are documented with network performance of "Up to 10 Gbps" or "Up to 25 Gbps" 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 bandwidth is below their baseline limits, and can use these credits when they perform network data transfers.

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

Instance type	Network performance	Enhanced networking
t2.nano, t2.micro, t2.small, t2.medium, t2.large, t2.xlarge, t2.2xlarge	Up to 1 Gbps	
t3.nano, t3.micro, t3.small, t3.medium, t3.large, t3.xlarge, t3.2xlarge	Up to 5 Gbps	ENAs (p. 685)

Instance type	Network performance	Enhanced networking
m4.large	Moderate	Intel 82599 VF (p. 688)
m4.xlarge, m4.2xlarge, m4.4xlarge	High	Intel 82599 VF (p. 688)
m5.large, m5.xlarge, m5.2xlarge, m5.4xlarge, m5a.large, m5a.xlarge, m5a.2xlarge, m5a.4xlarge, m5d.large, m5d.xlarge, m5d.2xlarge, m5d.4xlarge	Up to 10 Gbps	ENAs (p. 685)
m4.10xlarge	10 Gbps	Intel 82599 VF (p. 688)
m5.12xlarge, m5a.12xlarge, m5d.12xlarge	10 Gbps	ENAs (p. 685)
m5a.24xlarge	20 Gbps	ENAs (p. 685)
m4.16xlarge, m5.24xlarge, m5d.24xlarge	25 Gbps	ENAs (p. 685)

SSD I/O Performance

If you 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
m5d.large *	30,000	15,000
m5d.xlarge *	59,000	29,000
m5d.2xlarge *	117,000	57,000
m5d.4xlarge *	234,000	114,000
m5d.12xlarge	700,000	340,000
m5d.24xlarge	1,400,000	680,000

* For these instances, you can get up to the specified performance.

As you fill the SSD-based instance store volumes for your instance, the number of write IOPS that you can achieve decreases. This is due to the extra work the SSD controller must do to find available space, rewrite existing data, and erase unused space so that it can be rewritten. This process of garbage collection results in internal write amplification to the SSD, expressed as the ratio of SSD write operations to user write operations. This decrease in performance is even larger if the write operations are not in multiples of 4,096 bytes or not aligned to a 4,096-byte boundary. If you write a smaller amount of bytes or bytes that are not aligned, the SSD controller must read the surrounding data and store the result in a new location. This pattern results in significantly increased write amplification, increased latency, and dramatically reduced I/O performance.

SSD controllers can use several strategies to reduce the impact of write amplification. One such strategy is to reserve space in the SSD instance storage so that the controller can more efficiently manage the space available for write operations. This is called *over-provisioning*. The SSD-based instance store volumes provided to an instance don't have any space reserved for over-provisioning. To reduce write amplification, we recommend that you leave 10% of the volume unpartitioned so that the SSD controller can use it for over-provisioning. This decreases the storage that you can use, but increases performance even if the disk is close to full capacity.

For instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see [Instance Store Volume TRIM Support \(p. 851\)](#).

Instance Features

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

	EBS only	NVMe EBS	Instance store	Placement group
M4	Yes			Yes
M5	Yes	Yes		Yes
M5a	Yes	Yes		Yes
M5d		Yes	NVMe *	Yes
T2	Yes			
T3	Yes	Yes		

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

For more information, see the following:

- [Amazon EBS and NVMe \(p. 819\)](#)
- [Amazon EC2 Instance Store \(p. 843\)](#)
- [Placement Groups \(p. 692\)](#)

Release Notes

- M5, M5d, and T3 instances feature a 3.1 GHz Intel Xeon Platinum 8000 series processor.
- M5a instances feature a 2.5 GHz AMD EPYC 7000 series processor.
- M4, M5, M5a, M5d, t2.large and larger, and t3.large and larger instance types require 64-bit HVM AMIs. They have high-memory, and require a 64-bit operating system to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. In addition, you must use an HVM AMI to take advantage of enhanced networking.
- M5, M5a, M5d, and T3 instances have the following requirements:
 - Must have the NVMe drivers installed. EBS volumes are exposed as [NVMe block devices \(p. 819\)](#).
 - Must have the Elastic Network Adapter ([ENA \(p. 685\)](#)) 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, M5a, M5d, and T3 instances support a maximum of 28 attachments, including network interfaces, EBS volumes, and NVMe instance store volumes. Every instance has at least one network interface attachment. For example, if you have no additional network interface attachments on an EBS-only instance, you could attach 27 EBS volumes to that instance.
- 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

For more information, see [Amazon EC2 C5 Instances](#).

Contents

- [Hardware Specifications \(p. 148\)](#)
- [Instance Performance \(p. 149\)](#)
- [Network Performance \(p. 149\)](#)
- [SSD I/O Performance \(p. 150\)](#)
- [Instance Features \(p. 151\)](#)
- [Release Notes \(p. 151\)](#)

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

Instance type	Default vCPUs	Memory (GiB)
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
c5d.4xlarge	16	32
c5d.9xlarge	36	72
c5d.18xlarge	72	144
c5n.large	2	5.25
c5n.xlarge	4	10.5
c5n.2xlarge	8	21
c5n.4xlarge	16	42
c5n.9xlarge	36	96
c5n.18xlarge	72	192

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 \(p. 425\)](#).

Instance Performance

EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your instance. Some compute optimized instances are EBS-optimized by default at no additional cost. For more information, see [Amazon EBS-Optimized Instances \(p. 807\)](#).

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 Windows \(p. 684\)](#).

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 sizes that use the ENA and are documented with network performance of "Up to 10 Gbps" or "Up to 25 Gbps" 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 bandwidth is below their baseline limits, and can use these credits when they perform network data transfers.

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	ENAs (p. 685)
c5.9xlarge c5d.9xlarge	10 Gbps	ENAs (p. 685)
c5n.4xlarge and smaller	Up to 25 Gbps	ENAs (p. 685)
c5.18xlarge c5d.18xlarge	25 Gbps	ENAs (p. 685)
c5n.9xlarge	50 Gbps	ENAs (p. 685)
c5n.18xlarge	100 Gbps	ENAs (p. 685)
c4.large	Moderate	Intel 82599 VF (p. 688)
c4.xlarge c4.2xlarge c4.4xlarge	High	Intel 82599 VF (p. 688)
c4.8xlarge	10 Gbps	Intel 82599 VF (p. 688)

SSD I/O Performance

If you 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
c5d.large *	20,000	9,000
c5d.xlarge *	40,000	18,000
c5d.2xlarge *	80,000	37,000
c5d.4xlarge *	175,000	75,000
c5d.9xlarge	350,000	170,000
c5d.18xlarge	700,000	340,000

* For these instances, you can get up to the specified performance.

As you fill the SSD-based instance store volumes for your instance, the number of write IOPS that you can achieve decreases. This is due to the extra work the SSD controller must do to find available

space, rewrite existing data, and erase unused space so that it can be rewritten. This process of garbage collection results in internal write amplification to the SSD, expressed as the ratio of SSD write operations to user write operations. This decrease in performance is even larger if the write operations are not in multiples of 4,096 bytes or not aligned to a 4,096-byte boundary. If you write a smaller amount of bytes or bytes that are not aligned, the SSD controller must read the surrounding data and store the result in a new location. This pattern results in significantly increased write amplification, increased latency, and dramatically reduced I/O performance.

SSD controllers can use several strategies to reduce the impact of write amplification. One such strategy is to reserve space in the SSD instance storage so that the controller can more efficiently manage the space available for write operations. This is called *over-provisioning*. The SSD-based instance store volumes provided to an instance don't have any space reserved for over-provisioning. To reduce write amplification, we recommend that you leave 10% of the volume unpartitioned so that the SSD controller can use it for over-provisioning. This decreases the storage that you can use, but increases performance even if the disk is close to full capacity.

For instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see [Instance Store Volume TRIM Support \(p. 851\)](#).

Instance Features

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

	EBS only	NVMe EBS	Instance store	Placement group
C4	Yes			Yes
C5	Yes	Yes		Yes
C5d		Yes	NVMe *	Yes
C5n	Yes	Yes		Yes

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

For more information, see the following:

- [Amazon EBS and NVMe \(p. 819\)](#)
- [Amazon EC2 Instance Store \(p. 843\)](#)
- [Placement Groups \(p. 692\)](#)

Release Notes

- C4, C5, C5d, and C5n instances require 64-bit EBS-backed HVM AMIs. They have high-memory and require a 64-bit operating system to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. In addition, you must use an HVM AMI to take advantage of enhanced networking.
- C5, C5d, and C5n instances have the following requirements:
 - Must have the NVMe drivers installed. EBS volumes are exposed as [NVMe block devices \(p. 819\)](#).
 - Must have the Elastic Network Adapter ([ENA \(p. 685\)](#)) 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, C5d, and C5n instances support a maximum of 28 attachments, including network interfaces, EBS volumes, and NVMe instance store volumes. Every instance has at least one network interface attachment.
- 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, R5a, 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.

For more information, see [Amazon EC2 R5 Instances](#).

High memory instances

High memory instances (`u-6tb1.metal`, `u-9tb1.metal`, and `u-12tb1.metal`) offer 6 TiB, 9 TiB, and 12 TiB of memory per instance. These instances are designed to run large in-memory databases, including production installations of SAP HANA. They offer bare metal performance with direct access to host hardware.

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.

For more information, see [Amazon EC2 X1 Instances](#).

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.

For more information, see [Amazon EC2 X1e Instances](#).

z1d Instances

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

- Electronic Design Automation (EDA)
- Relational database workloads

For more information, see [Amazon EC2 z1d Instances](#).

Contents

- [Hardware Specifications \(p. 153\)](#)
- [Memory Performance \(p. 155\)](#)
- [Instance Performance \(p. 155\)](#)
- [Network Performance \(p. 155\)](#)
- [SSD I/O Performance \(p. 156\)](#)
- [Instance Features \(p. 157\)](#)
- [High Availability and Reliability \(X1\) \(p. 157\)](#)
- [Support for vCPUs \(p. 158\)](#)
- [Release Notes \(p. 158\)](#)

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

Instance type	Default vCPUs	Memory (GiB)
r5.4xlarge	16	128
r5.12xlarge	48	384
r5.24xlarge	96	768
r5a.large	2	16
r5a.xlarge	4	32
r5a.2xlarge	8	64
r5a.4xlarge	16	128
r5a.12xlarge	48	384
r5a.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
u-6tb1.metal	448 *	6,144
u-9tb1.metal	448 *	9,216
u-12tb1.metal	448 *	12,288
x1.16xlarge	64	976
x1.32xlarge	128	1,952
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

Instance type	Default vCPUs	Memory (GiB)
z1d.12xlarge	48	384

* Each logical processor is a hyperthread on 224 cores.

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

For more information about specifying CPU options, see [Optimizing CPU Options \(p. 425\)](#).

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 \(p. 153\)](#).

Memory optimized instances have high memory and require 64-bit HVM AMIs to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on memory optimized instances.

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.

High memory instances (`u-6tb1.metal`, `u-9tb1.metal`, and `u-12tb1.metal`) are the first instances to be powered by an eight-socket platform with the latest generation Intel Xeon Platinum 8176M (Skylake) processors that are optimized for mission-critical enterprise workloads.

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.

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 Windows \(p. 684\)](#).

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 sizes that use the ENA and are documented with network performance of "Up to 10 Gbps" or "Up to 25 Gbps" 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 bandwidth is below their baseline limits, and can use these credits when they perform network data transfers.

The following is a summary of network performance for memory optimized instances that support enhanced networking.

Instance type	Network performance	Enhanced networking
r4.4xlarge and smaller r5.4xlarge and smaller r5a.4xlarge and smaller r5d.4xlarge and smaller x1e.8large and smaller z1d.3xlarge and smaller	Up to 10 Gbps	ENAs (p. 685)
r4.8xlarge r5.12xlarge r5a.12xlarge r5d.12xlarge x1.16xlarge x1e.16xlarge z1d.6xlarge	10 Gbps	ENAs (p. 685)
r5a.24xlarge	20 Gbps	ENAs (p. 685)
r4.16xlarge r5.24xlarge r5d.24xlarge u-6tb1.metal u-9tb1.metal u-12tb1.metal x1.32xlarge x1e.32xlarge z1d.12xlarge	25 Gbps	ENAs (p. 685)

SSD I/O Performance

If you 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
r5d.large *	30,000	15,000
r5d.xlarge *	59,000	29,000
r5d.2xlarge *	117,000	57,000
r5d.4xlarge *	234,000	114,000
r5d.12xlarge	700,000	340,000
r5d.24xlarge	1,400,000	680,000
z1d.large *	30,000	15,000
z1d.xlarge *	59,000	29,000
z1d.2xlarge *	117,000	57,000
z1d.3xlarge *	175,000	75,000
z1d.6xlarge	350,000	170,000
z1d.12xlarge	700,000	340,000

* For these instances, you can get up to the specified performance.

As you fill the SSD-based instance store volumes for your instance, the number of write IOPS that you can achieve decreases. This is due to the extra work the SSD controller must do to find available space, rewrite existing data, and erase unused space so that it can be rewritten. This process of garbage collection results in internal write amplification to the SSD, expressed as the ratio of SSD write operations to user write operations. This decrease in performance is even larger if the write operations

are not in multiples of 4,096 bytes or not aligned to a 4,096-byte boundary. If you write a smaller amount of bytes or bytes that are not aligned, the SSD controller must read the surrounding data and store the result in a new location. This pattern results in significantly increased write amplification, increased latency, and dramatically reduced I/O performance.

SSD controllers can use several strategies to reduce the impact of write amplification. One such strategy is to reserve space in the SSD instance storage so that the controller can more efficiently manage the space available for write operations. This is called *over-provisioning*. The SSD-based instance store volumes provided to an instance don't have any space reserved for over-provisioning. To reduce write amplification, we recommend that you leave 10% of the volume unpartitioned so that the SSD controller can use it for over-provisioning. This decreases the storage that you can use, but increases performance even if the disk is close to full capacity.

For instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see [Instance Store Volume TRIM Support \(p. 851\)](#).

Instance Features

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

	EBS only	NVMe EBS	Instance store	Placement group
R4	Yes			Yes
R5	Yes	Yes		Yes
R5a	Yes	Yes		Yes
R5d		Yes	NVME *	Yes
u-6tb1.metal	Yes	Yes		No
u-9tb1.metal	Yes	Yes		No
u-12tb1.metal	Yes	Yes		No
X1			SSD	Yes
X1e			SSD	Yes
z1d		Yes	NVME *	Yes

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

For more information, see the following:

- [Amazon EBS and NVMe \(p. 819\)](#)
- [Amazon EC2 Instance Store \(p. 843\)](#)
- [Placement Groups \(p. 692\)](#)

High Availability and Reliability (X1)

X1 instances support Single Device Data Correction (SDDC +1), which detects and corrects multi-bit errors. SDDC +1 uses error checking and correction code to identify and disable a failed single DRAM device.

In addition, you can implement high availability (HA) and disaster recovery (DR) solutions to meet recovery point objective (RPO), recovery time objective (RTO), and cost requirements by leveraging [Amazon CloudFormation](#) and [Recover Your Instance \(p. 373\)](#). For more information about implementing HA and DR solutions, see the [Using AWS for Disaster Recovery](#) whitepaper.

If you run an SAP HANA production environment, you also have the option of using HANA System Replication (HSR) on X1 instances. For more information about architecting HA and DR solutions on X1 instances, see [SAP HANA on the Amazon Web Services Cloud: Quick Start Reference Deployment](#).

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

- R5 and R5d instances feature a 3.1 GHz Intel Xeon Platinum 8000 series processor.
- R5a instances feature a 2.5 GHz AMD EPYC 7000 series processor.
- The following are requirements for high memory, R5, R5a, R5d, and z1d instances:
 - NVMe drivers must be installed. EBS volumes are exposed as [NVMe block devices \(p. 819\)](#).
 - Elastic Network Adapter ([ENA \(p. 685\)](#)) 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
- R5, R5a, and R5d instances support a maximum of 28 attachments, including network interfaces, EBS volumes, and NVMe instance store volumes. Every instance has at least one network interface attachment. For example, if you have no additional network interface attachments on an EBS-only instance, you could attach 27 EBS volumes to that instance.
- 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.large` and `r4.4xlarge` instances. If you experience this issue, update to the latest version of this AMI.
- There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some instance types. For more information, see [How many instances can I run in Amazon EC2?](#). 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

For more information, see [Amazon EC2 I3 Instances](#).

Contents

- [Hardware Specifications \(p. 160\)](#)
- [Instance Performance \(p. 160\)](#)

- [Network Performance \(p. 161\)](#)
- [SSD I/O Performance \(p. 161\)](#)
- [Instance Features \(p. 162\)](#)
- [Release Notes \(p. 162\)](#)

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 \(p. 843\)](#) and [SSD Instance Store Volumes \(p. 850\)](#).

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 \(p. 425\)](#).

Instance Performance

For instances with NVMe instance store volumes, be sure to use the AWS NVMe driver. For more information, see [AWS NVMe Drivers for Windows Instances \(p. 424\)](#).

EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your instance. Some storage optimized instances are EBS-optimized by default at no additional cost. For more information, see [Amazon EBS-Optimized Instances \(p. 807\)](#).

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 Windows \(p. 684\)](#).

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 sizes that use the ENA and are documented with network performance of "Up to 10 Gbps" or "Up to 25 Gbps" 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 bandwidth is below their baseline limits, and can use these credits when they perform network data transfers.

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. 685)
i3.8xlarge h1.8xlarge	10 Gbps	ENA (p. 685)
i3.16xlarge i3.metal h1.16xlarge	25 Gbps	ENA (p. 685)
d2.xlarge	Moderate	Intel 82599 VF (p. 688)
d2.2xlarge d2.4xlarge	High	Intel 82599 VF (p. 688)
d2.8xlarge	10 Gbps	Intel 82599 VF (p. 688)

SSD I/O Performance

If you 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 \(p. 851\)](#).

Instance Features

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

	EBS only	NVMe EBS	Instance store	Placement group
D2			HDD	Yes
H1			HDD	Yes
I3			NVMe *	Yes

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

For more information, see the following:

- [Amazon EBS and NVMe \(p. 819\)](#)
- [Amazon EC2 Instance Store \(p. 843\)](#)
- [Placement Groups \(p. 692\)](#)

Release Notes

- You must launch storage optimized instances using an HVM AMI.
- 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 \(p. 819\)](#).
 - Elastic Network Adapter ([ENA \(p. 685\)](#)) 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.
- `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.
- 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](#).

Windows 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.

If your application needs a small amount of additional graphics acceleration, but is better suited for an instance type with different compute, memory, or storage specifications, use an Elastic Graphics accelerator instead. For more information, see [Amazon Elastic Graphics \(p. 475\)](#).

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 \(p. 692\)](#).

Contents

- [Accelerated Computing Instance Families \(p. 164\)](#)

- [Hardware Specifications \(p. 165\)](#)
- [Instance Performance \(p. 166\)](#)
- [Network Performance \(p. 166\)](#)
- [Instance Features \(p. 166\)](#)
- [Release Notes \(p. 167\)](#)
- [AMIs for GPU-Based Accelerated Computing Instances \(p. 167\)](#)
- [Installing the NVIDIA Driver on Windows \(p. 168\)](#)
- [Activate NVIDIA GRID Virtual Applications \(G3 Instances Only\) \(p. 169\)](#)
- [Optimizing GPU Settings \(P2, P3, and G3 Instances\) \(p. 170\)](#)

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

P3 Instances

P3 instances use NVIDIA Tesla V100 GPUs and are designed for general purpose GPU computing using the CUDA or OpenCL programming models or through a machine learning framework. P3 instances provide high-bandwidth networking, powerful half, single, and double-precision floating-point capabilities, and up to 32 GiB of memory per GPU, which makes them ideal for deep learning, computational fluid dynamics, computational finance, seismic analysis, molecular modeling, genomics, rendering, and other server-side GPU compute workloads. Tesla V100 GPUs do not support graphics mode. For more information, see [Amazon EC2 P3 Instances](#).

P3 instances support NVIDIA NVLink peer to peer transfers.

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 \(G3 Instances Only\) \(p. 169\)](#).

G2 Instances

G2 instances use NVIDIA GRID K520 GPUs and provide a cost-effective, high-performance platform for graphics applications using DirectX or OpenGL. NVIDIA GRID GPUs also support NVIDIA's fast capture and encode API operations. Example applications include video creation services, 3D visualizations, streaming graphics-intensive applications, and other server-side graphics workloads.

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
p3dn.24xlarge	96	768
g2.2xlarge	8	15
g2.8xlarge	32	60
g3s.xlarge	4	30.5
g3.4xlarge	16	122
g3.8xlarge	32	244
g3.16xlarge	64	488
f1.2xlarge	8	122
f1.4xlarge	16	244
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 \(p. 425\)](#).

Instance Performance

EBS-optimized instances enable you to get consistently high performance for your EBS volumes by eliminating contention between Amazon EBS I/O and other network traffic from your instance. Some accelerated computing instances are EBS-optimized by default at no additional cost. For more information, see [Amazon EBS-Optimized Instances \(p. 807\)](#).

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 Windows \(p. 684\)](#).

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 sizes that use the ENA and are documented with network performance of "Up to 10 Gbps" or "Up to 25 Gbps" 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 bandwidth is below their baseline limits, and can use these credits when they perform network data transfers.

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

Instance type	Network performance	Enhanced networking
f1.2xlarge f1.4xlarge g3.4xlarge p3.2xlarge	Up to 10 Gbps	ENA (p. 685)
g3s.xlarge g3.8xlarge p2.8xlarge p3.8xlarge	10 Gbps	ENA (p. 685)
f1.16xlarge g3.16.xlarge g3.16.xlarge p2.16xlarge p3.16xlarge	25 Gbps	ENA (p. 685)
p3dn.24xlarge	100 Gbps	ENA (p. 685)

Instance Features

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

	EBS only	NVMe EBS	Instance store	Placement group
G2			SSD	Yes
G3	Yes			Yes
P2	Yes			Yes
P3	p3dn.24xlarge: No All other sizes: Yes	p3dn.24xlarge: Yes All other sizes: No	p3dn.24xlarge: NVMe *	Yes

	EBS only	NVMe EBS	Instance store	Placement group
F1			NVMe *	Yes

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

For more information, see the following:

- [Amazon EBS and NVMe \(p. 819\)](#)
- [Amazon EC2 Instance Store \(p. 843\)](#)
- [Placement Groups \(p. 692\)](#)

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](#).
- If you launch a multi-GPU instance with a Windows AMI that was created on a single-GPU instance, Windows does not automatically install the NVIDIA driver for all GPUs. You must authorize the driver installation for the new GPU hardware. You can correct this manually in the Device Manager by opening the **Other** device category (the inactive GPUs do not appear under **Display Adapters**). For each inactive GPU, open the context (right-click) menu, choose **Update Driver Software**, and then choose the default **Automatic Update** option.
- When using Microsoft Remote Desktop Protocol (RDP), GPUs that use the WDDM driver model are replaced with a non-accelerated Remote Desktop display driver. To access your GPU hardware, you must use a different remote access tool, such as [Teradici Cloud Access Software](#), [NICE Desktop Cloud Visualization \(DCV\)](#), or VNC. You can also use one of the GPU AMIs from the AWS Marketplace because they provide remote access tools that support 3D acceleration.

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 Windows \(p. 168\)](#).

Installing the NVIDIA Driver on Windows

To install the NVIDIA driver on your Windows instance, log on to your instance as the administrator using Remote Desktop and download the appropriate driver. The driver that you download and install depends on the instance type.

NVIDIA GRID Drivers for (G3)

For G3 instances, you can download the NVIDIA GRID driver from Amazon S3 using the AWS Tools for Windows PowerShell.

Important

This download is available to AWS customers only. By downloading, you agree that you will only

use the downloaded software 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](#).

Prerequisites

Associate an IAM role with your instance that has permissions to use the s3>ListBucket and s3GetObject actions. For more information, see [IAM Roles for Amazon EC2 \(p. 633\)](#). Alternatively, configure the Tools for Windows PowerShell to use your AWS credentials. For more information, see [Using AWS Credentials](#).

To install the NVIDIA GRID driver (G3 instances)

1. Open a PowerShell window.
2. Download the drivers and the [NVIDIA GRID Cloud End User License Agreement](#) to your desktop with the following PowerShell commands (you can copy and paste the entire block of commands at one time).

```
$Bucket = "ec2-windows-nvidia-drivers"
$keyPrefix = "latest"
$localPath = "C:\Users\Administrator\Desktop\NVIDIA"
$objects = Get-S3Object -BucketName $Bucket -KeyPrefix $keyPrefix -Region us-east-1
foreach ($object in $objects) {
    $localFileName = $object.Key
    if ($localFileName -ne '' -and $object.Size -ne 0) {
        $localFilePath = Join-Path $localPath $localFileName
        Copy-S3Object -BucketName $Bucket -Key $object.Key -LocalFile $localFilePath -Region us-east-1
    }
}
```

Multiple versions of the NVIDIA GRID driver are stored in this bucket. You can download all of the available versions in the bucket by removing the `-KeyPrefix $keyPrefix` option in the above code block.

3. Navigate to the desktop and double-click the installation file to launch it (choose the driver version that corresponds to your instance OS version). Follow the instructions to install the driver and reboot your instance as required. To verify that the GPU is working properly, check Device Manager.
4. Disable the licensing page in the control panel to prevent users from accidentally changing the product type (NVIDIA GRID Virtual Workstation is enabled by default). For more information, see the [GRID Licensing User Guide](#).
 - a. Run `regedit.exe` to open the registry editor.

- b. Navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global\GridLicensing`.
 - c. Open the context (right-click) menu on the right pane and choose **New, DWORD**.
 - d. For **Name**, enter `NvCplDisableManageLicensePage` and type `Enter`.
 - e. Open the context (right-click) menu on `NvCplDisableManageLicensePage` and choose **Modify**.
 - f. For **Value data**, type `1` and choose **OK**.
5. (Optional) To enable NVIDIA GRID Virtual Applications, complete the GRID activation steps in [Activate NVIDIA GRID Virtual Applications \(G3 Instances Only\) \(p. 169\)](#) (NVIDIA GRID Virtual Workstation is enabled by default).
 6. Complete the optimization steps in [Optimizing GPU Settings \(P2, P3, and G3 Instances\) \(p. 170\)](#) to achieve the best performance from your GPU.

Public NVIDIA Drivers (G2, P2, P3)

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

To install a public NVIDIA driver

1. Download the public NVIDIA driver that is 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

2. Open the folder where you downloaded the driver and launch the installation file. Follow the instructions to install the driver and reboot your instance as required.
3. Disable the built-in display adapter using Device Manager. Install Media Foundation and Quality Windows Audio Video Experience.
4. To verify that the GPU is working correctly, check Device Manager.
5. [P2, P3, and G3] Complete the optimization steps in [Optimizing GPU Settings \(P2, P3, and G3 Instances\) \(p. 170\)](#) 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 registry.

To activate GRID Virtual Applications on G3 Windows instances

1. Run `regedit.exe` to open the registry editor.
2. Navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global\GridLicensing`.
3. Open the context (right-click) menu on the right pane and choose **New, DWORD**.
4. For **Name**, enter `FeatureType` and type `Enter`.
5. Open the context (right-click) menu on `FeatureType` and choose **Modify**.
6. For **Value data**, enter `0` for NVIDIA GRID Virtual Applications and choose **OK**.

7. Open the context (right-click) menu on the right pane and choose **New, DWORD**.
8. For **Name**, enter **IgnoreSP** and type **Enter**.
9. Open the context (right-click) menu on **IgnoreSP** and choose **Modify**.
10. For **Value data**, type **1** and choose **OK**.
11. Close the registry editor.

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.

To optimize GPU settings

1. Open a PowerShell window and navigate to the NVIDIA installation folder.

```
cd "C:\Program Files\NVIDIA Corporation\NVSMI"
```

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

```
.\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:

```
.\nvidia-smi -ac "2505,875"
```

- P3 instances:

```
.\nvidia-smi -ac "877,1530"
```

- G3 instances:

```
.\nvidia-smi -ac "2505,1177"
```

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 [Burstable Performance Instances \(p. 116\)](#).

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

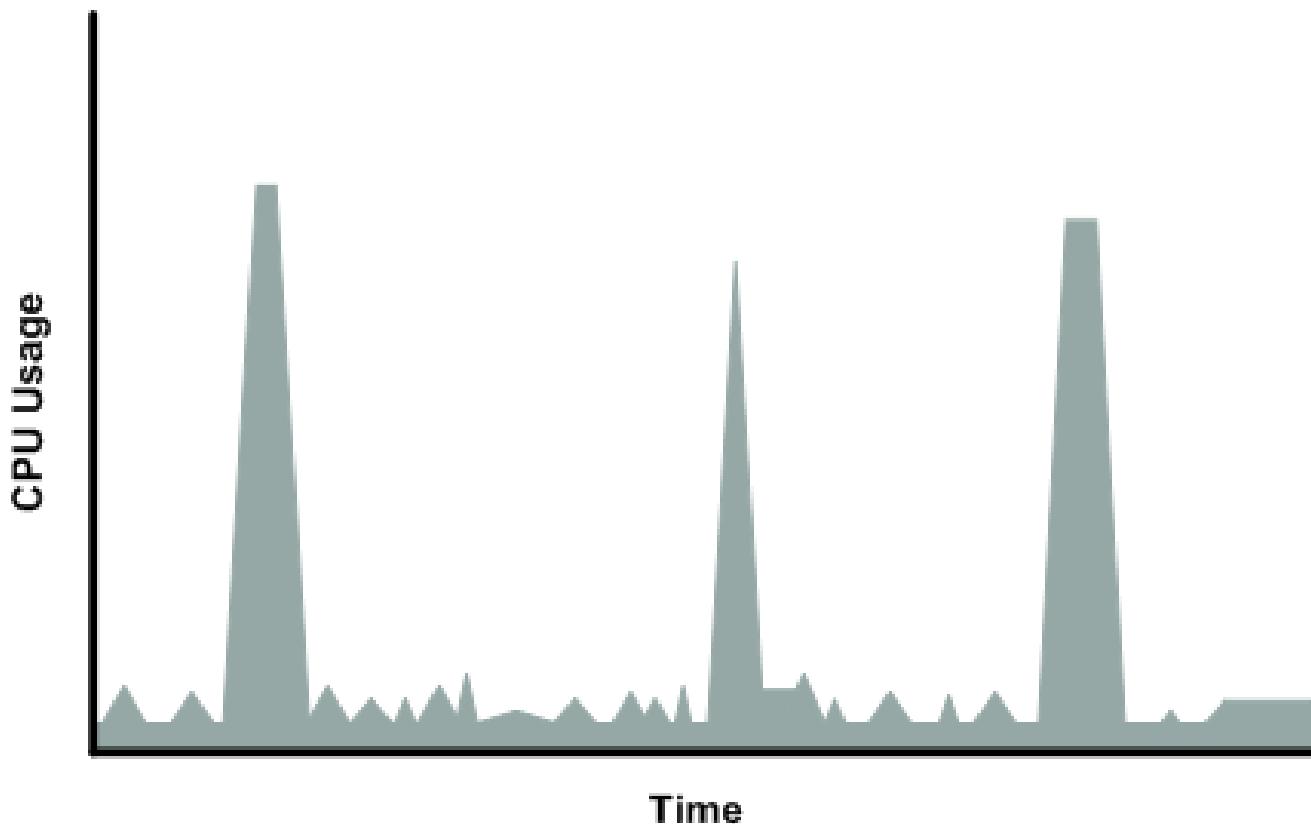
- [Hardware Specifications \(p. 171\)](#)
- [Optimal Application of T1 Micro Instances \(p. 171\)](#)
- [Available CPU Resources During Spikes \(p. 175\)](#)
- [When the Instance Uses Its Allotted Resources \(p. 175\)](#)
- [Comparison with the m1.small Instance Type \(p. 178\)](#)
- [AMI Optimization for Micro Instances \(p. 181\)](#)

Hardware Specifications

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

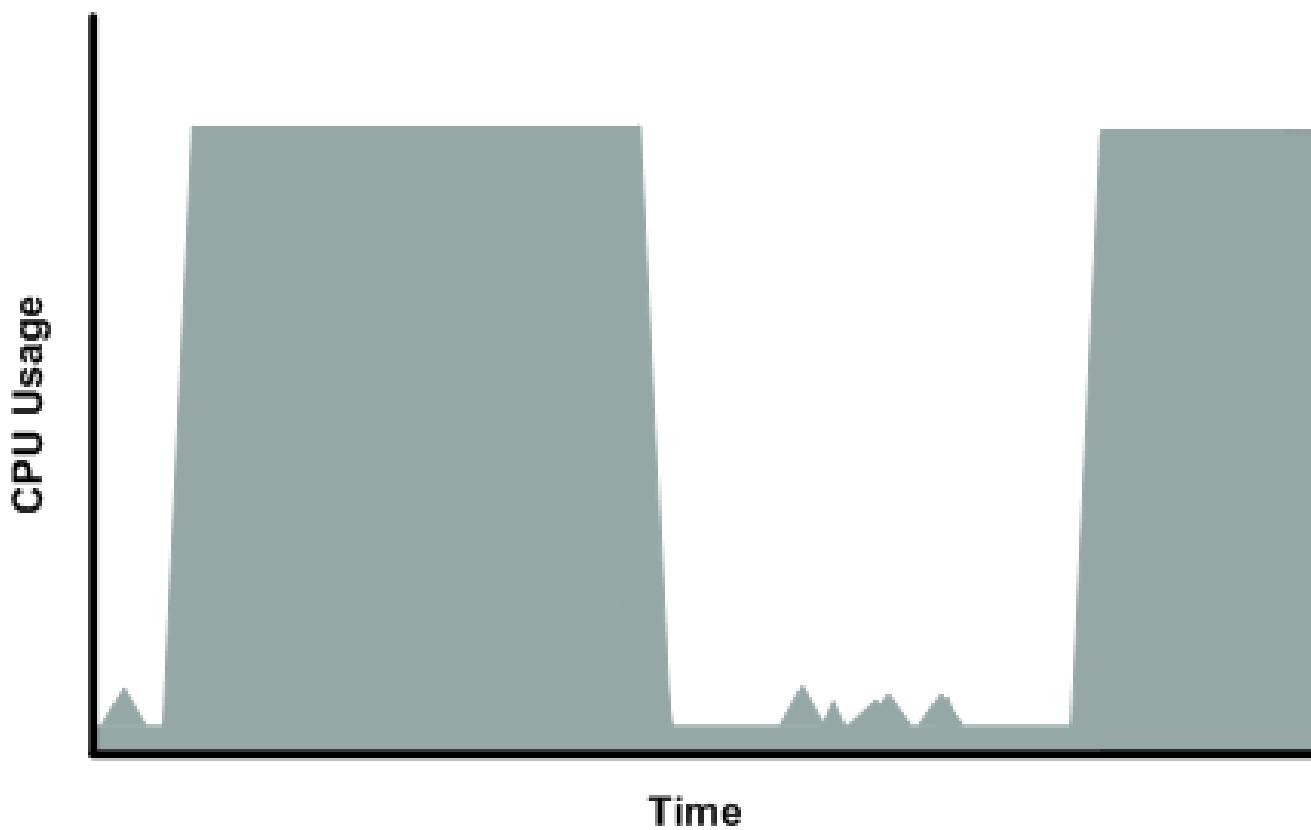
Optimal Application of T1 Micro Instances

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.

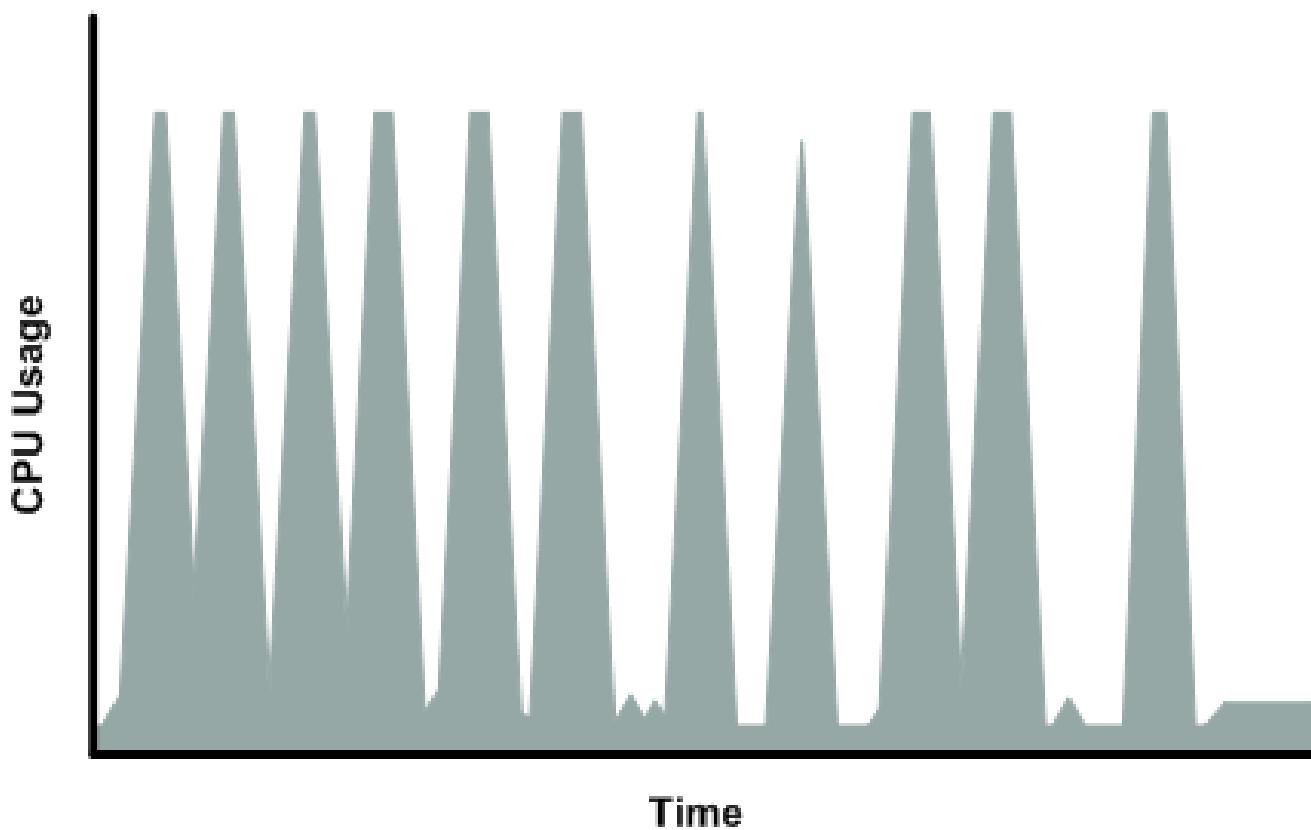


The instance is designed to operate with its CPU usage at essentially only two levels: the normal low background level, and then at brief spiked levels much higher than the background level. We allow the instance to operate at up to 2 EC2 compute units (ECUs) (one ECU provides the equivalent CPU capacity 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.

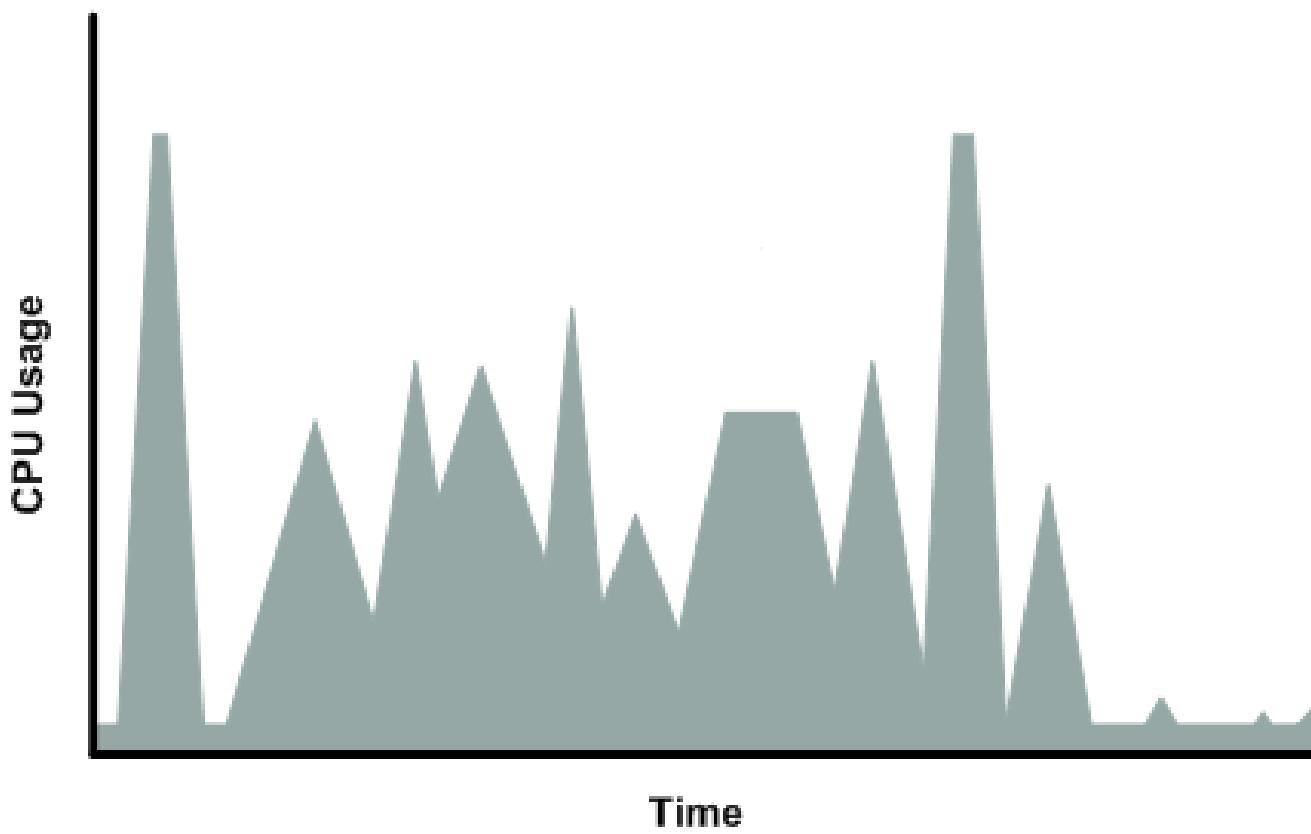
Your application might have a different CPU usage profile than that described in the preceding section. The following figure shows the profile for an application that isn't appropriate for a t1.micro instance. The application requires continuous data-crunching CPU resources for each request, resulting 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 \(p. 110\)](#).

Available CPU Resources During Spikes

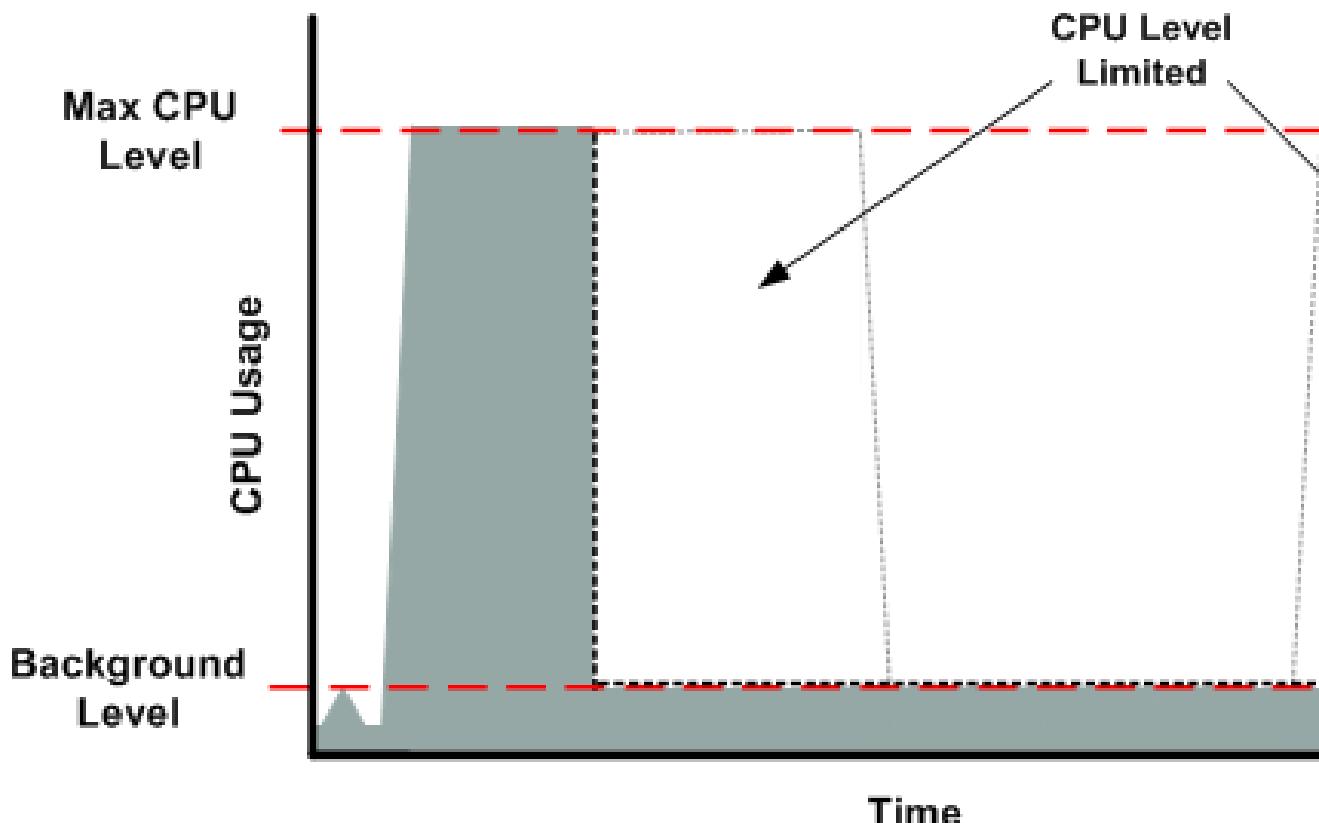
When your instance *bursts* to accommodate a spike in demand for compute resources, it uses unused resources on the host. The amount available depends on how much contention there is when the spike occurs. The instance is never left with zero CPU resources, whether other instances on the host are spiking or not.

When the Instance Uses Its Allotted Resources

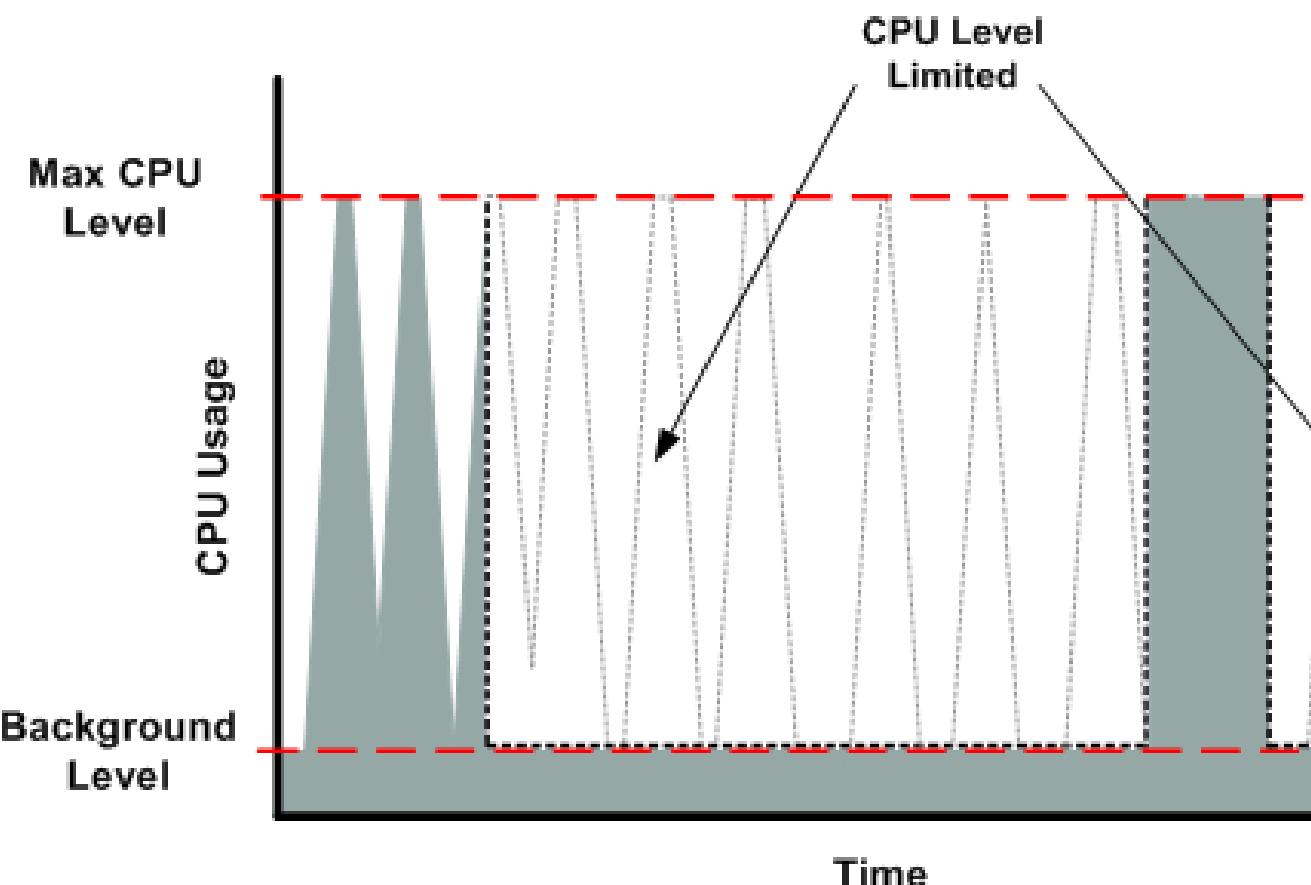
We expect your application to consume only a certain amount of CPU resources in a period of time. If the application consumes more than your instance's allotted CPU resources, we temporarily limit the instance so it operates at a low CPU level. If your instance continues to use all of its allotted resources, 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](#).

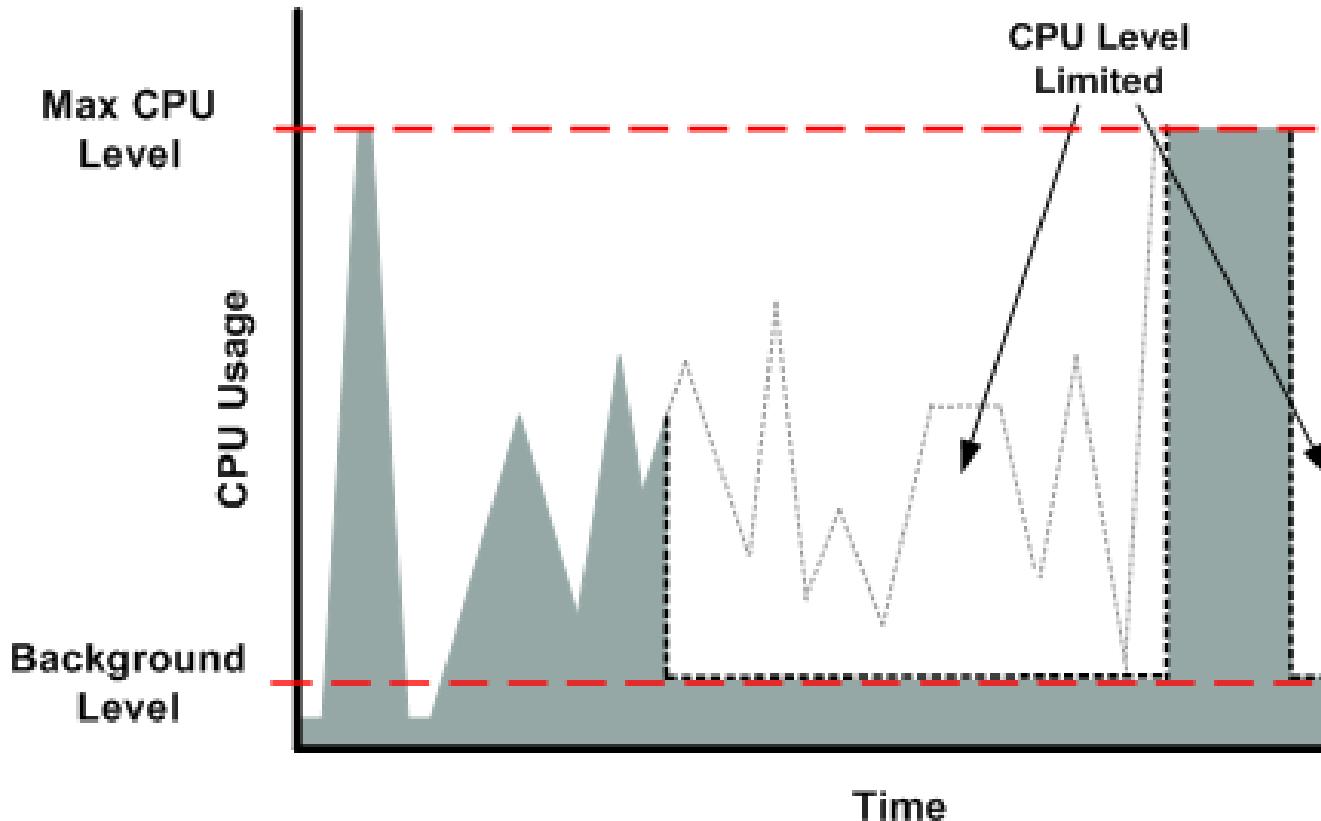
Consider the three suboptimal profiles from the preceding section and what it might look like when the instance consumes its allotted resources and we limit its CPU level. If an instance consumes its allotted resources, we restrict it to the low background level. The following figure shows long plateaus 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.

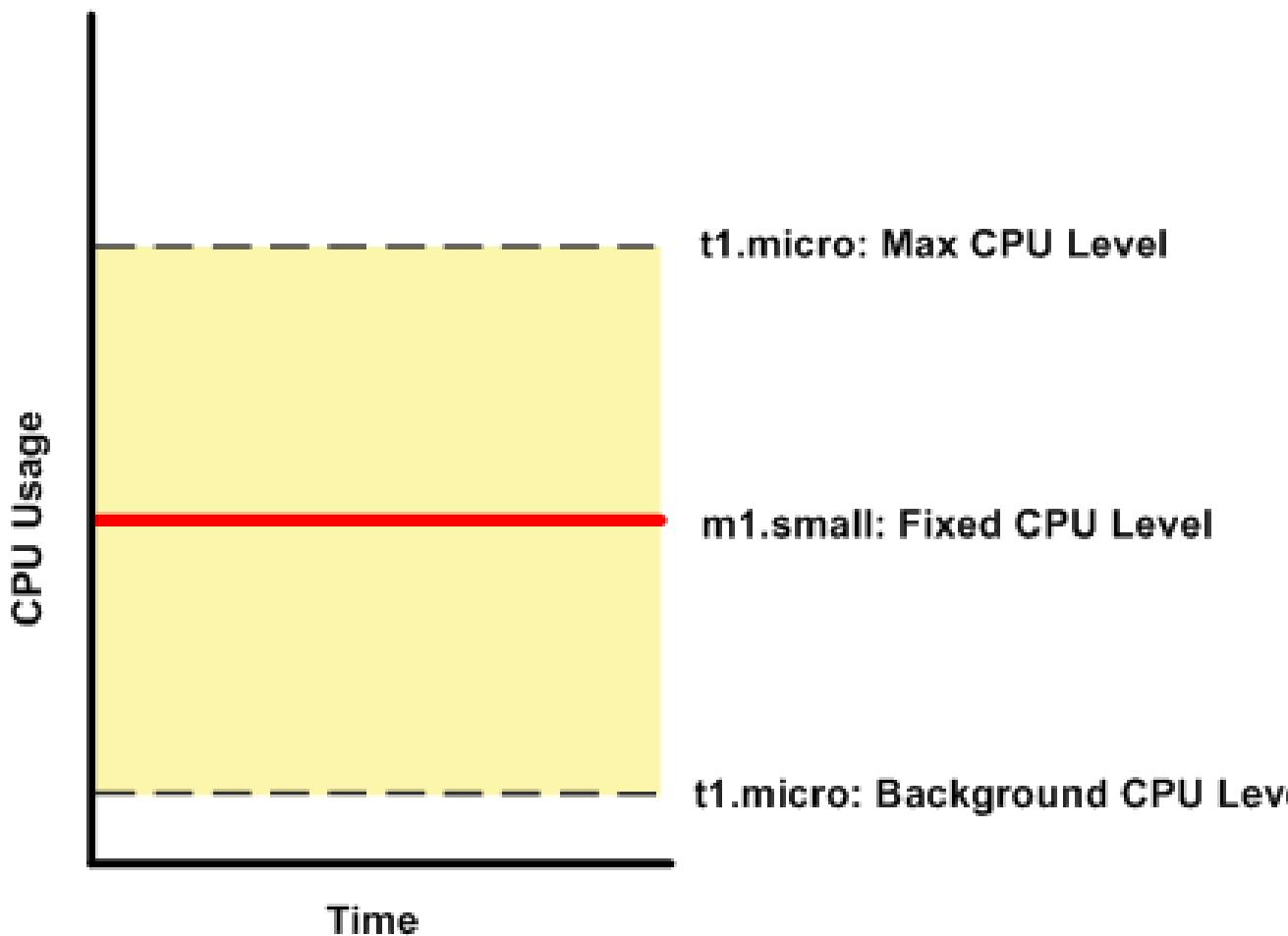


The following figure shows a background level that is too high. Notice that the instance doesn't have to be operating at the maximum CPU level for us to limit it. We limit the instance when it's operating above the normal background level and has consumed its allotted resources for the given period. In this case (as in the preceding one), the instance can't keep up with the work, and we limit it again.

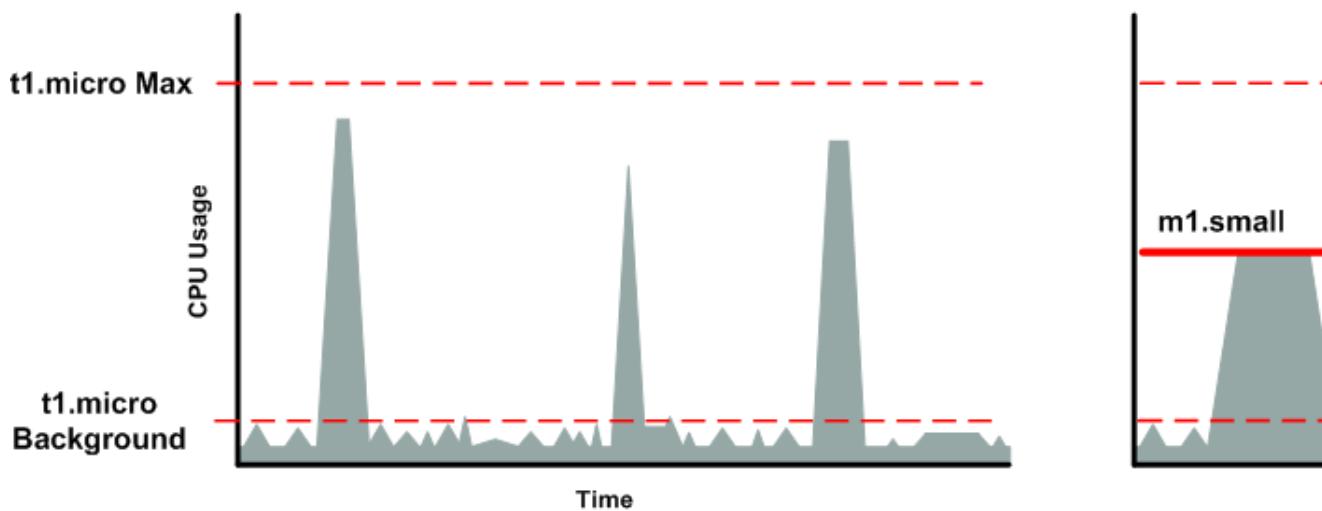


Comparison with the m1.small Instance Type

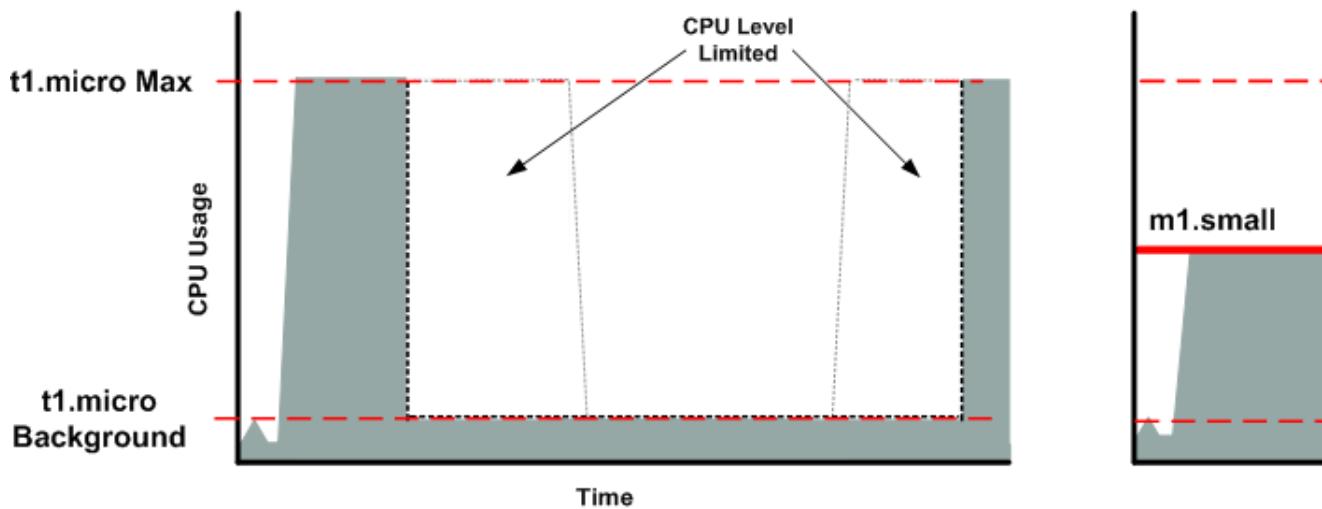
The `t1.micro` instance provides different levels of CPU resources at different times (up to 2 ECUs). By comparison, the `m1.small` instance type provides 1 ECU at all times. The following figure illustrates the difference.



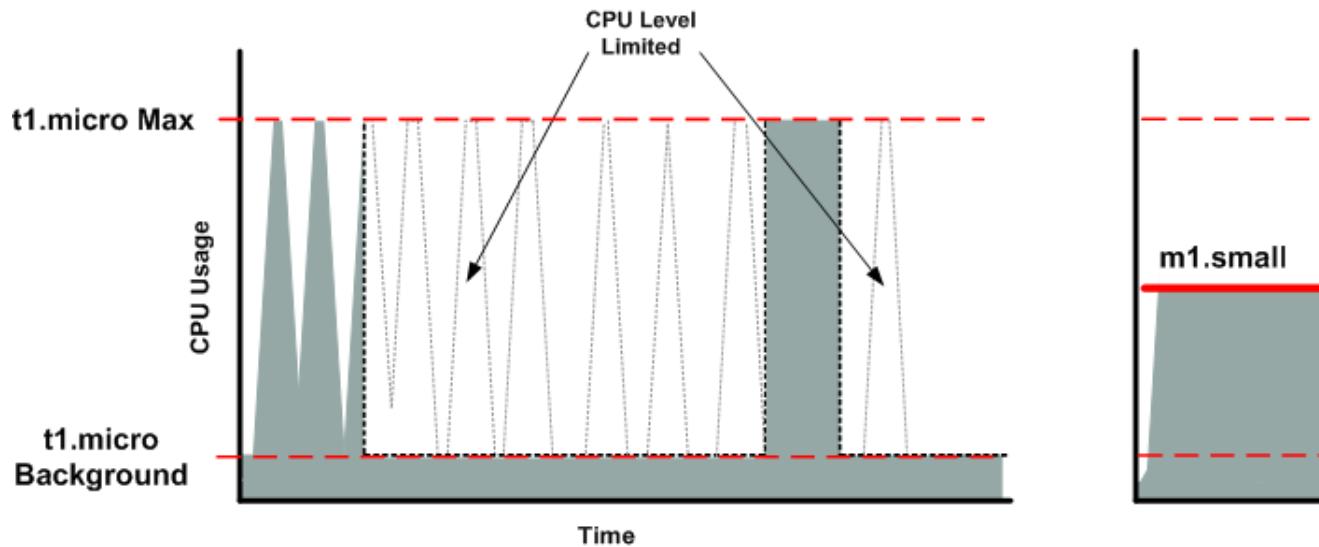
Let's compare the CPU usage of a **t1.micro** instance with an **m1.small** instance for the various scenarios we've discussed in the preceding sections. The following figure that follows shows an optimal scenario for a **t1.micro** instance (the left graph) and how it might look for an **m1.small** 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.



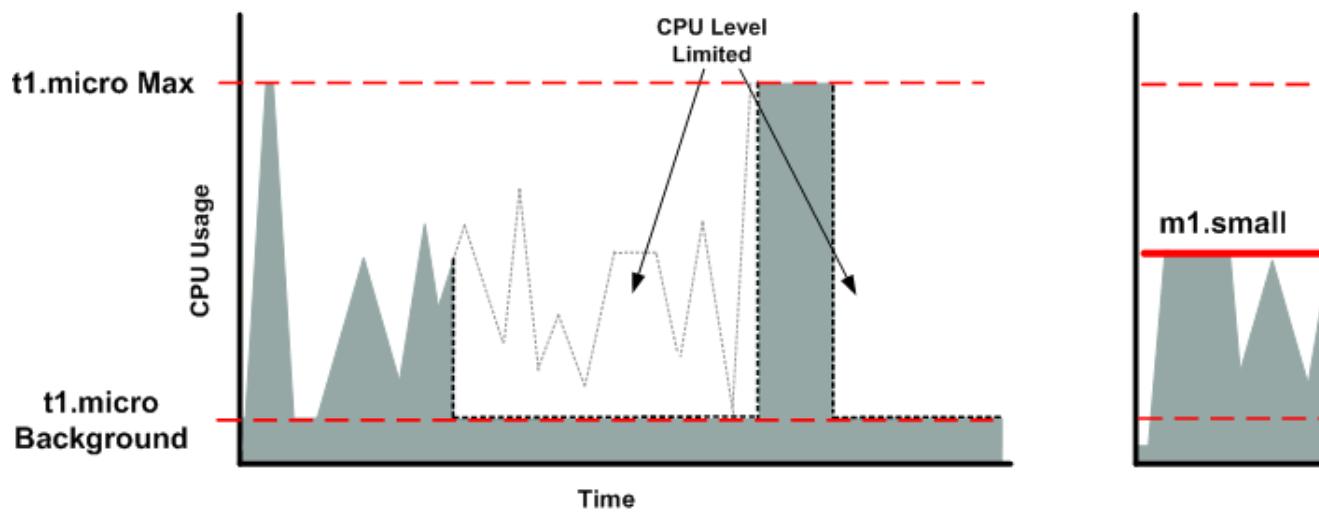
The following figure shows the scenario with the data-crunching requests that used up the allotted resources on the **t1.micro** instance, and how they might look with the **m1.small** instance.



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)

When you perform significant AMI or instance configuration changes (for example, enable server roles or install large applications), you might see limited instance performance, because these changes can be memory intensive and require long-running CPU resources. We recommend that you first use a larger

instance type when performing these changes to the AMI, and then run the AMI on a `t1.micro` instance for normal operations.

Changing the Instance Type

As your needs change, you might find that your instance is over-utilized (the instance type is too small) or under-utilized (the instance type is too large). If this is the case, you can change the size of your instance. For example, if your `t2.micro` instance is too small for its workload, you can change it to another instance type that is appropriate for the workload.

You might also want to migrate from a previous generation instance type to a current generation instance type to take advantage of some features; for example, support for IPv6.

You can change the size of the instance simply by changing its instance type, which is known as *resizing* it.

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 \(p. 843\)](#).

Contents

- [Compatibility for Resizing Instances \(p. 182\)](#)
- [Resizing an Amazon EBS-backed Instance \(p. 183\)](#)
- [Migrating to a New Instance Configuration \(p. 184\)](#)

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:

- **Architecture:** AMIs are specific to the architecture of the processor, so you must select an instance type with the same processor architecture as the current instance type. For example:
 - A1 instances are the only instances that support processors based on the Arm architecture. If you are resizing an instance type with a processor based on the Arm architecture, you are limited to the instance types that support a processor based on the Arm architecture.
 - The following instance types are the only instance types that support 32-bit AMIs: `t2.nano`, `t2.micro`, `t2.small`, `t2.medium`, `c3.large`, `t1.micro`, `m1.small`, `m1.medium`, and `c1.medium`. If you are resizing a 32-bit instance, you are limited to these instance types.
- **Network:** Newer instance types must be launched in a VPC. Therefore, you can't resize an instance in the EC2-Classic platform to a instance type that is available only in a VPC unless you have a nondefault VPC. To check whether your instance is in a VPC, check the **VPC ID** value on the details pane of the **Instances** screen in the Amazon EC2 console. For more information, see [Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC \(p. 721\)](#).
- **Enhanced networking:** Instance types that support [enhanced networking \(p. 684\)](#) require the necessary drivers installed. For example, the C5, C5d, C5n, M5, M5a, M5d, p3dn.24xlarge, R5, R5a, R5d, T3, 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 [ENAv drivers \(p. 685\)](#) or [ixgbevf drivers \(p. 688\)](#) on your instance, as appropriate.

- **NVMe:** EBS volumes are exposed as NVMe block devices on [Nitro-based instances \(p. 113\)](#). If you resize an instance from an instance type that does not support NVMe to an instance type that supports NVMe, you must first install the [NVMe drivers \(p. 819\)](#) on your instance. Also, the device names for devices that you specify in the block device mapping are renamed using NVMe device names (`/dev/nvme[0-26]n1`).
- **AMI:** For information about the AMIs required by instance types that support enhanced networking and NVMe, see the Release Notes in the following documentation:
 - [General Purpose Instances \(p. 143\)](#)
 - [Compute Optimized Instances \(p. 148\)](#)
 - [Memory Optimized Instances \(p. 152\)](#)
 - [Storage Optimized Instances \(p. 159\)](#)

Resizing an Amazon EBS-backed Instance

You must stop your Amazon EBS-backed instance before you can change its instance type. When you stop and start an instance, be aware of the following:

- We move the instance to new hardware; however, the instance ID does not change.
- If your instance has a public IPv4 address, we release the address and give it a new public IPv4 address. The instance retains its private IPv4 addresses, any Elastic IP addresses, and any IPv6 addresses.
- If your instance is in an Auto Scaling group, the Amazon EC2 Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. To prevent this, you can suspend the scaling processes for the group while you're resizing your instance. For more information, see [Suspending and Resuming Scaling Processes](#) in the *Amazon EC2 Auto Scaling User Guide*.
- 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 \(p. 362\)](#).

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 \(p. 182\)](#).

Note

The AWS PV driver package should be updated before changing instance families. For more information, see [Upgrading PV Drivers on Your Windows Instances \(p. 414\)](#).

2. (Optional) If you've configured your instance to use [static IP addressing \(p. 443\)](#) and you resize the instance from a type that doesn't support enhanced networking to an instance type that does support enhanced networking you might get a warning about a potential IP address conflict when you reconfigure static IP addressing. To prevent this, enable DHCP on the network interface for your instance before you change the instance type. From your instance, open the **Network and Sharing Center**, go to **Internet Protocol Version 4 (TCP/IPv4) Properties** for the network interface, and choose **Obtain an IP address automatically**. Change the instance type and reconfigure static IP addressing on the network interface.

3. Open the Amazon EC2 console.
4. [Windows Server 2016] Connect to your Windows instance and run the following EC2Launch PowerShell script to configure the instance after it is resized.

```
PS C:> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -  
Schedule
```

5. In the navigation pane, choose **Instances**, and select the instance.
6. Choose **Actions**, select **Instance State**, and then choose **Stop**.
7. In the confirmation dialog box, choose **Yes, Stop**. It can take a few minutes for the instance to stop.
8. 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.
9. 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.
10. To restart the stopped instance, select the instance, choose **Actions**, select **Instance State**, and then choose **Start**.
11. In the confirmation dialog box, choose **Yes, Start**. It can take a few minutes for the instance to enter the running state.

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.

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 \(p. 789\)](#)) 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 \(p. 771\)](#)).
2. Launch a new instance, selecting the following:
 - 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**, **Instance State**, **Terminate**.

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 hour, 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.
- **Capacity Reservations** – Reserve capacity for your EC2 instances in a specific Availability Zone for any duration.

If you require a capacity reservation, purchase Reserved Instances or Capacity Reservations for a specific Availability Zone, or purchase Scheduled Instances. Spot Instances are a cost-effective choice if you can be flexible about when your applications run and if they can be interrupted. Dedicated Hosts can help you address compliance requirements and reduce costs by using your existing server-bound software licenses. For more information, see [Amazon EC2 Pricing](#).

Contents

- [Determining the Instance Lifecycle \(p. 185\)](#)
- [Reserved Instances \(p. 186\)](#)
- [Scheduled Reserved Instances \(p. 221\)](#)
- [Spot Instances \(p. 225\)](#)
- [Dedicated Hosts \(p. 283\)](#)
- [Dedicated Instances \(p. 297\)](#)
- [On-Demand Capacity Reservations \(p. 302\)](#)

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 \(p. 901\)](#).

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:

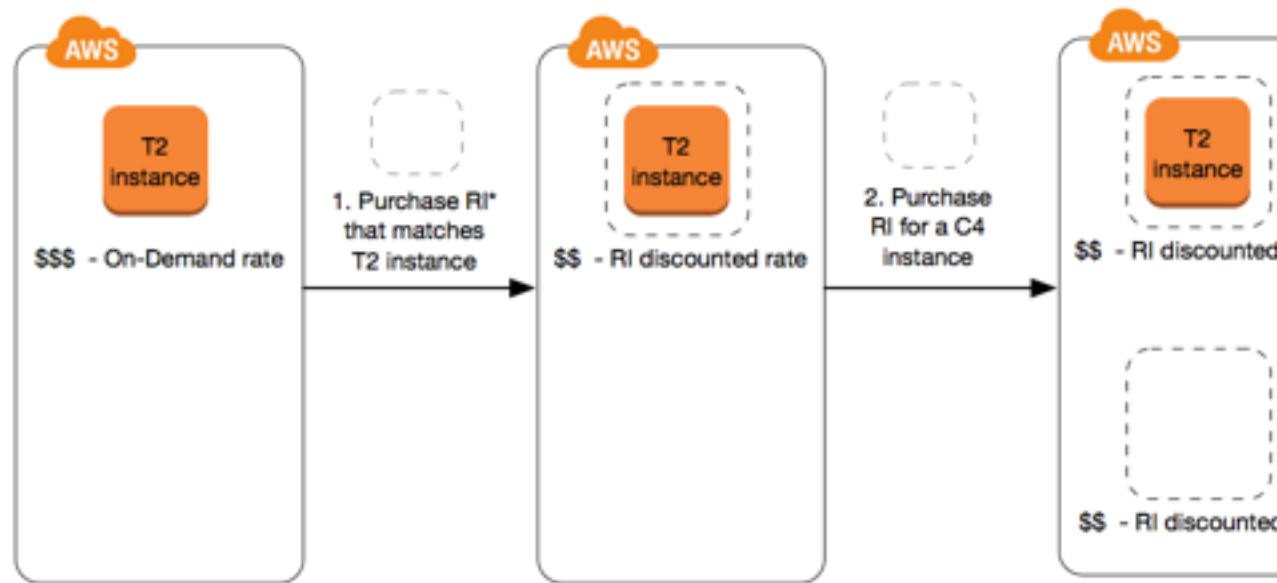
```
"InstanceLifecycle": "scheduled"
```

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 \(p. 297\)](#).
- **Platform:** The operating system; for example, Windows or Linux/Unix. For more information, see [Choosing a Platform \(p. 198\)](#).

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 \(p. 210\)](#), [exchange \(p. 217\)](#), or [sell \(p. 204\)](#) 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 \(p. 204\)](#).

For more information about pricing, see [Amazon EC2 Reserved Instances Pricing](#).

Reserved Instance Limits

There is a limit to the number of Reserved Instances that you can purchase per month. For each Region you can purchase 20 [regional \(p. 190\)](#) Reserved Instances per month plus an additional 20 [zonal \(p. 190\)](#) Reserved Instances per month for each Availability Zone.

For example, in a Region with three Availability Zones, the limit is 80 Reserved Instances per month: 20 regional Reserved Instances for the Region plus 20 zonal Reserved Instances for each of the three Availability Zones ($20 \times 3 = 60$).

A regional Reserved Instance applies a discount to a running On-Demand Instance. The default On-Demand Instance limit is 20. You cannot exceed your running On-Demand Instance limit by purchasing regional Reserved Instances. For example, if you already have 20 running On-Demand Instances, and you purchase 20 regional Reserved Instances, the 20 regional Reserved Instances are used to apply a discount to the 20 running On-Demand Instances. If you purchase more regional Reserved Instances, you will not be able to launch more instances because you have reached your On-Demand Instance limit.

Note

Before purchasing regional Reserved Instances, make sure your On-Demand Instance limit matches or exceeds the number of regional Reserved Instances you intend to own. If required, make sure you request an increase to your On-Demand Instance limit *before* purchasing more regional Reserved Instances.

A zonal Reserved Instance—a Reserved Instance that is purchased for a specific Availability Zone—provides capacity reservation as well as a discount. You *can exceed* your running On-Demand Instance limit by purchasing zonal Reserved Instances. For example, if you already have 20 running On-Demand Instances, and you purchase 20 zonal Reserved Instances, you can launch a further 20 On-Demand Instances that match the specifications of your zonal Reserved Instances, giving you a total of 40 running instances.

The Amazon EC2 console provides limit information. For more information, see [Viewing Your Current Limits \(p. 900\)](#).

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. 210) .	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. 217) . You can also modify some attributes of a Convertible Reserved Instance. For more information, see Modifying Reserved Instances (p. 210) .
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.

Note

- When you purchase a Reserved Instance for a specific Availability Zone, it's referred to as a *zonal* Reserved Instance. A zonal Reserved Instance provides a capacity reservation. For more information, see [How Zonal Reserved Instances Are Applied \(p. 190\)](#).
- When you purchase a Reserved Instance for a Region, it's referred to as a *regional* Reserved Instance. A Regional Reserved Instance does *not* provide a capacity reservation. For more information, see [How Regional Reserved Instances Are Applied \(p. 190\)](#).

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 \(p. 189\)](#).

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 \(p. 221\)](#).

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 \(p. 203\)](#).

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.

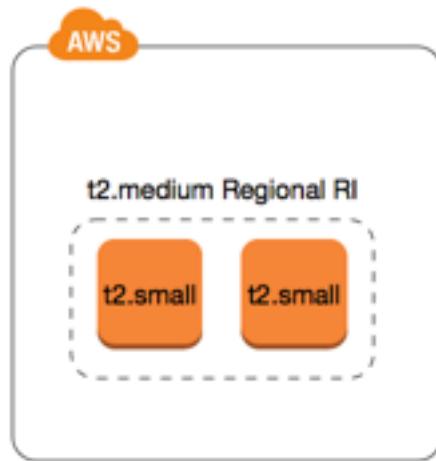
Instance size flexibility is applied from the smallest to the largest instance size within the instance family based on the normalization factor.

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.

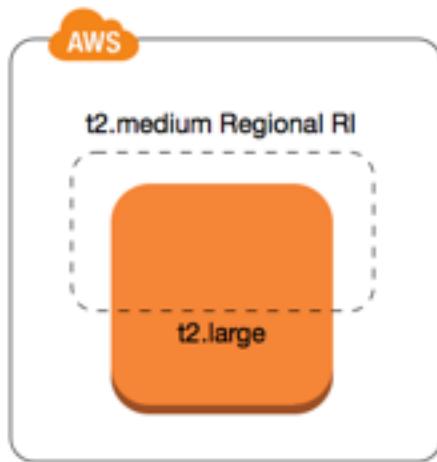
Instance size	Normalization factor
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8

Instance size	Normalization factor
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 \(p. 210\)](#).

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 \(p. 195\)](#). 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 \(p. 188\)](#) 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 \(p. 194\)](#)
- [Viewing Your Bill \(p. 195\)](#)
- [Reserved Instances and Consolidated Billing \(p. 195\)](#)
- [Reserved Instance Discount Pricing Tiers \(p. 195\)](#)

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 \(p. 309\)](#).

Reserved Instance billing benefits only apply to one instance-hour per clock-hour. An instance-hour begins when an instance is started and continues for 60 minutes or until the instance is stopped or terminated—whichever happens first.

A new instance-hour begins after an instance has run for 60 continuous minutes, or if an instance is stopped and then started. Rebooting an instance does not reset the running instance-hour.

For example, if an instance is stopped and then started again during a clock-hour and continues running for two more clock-hours, the first instance-hour (before the restart) is charged at the discounted Reserved Instance rate. The next instance-hour (after restart) is charged at the On-Demand rate and the next two instance-hours are charged at the discounted Reserved Instance rate.

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, SQL Server Web, and SQL Server Enterprise.
- Pricing tiers do not apply to Reserved Instances for Linux with SQL Server Standard, SQL Server Web, and SQL Server Enterprise.
- Pricing tier discounts only apply to purchases made from AWS. They do not apply to purchases of third-party Reserved Instances.
- Discount pricing tiers are currently not applicable to Convertible Reserved Instance purchases.

Topics

- [Calculating Reserved Instance Pricing Discounts \(p. 196\)](#)
- [Buying with a Discount Tier \(p. 197\)](#)
- [Crossing Pricing Tiers \(p. 197\)](#)
- [Consolidated Billing for Pricing Tiers \(p. 197\)](#)

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 \(p. 197\)](#).

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 \(p. 197\)](#).

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 \(p. 195\)](#).

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 \(p. 198\)](#)
- [Buying Standard Reserved Instances \(p. 198\)](#)
- [Buying Convertible Reserved Instances \(p. 201\)](#)
- [Viewing Your Reserved Instances \(p. 203\)](#)
- [Using Your Reserved Instances \(p. 203\)](#)

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 Windows with SQL Standard, Windows with SQL Server Enterprise, and Windows with SQL Server Web, you must choose offerings for those specific platforms. For all other Windows versions, choose an offering for the **Windows** 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 \(p. 203\)](#).

For examples of how Reserved Instances are applied to your running instances, see [How Reserved Instances Are Applied \(p. 189\)](#).

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",
    "InstanceTenancy": "default",
    "PricingDetails": [],
    "UsagePrice": 0.0,
    "RecurringCharges": [
        {
            "Amount": 0.0258,
            "Frequency": "Hourly"
        }
    ],
    "Marketplace": false,
    "CurrencyCode": "USD",
    "FixedPrice": 677.0,
    "Duration": 94608000,
    "Scope": "Region",
    "ReservedInstancesOfferingId": "13486b92-bdd6-4b68-894c-509bcf239ccd",
    "InstanceType": "t2.large"
},
{
    "OfferingClass": "convertible",
    "OfferingType": "All Upfront",
    "ProductDescription": "Linux/UNIX",
    "InstanceTenancy": "default",
    "PricingDetails": [],
    "UsagePrice": 0.0,
    "RecurringCharges": [],
    "Marketplace": false,
    "CurrencyCode": "USD",
    "FixedPrice": 1327.0,
    "Duration": 94608000,
    "Scope": "Region",
    "ReservedInstancesOfferingId": "e00ec34b-4674-4fb9-a0a9-213296ab93aa",
    "InstanceType": "t2.large"
}
]
```

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 \(p. 203\)](#).

For examples of how Reserved Instances are applied to your running instances, see [How Reserved Instances Are Applied \(p. 189\)](#).

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 \(p. 204\)](#). For more information, see [Reserved Instance Listing States \(p. 209\)](#).

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 \(p. 297\)](#).

For more information, see [Launching an Instance Using the Launch Instance Wizard \(p. 315\)](#). For examples of how Reserved Instances are applied to your running instances, see [How Reserved Instances Are Applied \(p. 189\)](#).

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 \(p. 204\)](#)
- [Buying in the Reserved Instance Marketplace \(p. 210\)](#)

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 \(p. 204\)](#)
- [Registering as a Seller \(p. 205\)](#)
- [Pricing Your Reserved Instances \(p. 207\)](#)
- [Listing Your Reserved Instances \(p. 208\)](#)
- [Lifecycle of a Listing \(p. 209\)](#)
- [After Your Reserved Instance Is Sold \(p. 210\)](#)

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 \(p. 205\)](#).

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 \(p. 207\)](#). You can also modify your Reserved Instances before listing them. For information, see [Modifying Reserved Instances \(p. 210\)](#).
- 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 \(p. 205\)](#).
- **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 \(p. 206\)](#).

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 \(p. 205\)](#)
- [Tax Information \(p. 206\)](#)
- [Sharing Information with the Buyer \(p. 206\)](#)
- [Getting Paid \(p. 207\)](#)

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 \(p. 208\)](#)
- [Listing Your Reserved Instances Using the AWS CLI or Amazon EC2 API \(p. 208\)](#)
- [Reserved Instance Listing States \(p. 209\)](#)

[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 **Status** information is about your reservation.

- **active**—The listing is available for purchase.
- **canceled**—The listing is canceled and isn't available for purchase in the Reserved Instance Marketplace.
- **closed**—The Reserved Instance is not listed. A Reserved Instance might be **closed** because the sale of the listing was completed.

Lifecycle of a Listing

When all the instances in your listing are matched and sold, the **My Listings** tab shows that the **Total instance count** matches the count listed under **Sold**. Also, there are no **Available** instances left for your listing, and its **Status** is **closed**.

When only a portion of your listing is sold, AWS retires the Reserved Instances in the listing and creates the number of Reserved Instances equal to the Reserved Instances remaining in the count. So, the listing ID and the listing that it represents, which now has fewer reservations for sale, is still active.

Any future sales of Reserved Instances in this listing are processed this way. When all the Reserved Instances in the listing are sold, AWS marks the listing as **closed**.

For example, you create a listing *Reserved Instances listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample* with a listing count of 5.

The **My Listings** tab in the **Reserved Instance** console page displays the listing this way:

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- Total reservation count = 5
- Sold = 0
- Available = 5
- Status = **active**

A buyer purchases two of the reservations, which leaves a count of three reservations still available for sale. Because of this partial sale, AWS creates a new reservation with a count of three to represent the remaining reservations that are still for sale.

This is how your listing looks in the **My Listings** tab:

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- Total reservation count = 5

- Sold = 2
- Available = 3
- Status = active

If you cancel your listing and a portion of that listing has already sold, the cancellation is not effective on the portion that has been sold. Only the unsold portion of the listing is no longer available in the Reserved Instance Marketplace.

After Your Reserved Instance Is Sold

When your Reserved Instance is sold, AWS sends you an email notification. Each day that there is any kind of activity, you receive one email notification capturing all the activities of the day. For example, you 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 \(p. 197\)](#).

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 \(p. 217\)](#).

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 \(p. 213\)](#).

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 \(p. 212\)](#)
- [Modifying the Instance Size of Your Reservations \(p. 213\)](#)
- [Submitting Modification Requests \(p. 215\)](#)
- [Troubleshooting Modification Requests \(p. 216\)](#)

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	Linux and Windows	-
Change the scope from Availability Zone to Region and vice versa	Linux and Windows	If you change the scope from Availability Zone to Region, you lose the capacity reservation benefit. If you change the scope from Region to Availability Zone, you lose Availability Zone flexibility and instance size flexibility (if applicable). For more information, see How Reserved Instances Are Applied (p. 189) .
Change the instance size within the same instance type	Linux only	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. 213) .
Change the network from EC2-Classic to Amazon VPC and vice versa	Linux and Windows	The network platform must be available in your AWS account. If you created your AWS account after 2013-12-04, it does not support EC2-Classic.

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 \(p. 213\)](#).

- 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.

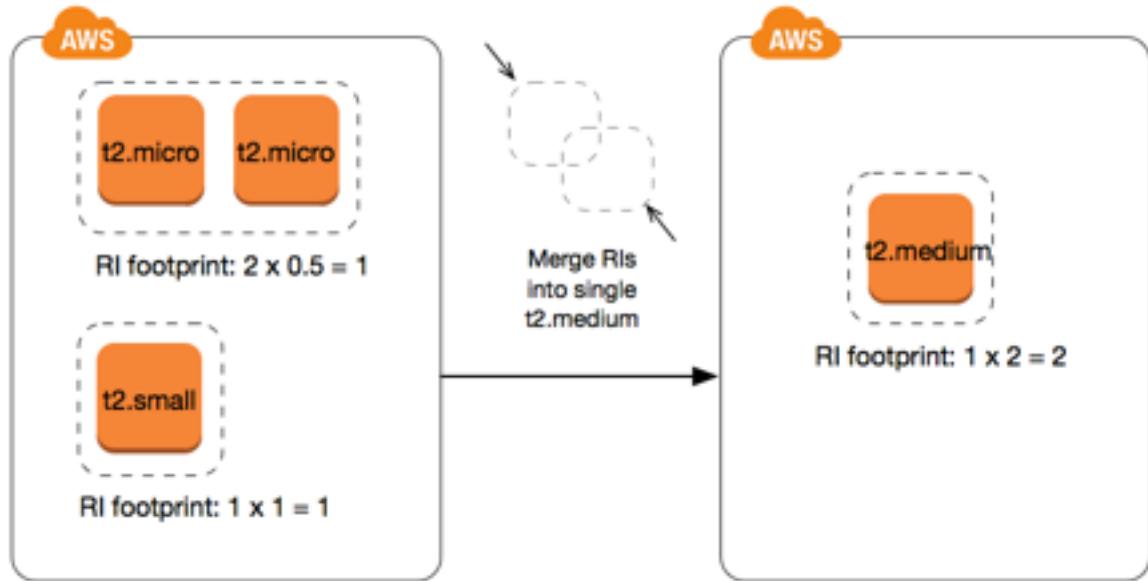
Instance size	Normalization factor
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
4xlarge	32
8xlarge	64
9xlarge	72
10xlarge	80

Instance size	Normalization factor
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256

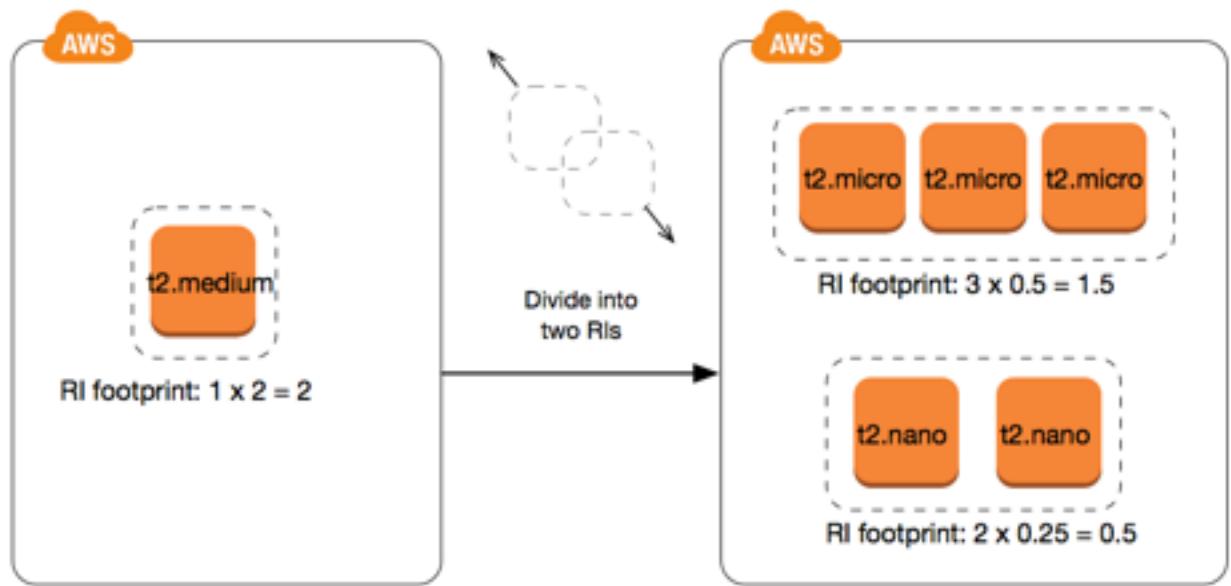
To calculate the instance size footprint of a Reserved Instance, multiply the number of instances by the normalization factor. For example, an `t2.medium` has a normalization factor of 2 so a reservation for four `t2.medium` instances has a footprint of 8 units.

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 \(p. 212\)](#). If you are modifying instance size, ensure that you've calculated the total [instance size footprint \(p. 213\)](#) 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:
 - **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 \(p. 212\)](#).
 - **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 \(p. 212\)](#) 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 \(p. 217\)](#)
- [Calculating Convertible Reserved Instances Exchanges \(p. 218\)](#)
- [Merging Convertible Reserved Instances \(p. 219\)](#)
- [Exchanging a Portion of a Convertible Reserved Instance \(p. 219\)](#)
- [Submitting Exchange Requests \(p. 220\)](#)

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 \(p. 219\)](#). For more information about modifying your Reserved Instances, see [Modifying Reserved Instances \(p. 210\)](#).
- 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 \(p. 219\)](#).

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.

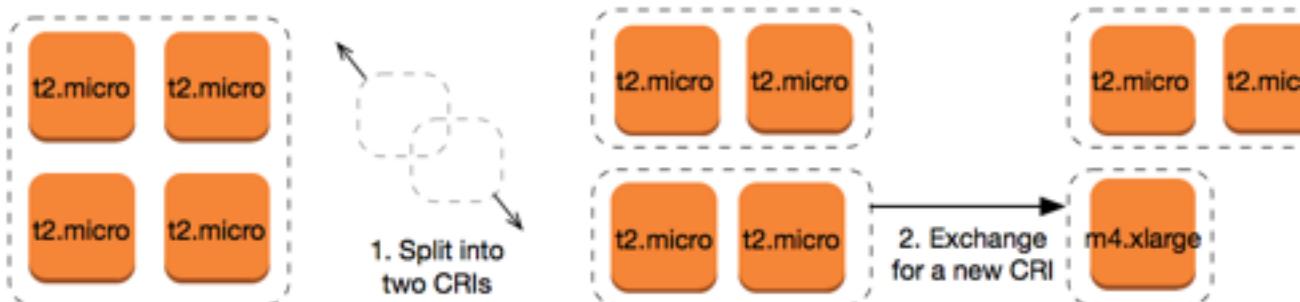
Exchanging a Portion of a Convertible Reserved Instance

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:

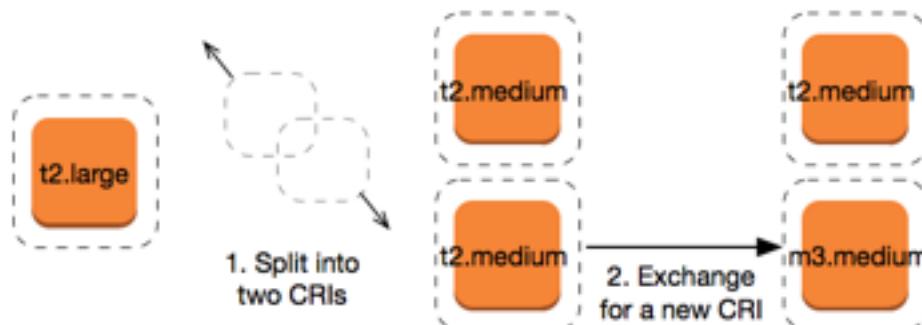
1. Modify the `t2.micro` Convertible Reserved Instance by splitting it into two `t2.micro` Convertible Reserved Instances with two instances each.
2. Exchange one of the new `t2.micro` Convertible Reserved Instances for an `m4.xlarge` Convertible Reserved Instance.



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 \(p. 213\)](#).
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 \(p. 213\)](#) and [Submitting Exchange Requests \(p. 220\)](#).

Not all Reserved Instances can be modified. Ensure that you read the applicable [restrictions \(p. 212\)](#).

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 \(p. 186\)](#). If you are flexible about when your instances run, Spot Instances might meet your needs and decrease costs. For more information, see [Spot Instances \(p. 225\)](#).

Contents

- [How Scheduled Instances Work \(p. 222\)](#)
- [Service-Linked Roles for Scheduled Instances \(p. 222\)](#)

- [Purchasing a Scheduled Instance \(p. 223\)](#)
- [Launching a Scheduled Instance \(p. 224\)](#)
- [Scheduled Instance Limits \(p. 224\)](#)

How Scheduled Instances Work

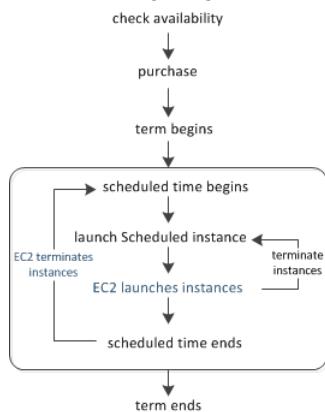
Amazon EC2 sets aside pools of EC2 instances in each Availability Zone for use as Scheduled Instances. Each pool supports a specific combination of instance type, operating system, and network.

To get started, you must search for an available schedule. You can search across multiple pools or a single pool. After you locate a suitable schedule, purchase it.

You must launch your Scheduled Instances during their scheduled time periods, using a launch configuration that matches the following attributes of the schedule that you purchased: instance type, Availability Zone, network, and platform. When you do so, Amazon EC2 launches EC2 instances on your behalf, based on the specified launch specification. Amazon EC2 must ensure that the EC2 instances have terminated by the end of the current scheduled time period so that the capacity is available for any other Scheduled Instances it is reserved for. Therefore, Amazon EC2 terminates the EC2 instances three minutes before the end of the current scheduled time period.

You can't stop or reboot Scheduled Instances, but you can terminate them manually as needed. If you terminate a Scheduled Instance before its current scheduled time period ends, you can launch it again after a few minutes. Otherwise, you must wait until the next scheduled time period.

The following diagram illustrates the lifecycle of a Scheduled Instance.



Service-Linked Roles for Scheduled Instances

Amazon EC2 creates a service-linked role when you purchase a Scheduled Instance. A service-linked role includes all the permissions that Amazon EC2 requires to call other AWS services on your behalf. For more information, see [Using Service-Linked Roles](#) in the *IAM User Guide*.

Amazon EC2 uses the service-linked role named **AWSServiceRoleForEC2ScheduledInstances** to complete the following actions:

- `ec2:TerminateInstances` – Terminate Scheduled Instances after their schedules complete
- `ec2:CreateTags` – Add system tags to Scheduled Instances

If you purchased Scheduled Instances before October 2017, when Amazon EC2 began supporting this service-linked role, Amazon EC2 created the **AWSServiceRoleForEC2ScheduledInstances** role in your AWS account. For more information, see [A New Role Appeared in My Account](#) in the *IAM User Guide*.

If you no longer need to use Scheduled Instances, we recommend that you delete the **AWSServiceRoleForEC2ScheduledInstances** role. After this role is deleted from your account, Amazon EC2 will create the role again if you purchase Scheduled Instances.

Purchasing a Scheduled Instance

To purchase a Scheduled Instance, you can use the Scheduled Reserved Instances Reservation Wizard.

Warning

After you purchase a Scheduled Instance, you can't cancel, modify, or resell your purchase.

To purchase a Scheduled Instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **INSTANCES**, choose **Scheduled Instances**. If the currently selected Region does not support Scheduled Instances, the page is unavailable. [Learn more \(p. 224\)](#)
3. Choose **Purchase Scheduled Instances**.
4. On the **Find available schedules** page, do the following:
 - a. Under **Create a schedule**, select the starting date from **Starting on**, the schedule recurrence (daily, weekly, or monthly) from **Recurring**, and the minimum duration from **for duration**. Note that the console ensures that you specify a value for the minimum duration that meets the minimum required utilization for your Scheduled Instance (1,200 hours per year).

Create a schedule

Starting on for duration 4 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 (AWS CLI)

Use the [describe-scheduled-instance-availability](#) command to list the available schedules that meet your needs, and then use the [purchase-scheduled-instances](#) command to complete the purchase.

Launching a Scheduled Instance

After you purchase a Scheduled Instance, it is available for you to launch during its scheduled time periods.

To launch a Scheduled Instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **INSTANCES**, choose **Scheduled Instances**. If the currently selected Region does not support Scheduled Instances, the page is unavailable. [Learn more \(p. 224\)](#)
3. Select the Scheduled Instance and choose **Launch Scheduled Instances**.
4. On the **Configure** page, complete the launch specification for your Scheduled Instances and choose **Review**.

Important

The launch specification must match the instance type, Availability Zone, network, and platform of the schedule that you purchased.

5. On the **Review** page, verify the launch configuration and modify it as needed. When you are finished, choose **Launch**.

To launch a Scheduled Instance (AWS CLI)

Use the [describe-scheduled-instances](#) command to list your Scheduled Instances, and then use the [run-scheduled-instances](#) command to launch each Scheduled Instance during its scheduled time periods.

Scheduled Instance Limits

Scheduled Instances are subject to the following limits:

- The following are the only supported instance types: C3, C4, 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.
- They are available in the following Regions: US East (N. Virginia), US West (Oregon), and Europe (Ireland).

Spot Instances

A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance is called a Spot price. The Spot price of each instance type in each Availability Zone is set by Amazon EC2, and adjusted gradually based on the long-term supply of and demand for Spot Instances. Your Spot Instance runs whenever capacity is available and the maximum price per hour for your request exceeds the Spot price.

Spot Instances are a cost-effective choice if you can be flexible about when your applications run and if your applications can be interrupted. For example, Spot Instances are well-suited for data analysis, batch jobs, background processing, and optional tasks. For more information, see [Amazon EC2 Spot Instances](#).

Topics

- [Concepts \(p. 225\)](#)
- [How to Get Started \(p. 226\)](#)
- [Related Services \(p. 227\)](#)
- [Pricing and Savings \(p. 227\)](#)

Concepts

Before you get started with Spot Instances, you should be familiar with the following concepts:

- **Spot Instance pool** – A set of unused EC2 instances with the same instance type (for example, `m5.large`), operating system, Availability Zone, and network platform.
- **Spot price** – The current price of a Spot Instance per hour.
- **Spot Instance request** – 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.

Key Differences between Spot Instances and On-Demand Instances

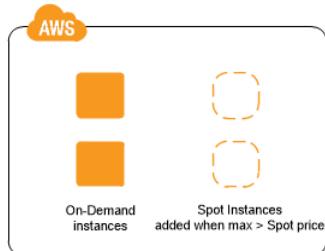
The following table lists the key differences between Spot Instances and On-Demand Instances.

	Spot Instances	On-Demand Instances
Launch time	Can only be launched immediately if the Spot Request is active and capacity is available.	Can only be launched immediately if you make a manual launch request and capacity is available.

	Spot Instances	On-Demand Instances
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. 276) 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).

Strategies for Using Spot Instances

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 \(p. 239\)](#).

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 \(p. 12\)](#)
- [Getting Started with Amazon EC2 Windows Instances \(p. 20\)](#)

Spot basics

- [How Spot Instances Work \(p. 228\)](#)
- [How Spot Fleet Works \(p. 229\)](#)

Working with Spot Instances

- [Preparing for Interruptions \(p. 278\)](#)
- [Creating a Spot Instance Request \(p. 241\)](#)
- [Getting Request Status Information \(p. 274\)](#)

Working with Spot Fleets

- [Spot Fleet Prerequisites \(p. 248\)](#)
- [Creating a Spot Fleet Request \(p. 251\)](#)

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 and Savings

You pay the Spot price for Spot Instances, which is set by Amazon EC2 and adjusted gradually based on the long-term supply of and demand for Spot Instances. If the maximum price for your request exceeds the current Spot price, Amazon EC2 fulfills your request if capacity is available. Your Spot Instances run until you terminate them, capacity is no longer available, 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 \(p. 235\)](#).

We independently map Availability Zones to codes for each AWS account. Therefore, you can get different results for the same Availability Zone code (for example, `us-west-2a`) between different accounts.

View Savings

You can view the savings made from using Spot Instances for a single Spot Fleet or for all Spot Instances. You can view the savings made in the last hour or the last three days, and you can view the average cost per vCPU hour and per memory (GiB) hour. Savings are estimated and may differ from actual savings because they do not include the billing adjustments for your usage. For more information about viewing savings information, see [Savings From Purchasing Spot Instances \(p. 236\)](#).

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 \(p. 276\)](#).

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 \(p. 228\)](#)
- [Launching Spot Instances in an Availability Zone Group \(p. 229\)](#)
- [Launching Spot Instances in a VPC \(p. 229\)](#)

Launching Spot Instances in a Launch Group

Specify a launch group in your Spot Instance request to tell Amazon EC2 to launch a set of Spot Instances only if it can launch them all. In addition, if the Spot service must terminate one of the

instances in a launch group (for example, if the Spot price exceeds your maximum price), it must terminate them all. However, if you terminate one or more of the instances in a launch group, Amazon EC2 does not terminate the remaining instances in the launch group.

Although this option can be useful, adding this constraint can decrease the chances that your Spot Instance request is fulfilled and increase the chances that your Spot Instances are terminated. For example, your launch group includes instances in multiple Availability Zones. If capacity in one of these Availability Zones decreases and is no longer available, then Amazon EC2 terminates all instances for the launch group.

If you create another successful Spot Instance request that specifies the same (existing) launch group as an earlier successful request, then the new instances are added to the launch group. Subsequently, if an instance in this launch group is terminated, all instances in the launch group are terminated, which includes instances launched by the first and second requests.

Launching Spot Instances in an Availability Zone Group

Specify an Availability Zone group in your Spot Instance request to tell the Spot service to launch a set of Spot Instances in the same Availability Zone. Amazon EC2 need not interrupt all instances in an Availability Zone group at the same time. If Amazon EC2 must interrupt one of the instances in an Availability Zone group, the others remain running.

Although this option can be useful, adding this constraint can lower the chances that your Spot Instance request is fulfilled.

If you specify an Availability Zone group but don't specify an Availability Zone in the Spot Instance request, the result depends on the network you specified.

Default VPC

Amazon EC2 uses the Availability Zone for the specified subnet. If you don't specify a subnet, it selects an Availability Zone and its default subnet, but not necessarily the lowest-priced zone. If you deleted the default subnet for an Availability Zone, then you must specify a different subnet.

Nondefault VPC

Amazon EC2 uses the Availability Zone for the specified subnet.

Launching Spot Instances in a VPC

You specify a subnet for your Spot Instances the same way that you specify a subnet for your On-Demand Instances.

- You should use the default maximum price (the On-Demand price), or base your maximum price on the Spot price history of Spot Instances in a VPC.
- [Default VPC] If you want your Spot Instance launched in a specific low-priced Availability Zone, you must specify the corresponding subnet in your Spot Instance request. If you do not specify a subnet, Amazon EC2 selects one for you, and the Availability Zone for this subnet might not have the lowest Spot price.
- [Nondefault VPC] You must specify the subnet for your Spot Instance.

How Spot Fleet Works

A *Spot Fleet* is a collection, or fleet, of Spot Instances, and optionally On-Demand Instances.

The Spot Fleet attempts to launch the number of Spot Instances and On-Demand Instances to meet the target capacity that you specified in the Spot Fleet request. The request for Spot Instances is fulfilled if 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 (for example, `m5.large`), operating system, Availability Zone, and network platform. When you make a Spot Fleet request, you can include multiple launch specifications, that vary by instance type, AMI, Availability Zone, or subnet. The Spot Fleet selects the Spot Instance pools that are used to fulfill the request, based on the launch specifications included in your Spot Fleet request, and the configuration of the Spot Fleet request. The Spot Instances come from the selected pools.

Contents

- [On-Demand in Spot Fleet \(p. 230\)](#)
- [Allocation Strategy for Spot Instances \(p. 230\)](#)
- [Spot Price Overrides \(p. 231\)](#)
- [Spot Fleet Instance Weighting \(p. 232\)](#)
- [Walkthrough: Using Spot Fleet with Instance Weighting \(p. 233\)](#)

On-Demand in Spot Fleet

To ensure that you always have instance capacity, you can include a request for On-Demand capacity in your Spot Fleet request. In your Spot Fleet request, you specify your desired target capacity and how much of that capacity must be On-Demand. The balance comprises Spot capacity, which is launched if there is available Amazon EC2 capacity and availability. For example, if in your Spot Fleet request you specify target capacity as 10 and On-Demand capacity as 8, Amazon EC2 launches 8 capacity units as On-Demand, and 2 capacity units ($10-8=2$) as Spot.

Prioritizing Instance Types for On-Demand Capacity

When Spot Fleet attempts to fulfill your On-Demand capacity, it defaults to launching the lowest-priced instance type first. If `OnDemandAllocationStrategy` is set to `prioritized`, Spot Fleet uses priority to determine which instance type to use first in fulfilling On-Demand capacity. The priority is assigned to the launch template override, and the highest priority is launched first.

For example, you have configured three launch template overrides, each with a different instance type: `c3.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`.

Allocation Strategy for Spot Instances

The allocation strategy for the Spot Instances in your Spot Fleet determines how it fulfills your Spot Fleet request from the possible Spot Instance pools represented by its launch specifications. The following are the allocation strategies that you can specify in your Spot Fleet request:

`lowestPrice`

The Spot Instances come from the pool with the lowest price. This is the default strategy.

`diversified`

The Spot Instances are distributed across all pools.

`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 may be interrupted is low, even with all the instances in a single Spot Instance pool. Therefore, the `lowestPrice` strategy is likely to meet your needs while providing the lowest cost.

If your fleet is large or runs for a long time, you can improve the availability of your fleet by distributing the Spot Instances across multiple pools. For example, if your Spot Fleet request specifies 10 pools and a target capacity of 100 instances, the fleet launches 10 Spot Instances in each pool. If the Spot price for one pool exceeds your maximum price for this pool, only 10% of your fleet is affected. Using this strategy also makes your fleet less sensitive to increases in the Spot price in any one pool over time.

With the `diversified` strategy, the Spot Fleet does not launch Spot Instances into any pools with a Spot price that is equal to or higher than the [On-Demand price](#).

To create a cheap and diversified fleet, use the `lowestPrice` strategy in combination with `InstancePoolsToUseCount`. You can use a low or high number of Spot pools across which to allocate your Spot Instances. For example, if you run batch processing, we recommend specifying a low number of Spot pools (for example, `InstancePoolsToUseCount=2`) to ensure that your queue always has compute capacity while maximizing savings. If you run a web service, we recommend specifying a high number of Spot pools (for example, `InstancePoolsToUseCount=10`) to minimize the impact if a Spot Instance pool becomes temporarily unavailable.

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 one instance in each of the three pools, and the fourth instance in whichever pool provides the lowest price per unit.

Walkthrough: Using Spot Fleet with Instance Weighting

This walkthrough uses a fictitious company called Example Corp to illustrate the process of requesting a Spot Fleet using instance weighting.

Objective

Example Corp, a pharmaceutical company, wants to leverage the computational power of Amazon EC2 for screening chemical compounds that might be used to fight cancer.

Planning

Example Corp first reviews [Spot Best Practices](#). Next, Example Corp determines the following requirements for their Spot Fleet.

Instance Types

Example Corp has a compute- and memory-intensive application that performs best with at least 60 GB of memory and eight virtual CPUs (vCPUs). They want to maximize these resources for the application at the lowest possible price. Example Corp decides that any of the following EC2 instance types would meet their needs:

Instance type	Memory (GiB)	vCPUs
<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 \(p. 248\)](#).

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 \(p. 246\)](#).

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).

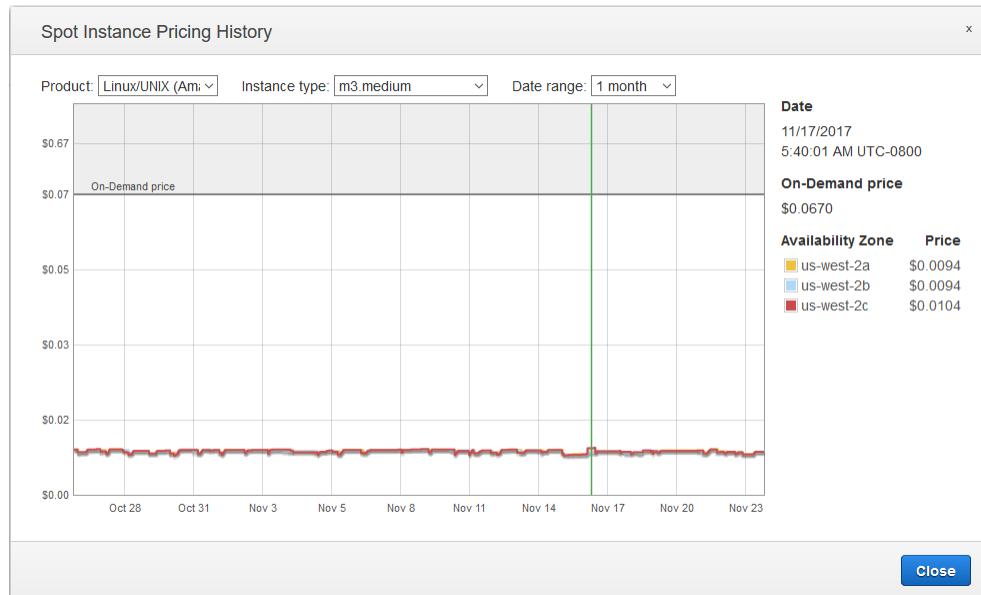
If Example Corp used the `diversified` strategy, the Spot Instances would come from all three pools. The Spot Fleet would launch 6 `r3.2xlarge` instances (which provide 6 units), 3 `r3.4xlarge` instances (which provide 6 units), and 2 `r3.8xlarge` instances (which provide 8 units), for a total of 20 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 (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 a 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

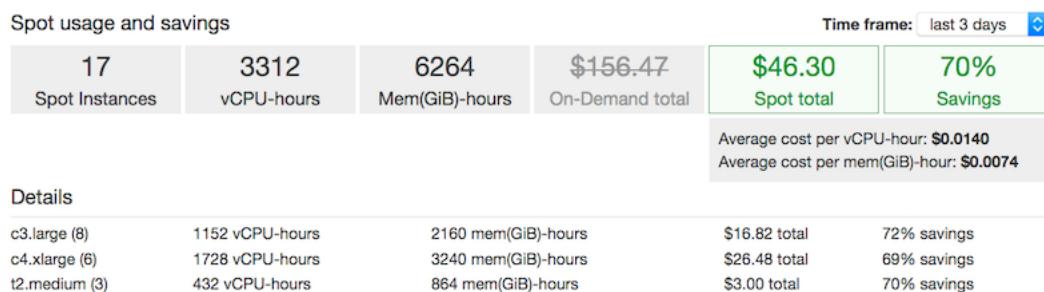
You can use one of the following commands. For more information, see [Accessing Amazon EC2 \(p. 3\)](#).

- [describe-spot-price-history](#) (AWS CLI)
- [Get-EC2SpotPriceHistory](#) (AWS Tools for Windows PowerShell)

Savings From Purchasing Spot Instances

You can view the usage and savings information for Spot Instances at the per-fleet level, or for all running Spot Instances. At the per-fleet level, the usage and savings information includes all instances launched and terminated by the fleet. You can view this information from the last hour or the last three days.

The following screenshot from the Spot Requests page shows the Spot usage and savings information for a Spot Fleet.



You can view the following usage and savings information:

- **Spot Instances** – The number of Spot Instances launched and terminated by the Spot Fleet. When viewing the savings summary, the number represents all your running Spot Instances.

- **vCPU-hours** – The number of vCPU hours used across all the Spot Instances for the selected time frame.
- **Mem(GiB)-hours** – The number of GiB hours used across all the Spot Instances for the selected time frame.
- **On-Demand total** – The total amount you would've paid for the selected time frame had you launched these instances as On-Demand Instances.
- **Spot total** – The total amount to pay for the selected time frame.
- **Savings** – The percentage that you are saving by not paying the On-Demand price.
- **Average cost per vCPU-hour** – The average hourly cost of using the vCPUs across all the Spot Instances for the selected time frame, calculated as follows: **Average cost per vCPU-hour = Spot total / vCPU-hours.**
- **Average cost per mem(GiB)-hour** – The average hourly cost of using the GiBs across all the Spot Instances for the selected time frame, calculated as follows: **Average cost per mem(GiB)-hour = Spot total / Mem(GiB)-hours.**
- **Details table** – The different instance types (the number of instances per instance type is in parentheses) that comprise the Spot Fleet. When viewing the savings summary, these comprise all your running Spot Instances.

Savings information can only be viewed using the Amazon EC2 console.

To view the savings information for a Spot Fleet (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, choose **Spot Requests**.
3. Select a Spot Fleet request and choose **Savings**.
4. By default, the page displays usage and savings information for the last three days. You can choose **last hour** or the **last three days**. For Spot Fleets that were launched less than an hour ago, the page shows the estimated savings for the hour.

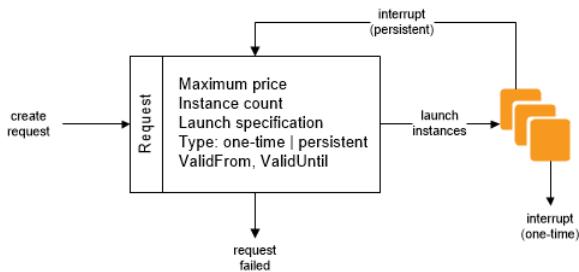
To view the savings information for all running Spot Instances (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, choose **Spot Requests**.
3. Choose **Savings Summary**.

Spot Instance Requests

To use Spot Instances, you create a Spot Instance request that includes the 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

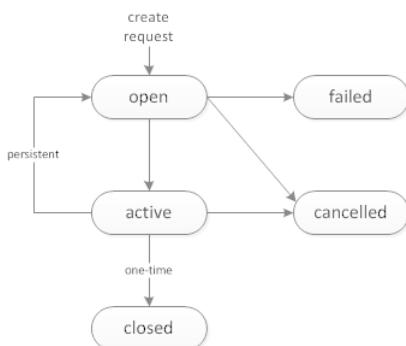
- [Spot Instance Request States \(p. 238\)](#)
- [Specifying a Duration for Your Spot Instances \(p. 239\)](#)
- [Specifying a Tenancy for Your Spot Instances \(p. 239\)](#)
- [Service-Linked Role for Spot Instance Requests \(p. 240\)](#)
- [Creating a Spot Instance Request \(p. 241\)](#)
- [Finding Running Spot Instances \(p. 243\)](#)
- [Tagging Spot Instance Requests \(p. 244\)](#)
- [Canceling a Spot Instance Request \(p. 244\)](#)
- [Terminating a Spot Instance \(p. 244\)](#)
- [Spot Request Example Launch Specifications \(p. 245\)](#)

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 your maximum price exceeds the Spot price or capacity becomes available again, the Spot Instance is started if stopped or resumed if hibernated. If the Spot Instance is terminated, the Spot Instance request is opened again and Amazon EC2 launches a new Spot Instance.

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 \(p. 270\)](#).

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 as a full hour.

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 (console)

Select the appropriate request type. For more information, see [Creating a Spot Instance Request \(p. 241\)](#).

To launch Spot Instances with a specified duration (AWS CLI)

To specify a duration for your Spot Instances, include the `--block-duration-minutes` option with the `request-spot-instances` command. For example, the following command creates a Spot request that launches Spot Instances that run for two hours:

```
aws ec2 request-spot-instances --instance-count 5 --block-duration-minutes 120 --type "one-time" --launch-specification file://specification.json
```

To retrieve the cost for Spot Instances with a specified duration (AWS CLI)

Use the `describe-spot-instance-requests` command to retrieve the fixed cost for your Spot Instances with a specified duration. The information is in the `actualBlockHourlyPrice` field.

Specifying a Tenancy for Your Spot Instances

You can run a Spot Instance on single-tenant hardware. Dedicated Spot Instances are physically isolated from instances that belong to other AWS accounts. For more information, see [Dedicated Instances \(p. 297\)](#) 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 \(p. 241\)](#).

- 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 \(p. 299\)](#). 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*.

Ensure that this role exists before you use the AWS CLI or an API to create a Spot Fleet. To create the role, use the IAM console as follows.

To create the IAM role (console)

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**.
3. Choose **Create role**.
4. On the **Select type of trusted entity** page, choose **EC2, EC2 - Spot Instances, Next: Permissions**.
5. On the next page, choose **Next:Review**.
6. On the **Review** page, choose **Create role**.

If you no longer need to use Spot Instances, we recommend that you delete the **AWSServiceRoleForEC2Spot** role. After this role is deleted from your account, Amazon EC2 will create the role again if you request Spot Instances.

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 Instance 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 Instance requests, see [Spot Request Status \(p. 270\)](#).

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 \(p. 235\)](#).

To create a Spot Instance request (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.

To use **Request and Maintain**, see [Creating a Spot Fleet Request \(p. 251\)](#).

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**, you can select an existing VPC or create a new one.

[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.

- 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.

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.

- f. (Optional) To add storage, specify additional instance store volumes or EBS volumes, depending on the instance type. You can also enable Amazon 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. To connect to your instances, 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 [Allocation Strategy for Spot Instances \(p. 230\)](#).
- 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 (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 \(p. 245\)](#). 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 remains running, depending on demand.

To find running Spot Instances (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.

You can see both Spot Instance requests and Spot Fleet requests. If a Spot Instance request has been fulfilled, **Capacity** is the ID of the Spot Instance. For a Spot Fleet, **Capacity** indicates how much of the requested capacity has been fulfilled. To view the IDs of the instances in a Spot Fleet, choose the expand arrow, or select the fleet and choose **Instances**.

Note

Spot Instance requests are not tagged instantly and for a period of time may appear separate from Spot Fleet Requests (SFR).

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 (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 \(p. 890\)](#).

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
```

Canceling 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. For more information about terminating a Spot Instance, see the next section.

To cancel a Spot Instance request (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests** and select the Spot request.
3. Choose **Actions, Cancel 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, and then choose **Actions, Instance State, Terminate**.

To cancel a Spot Instance request (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
```

Terminating a Spot Instance

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 you terminate

a running Spot Instance that was launched by a persistent Spot request, the Spot request returns to the open state so that a new Spot Instance can be launched. To cancel a persistent Spot request and terminate its Spot Instances, you must cancel the Spot request first and then terminate the Spot Instances. Otherwise, the persistent Spot request can launch a new instance. For more information about canceling a Spot Instance request, see the previous section.

To manually terminate a Spot Instance (AWS CLI)

- Use the following [terminate-instances](#) command to manually terminate Spot Instances:

```
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 \(p. 241\)](#).

1. [Launch Spot Instances \(p. 245\)](#)
2. [Launch Spot Instances in the specified Availability Zone \(p. 245\)](#)
3. [Launch Spot Instances in the specified subnet \(p. 246\)](#)
4. [Launch a Dedicated Spot Instance \(p. 246\)](#)

Example 1: Launch Spot Instances

The following example does not include an Availability Zone or subnet. Amazon EC2 selects an Availability Zone for you. Amazon EC2 launches the instances in the default subnet of the selected Availability Zone.

```
{  
    "ImageId": "ami-1a2b3c4d",  
    "KeyName": "my-key-pair",  
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],  
    "InstanceType": "m3.medium",  
    "IamInstanceProfile": {  
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"  
    }  
}
```

Example 2: Launch Spot Instances in the Specified Availability Zone

The following example includes an Availability Zone. Amazon EC2 launches the instances in the default subnet of the specified Availability Zone.

```
{  
    "ImageId": "ami-1a2b3c4d",  
    "KeyName": "my-key-pair",  
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],  
    "InstanceType": "m3.medium",  
    "Placement": {  
        "AvailabilityZone": "us-west-2a"  
    },  
    "IamInstanceProfile": {  
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"  
    }  
}
```

Example 3: Launch Spot Instances in the Specified Subnet

The following example includes a subnet. Amazon EC2 launches the instances in the specified subnet. If the VPC is a nondefault VPC, the instance does not receive a public IPv4 address by default.

```
{  
    "ImageId": "ami-1a2b3c4d",  
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],  
    "InstanceType": "m3.medium",  
    "SubnetId": "subnet-1a2b3c4d",  
    "IamInstanceProfile": {  
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"  
    }  
}
```

To assign a public IPv4 address to an instance in a nondefault VPC, specify the `AssociatePublicIpAddress` field as shown in the following example. When you specify a network interface, you must include the subnet ID and security group ID using the network interface, rather than using the `SubnetId` and `SecurityGroupIds` fields shown in example 3.

```
{  
    "ImageId": "ami-1a2b3c4d",  
    "KeyName": "my-key-pair",  
    "InstanceType": "m3.medium",  
    "NetworkInterfaces": [  
        {  
            "DeviceIndex": 0,  
            "SubnetId": "subnet-1a2b3c4d",  
            "Groups": [ "sg-1a2b3c4d" ],  
            "AssociatePublicIpAddress": true  
        }  
    ],  
    "IamInstanceProfile": {  
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"  
    }  
}
```

Example 4: Launch a Dedicated Spot Instance

The following example requests Spot Instance with a tenancy of dedicated. A Dedicated Spot Instance must be launched in a VPC.

```
{  
    "ImageId": "ami-1a2b3c4d",  
    "KeyName": "my-key-pair",  
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],  
    "InstanceType": "c3.8xlarge",  
    "SubnetId": "subnet-1a2b3c4d",  
    "Placement": {  
        "Tenancy": "dedicated"  
    }  
}
```

Spot Fleet Requests

To use a Spot Fleet, you create a Spot Fleet request that includes the target capacity, an optional On-Demand portion, one or more launch specifications for the instances, and the maximum price that you are willing to pay. Amazon EC2 attempts to maintain your Spot Fleet's target capacity as Spot prices change. For more information, see [How Spot Fleet Works \(p. 229\)](#).

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

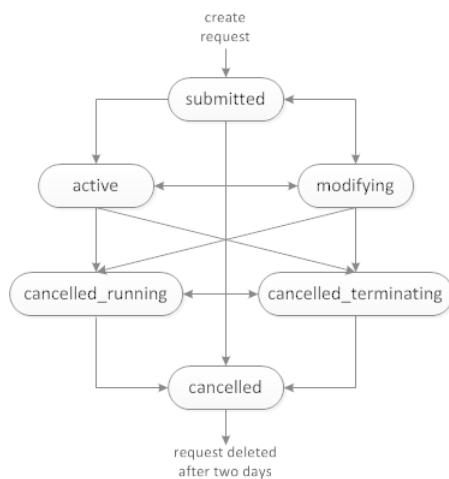
- [Spot Fleet Request States \(p. 247\)](#)
- [Spot Fleet Prerequisites \(p. 248\)](#)
- [Spot Fleet and IAM Users \(p. 248\)](#)
- [Spot Fleet Health Checks \(p. 249\)](#)
- [Planning a Spot Fleet Request \(p. 250\)](#)
- [Service-Linked Role for Spot Fleet Requests \(p. 250\)](#)
- [Creating a Spot Fleet Request \(p. 251\)](#)
- [Monitoring Your Spot Fleet \(p. 254\)](#)
- [Modifying a Spot Fleet Request \(p. 254\)](#)
- [Canceling a Spot Fleet Request \(p. 255\)](#)
- [Spot Fleet Example Configurations \(p. 256\)](#)

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 does not launch additional Spot Instances. Its existing Spot Instances continue to run until they are interrupted or terminated. The request remains in this state until all instances are interrupted or terminated.
- `cancelled_terminating` – The Spot Fleet is 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**, **Next: Permissions**.
 4. On the **Attached permissions policy** page, choose **Next:Review**.
 5. On the **Review** page, type a name for the role (for example, `aws-ec2-spot-fleet-tagging-role`) and choose **Create role**.

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 **JSON**, 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 choose **Create policy**.
5. In the navigation pane, choose **Users** and select the user.
6. Choose **Permissions, Add permissions**.
7. Choose **Attach existing policies directly**. Select the policy that you created earlier 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 \(p. 490\)](#).

You can configure your Spot Fleet to replace unhealthy instances. After enabling health check replacement, an instance is replaced after its health status is reported as unhealthy. The Spot Fleet could go below its target capacity for up to a few minutes while an unhealthy instance is being replaced.

Requirements

- Health check replacement is supported only with Spot Fleets that maintain a target capacity, not with one-time Spot Fleets.

- You can configure your Spot Fleet to replace unhealthy instances only when you create it.
- IAM users can use health check replacement only if they have permission to call the `ec2:DescribeInstanceStatus` action.

Planning a Spot Fleet Request

Before you create a Spot Fleet request, review [Spot Best Practices](#). Use these best practices when you plan your Spot Fleet request so that you can provision the type of instances you want at the lowest possible price. We also recommend that you do the following:

- Determine whether you want to create a Spot Fleet that submits a one-time request for the desired target capacity, or one that maintains a target capacity over time.
- Determine the instance types that meet your application requirements.
- Determine the target capacity for your Spot Fleet request. You can set the target capacity in instances or in custom units. For more information, see [Spot Fleet Instance Weighting \(p. 232\)](#).
- Determine what portion of the Spot Fleet target capacity must be On-Demand capacity. You can specify 0 for On-Demand capacity.
- Determine your price per unit, if you are using instance weighting. To calculate the price per unit, divide the price per instance hour by the number of units (or weight) that this instance represents. If you are not using instance weighting, the default price per unit is the price per instance hour.
- Review the possible options for your Spot Fleet request. For more information, see the [request-spot-fleet](#) command in the *AWS CLI Command Reference*. For additional examples, see [Spot Fleet Example Configurations \(p. 256\)](#).

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 \(p. 240\)](#).

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*.

Ensure that this role exists before you use the AWS CLI or an API to create a Spot Fleet. To create the role, use the IAM console as follows.

To create the IAM role for Spot Fleet (console)

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.

2. In the navigation pane, choose **Roles**.
3. Choose **Create role**.
4. On the **Select type of trusted entity** page, choose **EC2, EC2 - Spot Fleet, Next: Permissions**.
5. On the next page, choose **Next:Review**.
6. On the **Review** page, choose **Create role**.

If you no longer need to use Spot 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

Using the AWS Management Console, quickly create a Spot Fleet request by choosing only your application or task need and minimum compute specs. Amazon EC2 configures a fleet that best meets your needs and follows Spot best practice. For more information, see [Quickly Create a Spot Fleet Request \(Console\) \(p. 251\)](#). Otherwise, you can modify any of the default settings. For more information, see [Create a Spot Fleet Request Using Defined Parameters \(Console\) \(p. 251\)](#).

Quickly Create a Spot Fleet Request (Console)

Follow these steps to quickly create a Spot Fleet request.

To create a Spot Fleet request using the recommended settings (console)

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot>.
2. If you are new to Spot, you see a welcome page; choose **Get started**. Otherwise, choose **Request Spot Instances**.
3. For **Tell us your application or task need**, choose **Flexible workloads, Load balancing workloads, Big data workloads, or Defined duration workloads**.
4. Under **Configure your instances**, for **Minimum compute unit**, choose the minimum hardware specifications (vCPUs, memory, and storage) that you need for your application or task, either **as specs** or **as an instance type**.
 - For **as specs**, specify the required number of vCPUs and amount of memory.
 - For **as an instance type**, accept the default instance type, or choose **Change instance type** to choose a different instance type.
5. Under **Tell us how much capacity you need**, for **Total target capacity**, specify the number of units to request for target capacity. You can choose instances or vCPUs.
6. Review the recommended **Fleet request settings** based on your application or task selection, and choose **Launch**.

Create a Spot Fleet Request Using Defined Parameters (Console)

You can create a Spot Fleet using the parameters that you define.

To create a Spot Fleet request using defined parameters (console)

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot>.
2. If you are new to Spot, you see a welcome page; choose **Get started**. Otherwise, choose **Request Spot Instances**.
3. For **Tell us your application or task need**, choose **Flexible workloads, Load balancing workloads, Big data workloads, or Defined duration workloads**.
4. For **Configure your instances**, do the following:

- a. (Optional) For **Launch template**, choose a launch template. The launch template must specify an Amazon Machine Image (AMI), as you cannot override the AMI using Spot Fleet if you specify a launch template.

Important

If you intend to specify **Optional On-Demand portion**, you must choose a launch template.

- b. For **AMI**, choose one of the basic AMIs provided by AWS, or choose **Search for AMI** to use an AMI from our user community, the AWS Marketplace, or one of your own.
- c. For **Minimum compute unit**, choose the minimum hardware specifications (vCPUs, memory, and storage) that you need for your application or task, either **as specs** or **as an instance type**.
 - For **as specs**, specify the required number of vCPUs and amount of memory.
 - For **as an instance type**, accept the default instance type, or choose **Change instance type** to choose a different instance type.
- d. (Optional) For **Network**, choose an existing VPC or create a new one.

[Existing VPC] Choose the VPC.

[New VPC] Choose **Create new VPC** to go the Amazon VPC console. When you are done, return to the wizard and refresh the list.

- e. (Optional) For **Availability Zone**, let AWS choose the Availability Zones for your Spot Instances, or specify one or more Availability Zones.

If you have more than one subnet in an Availability Zone, choose the appropriate subnet from **Subnet**. To add subnets, choose **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and refresh the list.

- f. (Optional) For **Key pair name**, choose an existing key pair or create a new one.

[Existing key pair] Choose the key pair.

[New key pair] Choose **Create new key pair** to go the Amazon VPC console. When you are done, return to the wizard and refresh the list.

5. (Optional) For **Additional configurations**, do the following:

- a. (Optional) To add storage, specify additional instance store volumes or Amazon EBS volumes, depending on the instance type.
- b. (Optional) To enable Amazon EBS optimization, for **EBS-optimized**, choose **Launch EBS-optimized instances**.
- c. (Optional) To add temporary block-level storage for your instances, for **Instance store**, choose **Attach at launch**.
- d. (Optional) By default, basic monitoring is enabled for your instances. To enable detailed monitoring, for **Monitoring**, choose **Enable CloudWatch detailed monitoring**.
- e. (Optional) To run a Dedicated Spot Instance, for **Tenancy**, choose **Dedicated - run a dedicated instance**.
- f. (Optional) For **Security groups**, choose one or more security groups or create a new one.

[Existing security group] Choose one or more security groups.

[New security group] Choose **Create new security group** to go the Amazon VPC console. When you are done, return to the wizard and refresh the list.

- g. (Optional) To make your instances reachable from the internet, for **Auto-assign IPv4 Public IP**, choose **Enable**.
- h. (Optional) To launch your Spot Instances with an IAM role, for **IAM instance profile**, choose the role .

- i. (Optional) To run a start-up script, copy it to **User data**.
 - j. (Optional) To add a tag, choose **Add new tag** and enter the key and value for the tag. Repeat for each tag.
6. For **Tell us how much capacity you need**, do the following:
- a. For **Total target capacity**, specify the number of units to request for target capacity. You can choose instances or vCPUs. To specify a target capacity of 0 so that you can add capacity later, choose **Maintain target capacity**.
 - b. (Optional) For **Optional On-Demand portion**, specify the number of On-Demand units to request. The number must be less than the **Total target capacity**. Amazon EC2 calculates the difference, and allocates the difference to Spot units to request.
- Important**
To specify an optional On-Demand portion, you must first choose a launch template.
- c. (Optional) To replace unhealthy instances in a **Request and Maintain** Spot Fleet, select **Replace unhealthy instances**.
 - d. (Optional) By default, the Spot service terminates Spot Instances when they are interrupted. To maintain the target capacity, choose **Maintain target capacity**. You can then specify that the Spot service terminates, stops, or hibernates Spot Instances when they are interrupted. To do so, choose the corresponding option from **Interruption behavior**.
7. For **Fleet request settings**, do the following:
- a. Review the fleet request and fleet allocation strategy based on your application or task selection. To change the instance types or allocation strategy, clear **Apply recommendations**.
 - b. (Optional) To remove instance types, for **Fleet request**, choose **Remove**. To add instance types, choose **Select instance types**.
 - c. (Optional) For **Fleet allocation strategy**, choose the strategy that meets your needs. For more information, see [Allocation Strategy for Spot Instances \(p. 230\)](#).
8. For **Additional request details**, do the following:
- a. Review the additional request details. To make changes, clear **Apply defaults**.
 - b. (Optional) For **IAM fleet role**, you can use the default role or choose a different role. To use the default role after changing the role, choose **Use default role**.
 - c. (Optional) For **Maximum price**, you can use the default maximum price (the On-Demand price) or specify the maximum price you are willing to pay. If your maximum price is lower than the Spot price for the instance types that you selected, your Spot Instances are not launched.
 - d. (Optional) To create a request that is valid only during a specific time period, edit **Request valid from** and **Request valid until**.
 - e. (Optional) By default, we terminate your Spot Instances when the request expires. To keep them running after your request expires, clear **Terminate the instances when the request expires**.
 - f. (Optional) To register your Spot Instances with a load balancer, choose **Receive traffic from one or more load balancers** and choose one or more Classic Load Balancers or target groups.
9. (Optional) To download a copy of the launch configuration for use with the AWS CLI, choose **JSON config**.
10. Choose **Launch**.

The Spot Fleet request type is `fleet`. When the request is fulfilled, requests of type `instance` are added, where the state is `active` and the status is `fulfilled`.

To create a Spot Fleet request using the AWS CLI

- Use the 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 \(p. 256\)](#).

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 (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request. To see the configuration details, choose **Description**.
4. To list the Spot Instances for the Spot Fleet, choose **Instances**.
5. To view the history for the Spot Fleet, choose **History**.

To monitor your Spot Fleet (AWS CLI)

Use the 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 (console)

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot/home/fleet>.
2. Select your Spot Fleet request.
3. Choose **Actions, Modify target capacity**.
4. In **Modify target capacity**, do the following:
 - a. Enter the new target capacity.
 - b. (Optional) If you are decreasing the target capacity but want to keep the fleet at its current size, clear **Terminate instances**.
 - c. Choose **Submit**.

To modify a Spot Fleet request using the AWS CLI

Use the 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 (console)

1. Open the Spot console at <https://console.aws.amazon.com/ec2spot/home/fleet>.
2. Select your Spot Fleet request.
3. Choose **Actions, Cancel spot request**.
4. In **Cancel spot request**, verify that you want to cancel the Spot Fleet. To keep the fleet at its current size, clear **Terminate instances**. When you are ready, choose **Confirm**.

To cancel a Spot Fleet request using the AWS CLI

Use the 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-73fdbd2ce-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-73fdbd2ce-  
aa30-494c-8788-1cee4EXAMPLE --no-terminate-instances
```

The following is example output:

```
{  
    "SuccessfulFleetRequests": [  
        {  
            "SpotFleetRequestId": "sfr-73fdbd2ce-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 \(p. 251\)](#).

1. Launch Spot Instances using the lowest-priced Availability Zone or subnet in the region (p. 257)

2. Launch Spot Instances using the lowest-priced Availability Zone or subnet in a specified list (p. 257)
3. Launch Spot Instances using the lowest-priced instance type in a specified list (p. 259)
4. Override the price for the request (p. 260)
5. Launch a Spot Fleet using the diversified allocation strategy (p. 261)
6. Launch a Spot Fleet using instance weighting (p. 262)
7. Launch a Spot Fleet with On-Demand capacity (p. 263)

Example 1: Launch Spot Instances Using the Lowest-Priced Availability Zone or Subnet in the Region

The following example specifies a single launch specification without an Availability Zone or subnet. The Spot Fleet launches the instances in the lowest-priced Availability Zone that has a default subnet. The price you pay does not exceed the On-Demand price.

```
{  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "KeyName": "my-key-pair",  
            "SecurityGroups": [  
                {  
                    "GroupId": "sg-1a2b3c4d"  
                }  
            ],  
            "InstanceType": "m3.medium",  
            "IamInstanceProfile": {  
                "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"  
            }  
        }  
    ]  
}
```

Example 2: Launch Spot Instances Using the Lowest-Priced Availability Zone or Subnet in a Specified List

The following examples specify two launch specifications with different Availability Zones or subnets, but the same instance type and AMI.

Availability Zones

The Spot Fleet launches the instances in the default subnet of the lowest-priced Availability Zone that you specified.

```
{  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "KeyName": "my-key-pair",  
            "SecurityGroups": [  
                {  
                    "GroupId": "sg-1a2b3c4d"  
                }  
            ],  
            "InstanceType": "m3.medium",  
            "Placement": {  
                "AvailabilityZone": "us-west-2a"  
            }  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "KeyName": "my-key-pair",  
            "SecurityGroups": [  
                {  
                    "GroupId": "sg-1a2b3c4d"  
                }  
            ],  
            "InstanceType": "m3.medium",  
            "Placement": {  
                "AvailabilityZone": "us-west-2b"  
            }  
        }  
    ]  
}
```

```
        "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": "cc2.8xlarge",  
            "SubnetId": "subnet-1a2b3c4d"  
        }  
    ]  
}
```

```
        },
        ],
        "InstanceType": "r3.8xlarge",
        "SubnetId": "subnet-1a2b3c4d"
    }
}
```

Example 4. Override the Price for the Request

We recommend that you use the default maximum price, which is the On-Demand price. If you prefer, you can specify a maximum price for the fleet request and maximum prices for individual launch specifications.

The following examples specify a maximum price for the fleet request and maximum prices for two of the three launch specifications. The maximum price for the fleet request is used for any launch specification that does not specify a maximum price. The Spot Fleet launches the instances using the instance type with the lowest price.

Availability Zone

```
{
    "SpotPrice": "1.00",
    "TargetCapacity": 30,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            },
            "SpotPrice": "0.10"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.4xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            },
            "SpotPrice": "0.20"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.8xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        }
    ]
}
```

Subnet

```
{
    "SpotPrice": "1.00",
    "TargetCapacity": 30,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.2xlarge",

```

```
        "SubnetId": "subnet-1a2b3c4d",
        "SpotPrice": "0.10"
    },
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "c3.4xlarge",
        "SubnetId": "subnet-1a2b3c4d",
        "SpotPrice": "0.20"
    },
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "c3.8xlarge",
        "SubnetId": "subnet-1a2b3c4d"
    }
]
```

Example 5: Launch a Spot Fleet Using the Diversified Allocation Strategy

The following example uses the diversified allocation strategy. The launch specifications have different instance types but the same AMI and Availability Zone or subnet. The Spot Fleet distributes the 30 instances across the three launch specifications, such that there are 10 instances of each type. For more information, see [Allocation Strategy for Spot Instances \(p. 230\)](#).

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

Availability Zone

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            },
            "WeightedCapacity": 6
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            },
            "WeightedCapacity": 3
        }
    ]
}
```

Subnet

```
{  
    "SpotPrice": "0.70",  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.2xlarge",  
            "SubnetId": "subnet-1a2b3c4d",  
            "WeightedCapacity": 6  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c3.xlarge",  
            "SubnetId": "subnet-1a2b3c4d",  
            "WeightedCapacity": 3  
        }  
    ]  
}
```

Example 7: Launch a Spot Fleet with On-Demand Capacity

To ensure that you always have instance capacity, you can include a request for On-Demand capacity in your Spot Fleet request. If there is capacity, the On-Demand request is always fulfilled. The balance of the target capacity is fulfilled as Spot if there is capacity and availability.

The following example specifies the desired target capacity as 10, of which 5 must be On-Demand capacity. Spot capacity is not specified; it is implied in the balance of the target capacity minus the On-Demand capacity. Amazon EC2 launches 5 capacity units as On-Demand, and 5 capacity units ($10-5=5$) as Spot if there is available Amazon EC2 capacity and availability.

For more information, see [On-Demand in Spot Fleet \(p. 230\)](#).

```
{  
    "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 \(p. 499\)](#).

For more information about CloudWatch metrics provided by Amazon EC2, see [Monitoring Your Instances Using CloudWatch \(p. 498\)](#).

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 \(p. 500\)](#).

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 does 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 Fleet 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> . Units: Percent
<code>TargetCapacity</code>	The target capacity of the Spot Fleet request. Units: Count

Metric	Description
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, 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.

FleetRequestId	Metric Name
sfr-4a707781-8fac-459b-a5ae-4701fce47d7	AvailableInstancePoolsCount
sfr-4a707781-8fac-459b-a5ae-4701fce47d7	BidsSubmittedForCapacity
<input checked="" type="checkbox"/> sfr-4a707781-8fac-459b-a5ae-4701fce47d7	CPUUtilization
<input type="checkbox"/> sfr-4a707781-8fac-459b-a5ae-4701fce47d7	DiskReadBytes

Automatic Scaling for Spot Fleet

Automatic scaling is the ability to increase or decrease the target capacity of your Spot Fleet automatically based on demand. A Spot Fleet can either launch instances (scale out) or terminate instances (scale in), within the range that you choose, in response to one or more scaling policies.

If you are using instance weighting, keep in mind that Spot Fleet can exceed the target capacity as needed. Fulfilled capacity can be a floating-point number but target capacity must be an integer, so Spot Fleet rounds up to the next integer. You must take these behaviors into account when you look at the outcome of a scaling policy when an alarm is triggered. For example, suppose that the target capacity is 30, the fulfilled capacity is 30.1, and the scaling policy subtracts 1. When the alarm is triggered, the automatic scaling process subtracts 1 from 30.1 to get 29.1 and then rounds it up to 30, so no scaling action is taken. As another example, suppose that you selected instance weights of 2, 4, and 8, and a target capacity of 10, but no weight 2 instances were available so Spot Fleet provisioned instances of weights 4 and 8 for a fulfilled capacity of 12. If the scaling policy decreases target capacity by 20% and an alarm is triggered, the automatic scaling process subtracts $12 * 0.2$ from 12 to get 9.6 and then rounds it up to 10, so no scaling action is taken.

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, automatic scaling scales out your scalable target immediately.

Spot Fleet supports the following types of automatic scaling:

- [Target tracking scaling \(p. 266\)](#)—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 \(p. 267\)](#)—Increase or decrease the current capacity of the fleet based on a set of scaling adjustments, known as step adjustments, that vary based on the size of the alarm breach.
- [Scheduled scaling \(p. 269\)](#)—Increase or decrease the current capacity of the fleet based on the date and time.

Scale Spot Fleet Using a Target Tracking Policy

With target tracking scaling policies, you select a metric and set a target value. Spot Fleet creates and manages the CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value. The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value. In addition to keeping the metric close to the target value, a target tracking scaling policy also adjusts to the fluctuations in the metric due to a fluctuating load pattern and minimizes rapid fluctuations in the capacity of the fleet.

You can create multiple target tracking scaling policies for a Spot Fleet, provided that each of them uses a different metric. The fleet scales based on the policy that provides the largest fleet capacity. This enables you to cover multiple scenarios and ensure that there is always enough capacity to process your application workloads.

To ensure application availability, the fleet scales out proportionally to the metric as fast as it can, but scales in more gradually.

When a Spot Fleet terminates an instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

Do not edit or delete the CloudWatch alarms that Spot Fleet manages for a target tracking scaling policy. Spot Fleet deletes the alarms automatically when you delete the target tracking scaling policy.

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 (console)

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, choose **Spot Requests**.
- Select your Spot Fleet request and choose **Auto Scaling**.
- If automatic scaling is not configured, choose **Configure**.
- Use **Scale capacity between** to set the minimum and maximum capacity for your fleet. Automatic scaling does not scale your fleet below the minimum capacity or above the maximum capacity.
- For **Policy name**, type a name for the policy.
- Choose a **Target metric**.
- 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 automatic scaling process calculates the new target capacity using the fulfilled capacity and the scaling policy, and then updates the target capacity accordingly. For example, suppose that the target capacity and fulfilled capacity are 10 and the scaling policy adds 1. When the alarm is triggered, the automatic scaling process adds 1 to 10 to get 11, so Spot Fleet launches 1 instance.

When a Spot Fleet terminates an instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

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

- Open the IAM console at <https://console.aws.amazon.com/iam/>.
- In the navigation pane, choose **Roles**.
- Choose **Create role**.
- On the **Select type of trusted entity** page, choose **AWS service, EC2, EC2 - Spot Fleet Auto Scaling**, and then choose **Next: Permissions**.
- On the **Attached permissions policy** page, choose **Next:Review**.
- On the **Review** page, type a name for the role and choose **Create role**.

To create a CloudWatch alarm

- Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
- In the navigation pane, choose **Alarms**.
- Choose **Create Alarm**.
- For **CloudWatch Metrics by Category**, choose a category. For example, choose **EC2 Spot Metrics, Fleet Request Metrics**.
- Select a metric and choose **Next**.
- For **Alarm Threshold**, type a name and description for the alarm, and set the threshold value and number of time periods for the alarm.
- (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.
- Choose **Create Alarm**.

To configure step scaling policies for your Spot Fleet (console)

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, choose **Spot Requests**.
- Select your Spot Fleet request and choose **Auto Scaling**.
- If automatic scaling is not configured, choose **Configure**.
- Use **Scale capacity between** to set the minimum and maximum capacity for your fleet. Automatic scaling does not scale your fleet below the minimum capacity or above the maximum capacity.
- Initially, **Scaling policies** contains policies named `ScaleUp` and `ScaleDown`. You can complete these policies, or choose **Remove policy** to delete them. You can also choose **Add policy**.

7. To define a policy, do the following:
 - a. For **Policy name**, type a name for the policy.
 - b. For **Policy trigger**, select an existing alarm or choose **Create new alarm** to open the Amazon CloudWatch console and create an alarm.
 - c. For **Modify capacity**, select a scaling adjustment type, select a number, and select a unit.
 - d. (Optional) To perform step scaling, choose **Define steps**. By default, an add policy has a lower bound of -infinity and an upper bound of the alarm threshold. By default, a remove policy has a lower bound of the alarm threshold and an upper bound of +infinity. To add another step, choose **Add step**.
 - e. (Optional) To modify the default value for the cooldown period, select a number from **Cooldown period**.
8. Choose **Save**.

To configure step scaling policies for your Spot Fleet using the AWS CLI

1. Register the Spot Fleet request as a scalable target using the [register-scalable-target](#) command.
2. Create a scaling policy using the [put-scaling-policy](#) command.
3. Create an alarm that triggers the scaling policy using the [put-metric-alarm](#) command.

Scale Spot Fleet Using Scheduled Scaling

Scaling based on a schedule enables you to scale your application in response to predictable changes in demand. To use scheduled scaling, you create *scheduled actions*, which tell Spot Fleet to perform scaling activities at specific times. When you create a scheduled action, you specify the Spot Fleet, when the scaling activity should occur, minimum capacity, and maximum capacity. You can create scheduled actions that scale one time only or that scale on a recurring schedule.

To create a one-time scheduled action

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request and choose **Scheduled Scaling**.
4. Choose **Create Scheduled Action**.
5. For **Name**, specify a name for the scheduled action.
6. Type a value for **Minimum capacity**, **Maximum capacity**, or both.
7. For **Recurrence**, choose **Once**.
8. (Optional) Choose a date and time for **Start time**, **End time**, or both.
9. Choose **Submit**.

To scale on a recurring schedule

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request and choose **Scheduled Scaling**.
4. For **Recurrence**, choose one of the predefined schedules (for example, **Every day**), or choose **Custom** and type a cron expression. For more information about the cron expressions supported by scheduled scaling, see [Cron Expressions](#) in the *Amazon CloudWatch Events User Guide*.
5. (Optional) Choose a date and time for **Start time**, **End time**, or both.
6. Choose **Submit**.

To edit a scheduled action

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request and choose **Scheduled Scaling**.
4. Select the scheduled action and choose **Actions, Edit**.
5. Make the needed changes and choose **Submit**.

To delete a scheduled action

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request and choose **Scheduled Scaling**.
4. Select the scheduled action and choose **Actions, Delete**.
5. When prompted for confirmation, choose **Delete**.

To manage scheduled scaling using the AWS CLI

Use the following commands:

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Spot Request Status

To help you track your Spot Instance requests and plan your use of Spot Instances, use the request status provided by Amazon EC2. For example, the request status can provide the reason why your Spot request isn't fulfilled yet, or list the constraints that are preventing the fulfillment of your Spot request.

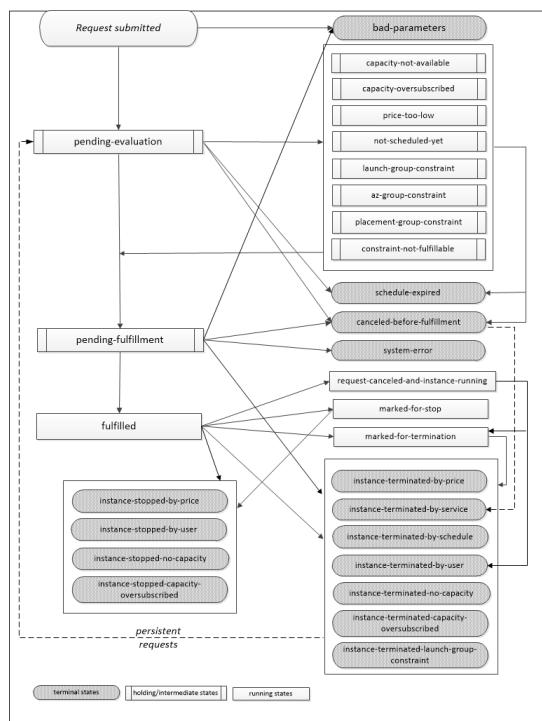
At each step of the process—also called the Spot request *lifecycle*, specific events determine successive request states.

Contents

- [Life Cycle of a Spot Request \(p. 270\)](#)
- [Getting Request Status Information \(p. 274\)](#)
- [Spot Request Status Codes \(p. 274\)](#)

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 are not valid (bad-parameters).

Status Code	Request State	Instance State
pending-evaluation	open	n/a
bad-parameters	closed	n/a

Holding

If one or more request constraints are valid but can't be met yet, or if there is not enough capacity, the request goes into a holding state waiting for the constraints to be met. The request options affect the likelihood of the request being fulfilled. For example, if you specify a maximum price below the current Spot price, your request stays in a holding state until the Spot price goes below your maximum price. If you specify an Availability Zone group, the request stays in a holding state until the Availability Zone constraint is met.

Status Code	Request State	Instance State
capacity-not-available	open	n/a
capacity-oversubscribed	open	n/a
price-too-low	open	n/a
not-scheduled-yet	open	n/a
launch-group-constraint	open	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. It might also happen if you cancel the request, or if a system error occurs.

Status Code	Request State	Instance State
schedule-expired	cancelled	n/a
canceled-before-fulfillment*	cancelled	n/a
bad-parameters	failed	n/a
system-error	closed	n/a

* If you cancel the request.

Pending fulfillment

When the constraints you specified (if any) are met and your maximum price is equal to or higher than the current Spot price, your Spot request goes into the pending-fulfillment state.

At this point, Amazon EC2 is getting ready to provision the instances that you requested. If the process stops at this point, it is likely to be because it was cancelled by the user before a Spot Instance was launched. It may also be because an unexpected system error occurred.

Status Code	Request State	Instance State
pending-fulfillment	open	n/a

Fulfilled

When all the specifications for your Spot Instances are met, your Spot request is fulfilled. Amazon EC2 launches the Spot Instances, which can take a few minutes. If a Spot Instance is hibernated or stopped when interrupted, it remains in this state until the request can be fulfilled again or the request is cancelled.

Status Code	Request State	Instance State
fulfilled	active	pending → running
fulfilled	active	stopped → running

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. Even if you terminate a Spot Instance before you cancel its request, there might be a delay before Amazon EC2 detects that your Spot Instance was terminated. In this case, the request state can either be `closed` or `cancelled`.

Persistent requests

When your Spot Instances are terminated (either by you or Amazon EC2), if the Spot request is a persistent request, it returns to the pending-evaluation state and then Amazon EC2 can launch a new Spot Instance when the constraints are met.

Getting Request Status Information

You can get request status information using the AWS Management Console or a command line tool.

To get request status information (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Spot Requests** and select the Spot request.
3. To check the status, choose **Description, Status**.

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. 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. 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 is not evaluated until the scheduled date.

pending-evaluation

After you make a Spot Instance request, it goes into the pending-evaluation state while the system evaluates the parameters of your request.

pending-fulfillment

Amazon EC2 is trying to provision your Spot Instances.

placement-group-constraint

The Spot request can't be fulfilled yet because a Spot Instance can't be added to the placement group at this time.

price-too-low

The request can't be fulfilled yet because your maximum price is below the Spot price. In this case, no instance is launched and your request remains open.

request-canceled-and-instance-running

You canceled the Spot request while the Spot Instances are still running. The request is cancelled, but the instances remain running.

schedule-expired

The Spot request expired because it was not fulfilled before the specified date.

system-error

There was an unexpected system error. If this is a recurring issue, please contact AWS Support for assistance.

Spot Instance Interruptions

Demand for Spot Instances can vary significantly from moment to moment, and the availability of Spot Instances can also vary significantly depending on how many unused EC2 instances are available. It is always possible that your Spot Instance might be interrupted. Therefore, you must ensure that your application is prepared for a Spot Instance interruption.

The following are the possible reasons that Amazon EC2 might interrupt your Spot Instances:

- Price – The Spot price is greater than your maximum price.
- Capacity – If there are not enough unused EC2 instances to meet the demand for 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 \(p. 815\)](#) and [Copying an AMI \(p. 75\)](#).

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 \(p. 279\)](#).

Preparing for Interruptions

Here are some best practices to follow when you use Spot Instances:

- Use the default maximum price, which is the On-Demand price.
- Ensure that your instance is ready to go as soon as the request is fulfilled by using an Amazon Machine Image (AMI) that contains the required software configuration. You can also use user data to run commands at start-up.
- Store important data regularly in a place that isn't affected when the Spot Instance terminates. For example, you can use Amazon S3, Amazon EBS, or DynamoDB.
- Divide the work into small tasks (using a Grid, Hadoop, or queue-based architecture) or use checkpoints so that you can save your work frequently.
- Use Spot Instance interruption notices to monitor the status of your Spot Instances.
- While we make every effort to provide this warning as soon as possible, it is possible that your Spot Instance is terminated before the warning can be made available. Test your application to ensure that it handles an unexpected instance termination gracefully, even if you are testing for interruption notices. You can do so by running the application using an On-Demand Instance and then terminating the On-Demand Instance yourself.

Preparing for Instance Hibernation

You must install a hibernation agent on your instance, unless you used an AMI that already includes the agent. You must run the agent on instance startup, whether the agent was included in your AMI or you installed it yourself.

The following procedure helps you prepare a Windows instance. For directions to prepare a Linux instance, see [Preparing for Instance Hibernation](#) in the *Amazon EC2 User Guide for Linux Instances*.

To prepare a Windows instance

1. If your AMI doesn't include the agent, download the following files to the C:\Program Files\Amazon\Hibernate folder on your Windows instance:
 - [EC2HibernateAgent.exe](#)
 - [EC2HibernateAgent.ps1](#)
 - [LICENSE.txt](#)
2. Add the following command to the user data:

```
<powershell>."C:\Program Files\Amazon\Hibernate\EC2HibernateAgent.exe"</powershell>
```

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 an item in the [instance metadata \(p. 452\)](#) 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.

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/spot/instance-action
```

The `instance-action` item specifies the action and the approximate time, in UTC, when the action will occur. The following example indicates the time at which this instance will be stopped:

```
{"action": "stop", "time": "2017-09-18T08:22:00Z"}
```

The following example indicates the time at which this instance will be terminated:

```
{"action": "terminate", "time": "2017-09-18T08:22:00Z"}
```

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.

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/spot/termination-time
```

The `termination-time` item specifies the approximate time in UTC when the instance receives the shutdown signal. For example:

```
2015-01-05T18:02:00Z
```

If Amazon EC2 is not preparing to terminate the instance, or if you terminated the Spot Instance yourself, the `termination-time` item is either not present (so you receive an HTTP 404 error) or contains a value that is not a time value.

If Amazon EC2 fails to terminate the instance, the request status is set to `fulfilled`. The `termination-time` value remains in the instance metadata with the original approximate time, which is now in the past.

Spot Instance Data Feed

To help you understand the charges for your Spot Instances, Amazon EC2 provides a data feed that describes your Spot Instance usage and pricing. This data feed is sent to an Amazon S3 bucket that you specify when you subscribe to the data feed.

Data feed files arrive in your bucket typically once an hour, and each hour of usage is typically covered in a single data file. These files are compressed (gzip) before they are delivered to your bucket. Amazon EC2 can write multiple files for a given hour of usage where files are large (for example, when file contents for the hour exceed 50 MB before compression).

Note

If you don't have a Spot Instance running during a certain hour, you don't receive a data feed file for that hour.

Contents

- [Data Feed File Name and Format \(p. 281\)](#)
- [Amazon S3 Bucket Requirements \(p. 281\)](#)
- [Subscribing to Your Spot Instance Data Feed \(p. 282\)](#)
- [Deleting Your Spot Instance Data Feed \(p. 282\)](#)

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.
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 \(p. 282\)](#)
- [Spot Fleet Limits \(p. 283\)](#)
- [T3 Instances \(p. 283\)](#)
- [T2 Instances \(p. 283\)](#)

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 \(p. 899\)](#).

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.

T3 Instances

If you plan to use your T3 Spot Instances immediately and for a short duration, with no idle time for accruing CPU credits, we recommend that you launch your T3 Spot Instances in [standard \(p. 125\)](#) mode to avoid paying higher costs.

If you launch your T3 Spot Instances in [unlimited \(p. 120\)](#) mode and burst CPU immediately, you'll spend surplus credits for bursting. If you use the instance for a short duration, your instance doesn't have time to accrue CPU credits to pay down the surplus credits, and you are charged for the surplus credits when you terminate your instance.

Unlimited mode for T3 Spot Instances is suitable only if the instance runs for long enough to accrue CPU credits for bursting. Otherwise, paying for surplus credits makes T3 Spot Instances more expensive than M5 or C5 instances.

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 \(p. 120\)](#), 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 \(p. 284\)](#)
- [Bring Your Own License \(p. 284\)](#)
- [Dedicated Host Instance Capacity \(p. 285\)](#)
- [Dedicated Hosts Limitations and Restrictions \(p. 285\)](#)
- [Pricing and Billing \(p. 285\)](#)
- [Working with Dedicated Hosts \(p. 286\)](#)

- [Tracking Configuration Changes \(p. 295\)](#)

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. For more information about Microsoft Licensing, see [Amazon Web Services and Microsoft Licensing](#).
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 (for example, Windows Server or Windows SQL Server). You cannot activate your imported Windows AMI against the Amazon Windows 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 \(p. 295\)](#).

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 (whether 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 \(p. 293\)](#).

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 instances 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 \(p. 286\)](#)
- [Allocating Dedicated Hosts \(p. 287\)](#)
- [Launching Instances onto Dedicated Hosts \(p. 288\)](#)
- [Modifying Dedicated Host Auto-Placement \(p. 289\)](#)
- [Modifying Instance Tenancy and Affinity \(p. 290\)](#)
- [Viewing Dedicated Hosts \(p. 291\)](#)
- [Tagging Dedicated Hosts \(p. 291\)](#)
- [Monitoring Dedicated Hosts \(p. 292\)](#)
- [Releasing Dedicated Hosts \(p. 293\)](#)
- [Purchasing Dedicated Host Reservations \(p. 294\)](#)
- [Viewing Dedicated Host Reservations \(p. 295\)](#)

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**, **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 \(p. 286\)](#).

- d. **Quantity**—The number of Dedicated Hosts to allocate with these options.
4. (Optional) Choose **Add Tag** and enter a tag key and a tag value.
5. Choose **Allocate host**.

To allocate Dedicated Hosts using the command line tools

Use one of the following commands. The following commands allocate a Dedicated Host that supports *untargeted m4.1large* instance launches in the `eu-west-1a` Availability Zone, and apply a tag with a key of `purpose` and a value of `production`.

- [allocate-hosts \(AWS CLI\)](#)

```
aws ec2 allocate-hosts --instance-type "m4.large" --availability-zone "eu-west-1a"
--auto-placement "off" --quantity 1 --tag-specifications 'ResourceType=dedicated-
host,Tags=[{Key=purpose,Value=production}]'
```

- [New-EC2Host \(AWS Tools for Windows PowerShell\)](#)

The `TagSpecification` parameter used to tag a Dedicated Host on creation requires an object that specifies the type of resource to be tagged, the tag key, and the tag value. The following commands create the required object.

```
PS C:\> $tag = @{ Key="purpose"; Value="production" }
PS C:\> $tagspec = new-object Amazon.EC2.Model.TagSpecification
PS C:\> $tagspec.ResourceType = "dedicated-host"
PS C:\> $tagspec.Tags.Add($tag)
```

The following command allocates the Dedicated Host and applies the tag specified in the `$tagspec` object.

```
PS C:\> New-EC2Host -InstanceType m4.large -AvailabilityZone eu-west-1a -
AutoPlacement Off -Quantity 1 -TagSpecification $tagspec
```

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 that you are launching.

Note

The instances launched onto Dedicated Hosts can only be launched in a VPC. For more information, see [Introduction to VPC](#).

Before you launch your instances, take note of the limitations. For more information, see [Dedicated Hosts Limitations and Restrictions \(p. 285\)](#).

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, 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 \(p. 286\)](#).

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**, **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. If Dedicated Hosts do not support the selected instance type, they are disabled in the list.
 - Affinity—Choose one of the following options:
 - **Off**—The instance launches onto the specified host, but it is not guaranteed to restart on it if stopped.
 - **Host**—If stopped, the instance always restarts on the specified host.

For more information, see [Understanding Auto-Placement and Affinity \(p. 286\)](#).

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 and choose **Actions, Modify Auto-Placement**.
4. On the Modify Auto-placement window, for **Allow instance auto-placement**, choose **Yes** to enable auto-placement, or choose **No** to disable auto-placement. For more information, see [Understanding Auto-Placement and Affinity \(p. 286\)](#).
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**, and select the instance to modify.
3. Choose **Actions, Instance State**, and **Stop**.
4. Open the context (right-click) menu on the instance and choose **Instance Settings, Modify Instance Placement**.
5. On the **Modify Instance Placement** page, configure the following:
 - **Tenancy**—Choose one of the following:
 - Run a dedicated hardware instance—Launches the instance as a Dedicated Instance. For more information, see [Dedicated Instances \(p. 297\)](#).
 - 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, see [Understanding Auto-Placement and Affinity \(p. 286\)](#).

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. In the navigation pane, choose **Dedicated Hosts**.
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 existing Dedicated Hosts to categorize them in different ways, for example, by purpose, owner, or environment. This helps you to quickly find a specific Dedicated Host based on the custom tags that you've assigned. Dedicated Host tags can also be used for cost allocation tracking.

You can also apply tags to Dedicated Hosts at the time of creation. For more information, see [Allocating Dedicated Hosts \(p. 287\)](#).

You can tag a Dedicated Host using the Amazon EC2 console and command line tools.

To tag a Dedicated Host using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Select the Dedicated Host 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 Dedicated Host.
7. Choose **Save**.

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. AWS does not restart instances in this state. 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 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** and select the reservation from the list provided.
4. Choose **Details** for information about the reservation.
5. Choose **Hosts** for information about the Dedicated Hosts with which the reservation is associated.

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 \(p. 283\)](#).

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 \(p. 301\)](#).

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 \(p. 186\)](#).

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 \(p. 373\)](#).

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 \(p. 239\)](#).

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 \(p. 299\)](#)
- [Launching Dedicated Instances into a VPC \(p. 300\)](#)
- [Displaying Tenancy Information \(p. 300\)](#)
- [Changing the Tenancy of an Instance \(p. 301\)](#)
- [Changing the Tenancy of a VPC \(p. 301\)](#)

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 \(p. 288\)](#).

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 \(p. 286\)](#). 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
```

On-Demand Capacity Reservations

On-Demand Capacity Reservations enable you to reserve capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. This gives you the ability to create and manage capacity reservations independently from the billing discounts offered by Reserved Instances (RI). By creating Capacity Reservations, you ensure that you always have access to EC2 capacity when you need it, for as long as you need it. Capacity Reservations can be created at any time, without entering into a one-year or three-year term commitment, and the capacity is available immediately. When you no longer need the reservation, cancel the Capacity Reservation to stop incurring charges for it.

When you create a Capacity Reservation, you specify the Availability Zone in which you want to reserve the capacity, the number of instances for which you want to reserve capacity, and the instance attributes, including the instance type, tenancy, and platform/OS. Capacity Reservations can only be used by instances that match their attributes. By default, they are automatically used by running instances that match the attributes. If you don't have any running instances that match the attributes of the Capacity Reservation, it remains unused until you launch an instance with matching attributes.

In addition, you can use your Regional RIs with your Capacity Reservations to benefit from billing discounts. This gives you the flexibility to selectively add capacity reservations and still get the Regional RI discounts for that usage. AWS automatically applies your RI discount when the attributes of a Capacity Reservation match the attributes of an active Regional RI.

Contents

- [Differences between Capacity Reservations and RIs \(p. 302\)](#)
- [Capacity Reservation Limits \(p. 303\)](#)
- [Capacity Reservation Limitations and Restrictions \(p. 303\)](#)
- [Capacity Reservation Pricing and Billing \(p. 303\)](#)
- [Working with Capacity Reservations \(p. 304\)](#)

Differences between Capacity Reservations and RIs

The following table highlights some key differences between Capacity Reservations and RIs:

	Capacity Reservations	Zonal RIs	Regional RIs
Term	No commitment required. Can be created and cancelled as needed.	Require fixed one-year or three-year commitment.	
Capacity benefit	Reserve capacity in a specific Availability Zone.	Reserve capacity in a specific Availability Zone.	Do not reserve capacity in an Availability Zone.
Billing discount	No billing discount. Instances launched into a Capacity Reservation are charged at their standard On-Demand rates.	Provide billing discounts.	

	Capacity Reservations	Zonal RIs	Regional RIs
	However, Regional RIs can be used with Capacity Reservations to get a billing discount.		
Instance Limits	Limited to your On-Demand Instance limits per Region.	Limited to 20 per Availability Zone. A limit increase can be requested.	Limited to 20 per Region. A limit increase can be requested.

Capacity Reservation Limits

The number of instances for which you are allowed to reserve capacity is based on your account's On-Demand Instance limit. You can reserve capacity for as many instances as that limit allows, minus the number of instances that are already running.

Capacity Reservation Limitations and Restrictions

Before you create Capacity Reservations, take note of the following limitations and restrictions.

- Active and unused Capacity Reservations count towards your On-Demand Instance limits
- Capacity Reservations can't be shared across AWS accounts
- Capacity Reservations are not transferable from one AWS account to another
- Zonal RI billing discounts do not apply to Capacity Reservations
- Capacity Reservations can't be created in Placement Groups
- Capacity Reservations can't be used with Dedicated Hosts

Capacity Reservation Pricing and Billing

Pricing

When the Capacity Reservation is active, you will be charged equivalent On-Demand rate whether you run the instances or not. If you do not use the reservation, this will show up as unused reservation on your EC2 bill. When you run an instance that matches the attributes of a reservation, you just pay for the instance and nothing for the reservation. There are no upfront or additional charges.

For example, if you create a Capacity Reservation for 20 m4.large Linux instances and run 15 m4.large Linux instances in the same Availability Zone, you will be charged for 15 instances and for 5 unused spots in the reservation.

Note

Regional RIs billing discounts apply to Capacity Reservations. AWS automatically applies your active Regional RIs to active and unused Capacity Reservations that have matching attributes.

For more information about Regional RIs, see [Reserved Instances \(p. 186\)](#).

For more information about Amazon EC2 pricing, see [Amazon EC2 Pricing](#).

Billing

Capacity Reservations are billed at per-second granularity. This means that you are charged for partial hours. For example, if a reservation remains active in your account for 24 hours and 15 minutes, you will be billed for 24.25 reservation hours.

The following example shows how a Capacity Reservation is billed. The Capacity Reservation is created for one m4.large Linux instance, which has an On-Demand rate of \$0.10 per usage hour. In this example,

the Capacity Reservation is active in the account for five hours. The Capacity Reservation is unused for the first hour, so it is billed for one unused hour at the m4.large instance type's standard On-Demand rate. In hours two through five, the Capacity Reservation is occupied by an m4.large instance. During this time, the Capacity Reservation accrues no charges, and the account is instead billed for the m4.large instance occupying it. In the sixth hour, the Capacity Reservation is cancelled and the m4.large instance runs normally outside of the reserved capacity. For that hour, it is charged at the On-Demand rate of the m4.large instance type.

Hour	1	2	3	4	5	6
Unused Capacity Reservation	\$0.10	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
On-demand Instance Usage	\$0.00	\$0.10	\$0.10	\$0.10	\$0.10	\$0.10
Hourly cost	\$0.10	\$0.10	\$0.10	\$0.10	\$0.10	\$0.10

Billing Discounts

Regional RIs billing discounts apply to Capacity Reservations. AWS automatically applies your active Regional RIs to active Capacity Reservations that have matching attributes. For more information about Regional RIs, see [Reserved Instances \(p. 186\)](#).

Note

Zonal RI billing discounts do not apply to Capacity Reservations.

When your instance-hours and reservation-hours combined exceed your total eligible discounted Regional RI hours, discounts are preferentially applied to instance-hours first and then to unused reservation-hours.

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 Capacity Reservations.

You can view the charges online, or you can download a CSV file. For more information, see [Capacity Reservation Line Items](#) in the *AWS Billing and Cost Management User Guide*.

Working with Capacity Reservations

To start using Capacity Reservations, you need to create the capacity reservation in the required Availability Zone. After you created a Capacity Reservation, you can launch instances into the reserved capacity, view its capacity utilization in real time, and increase or decrease its capacity as needed.

By default, Capacity Reservations automatically match new instances and running instances that have matching attributes (instance type, platform, and Availability Zone). In other words, instances that have matching attributes automatically run in the Capacity Reservation's capacity. However, you can also target a Capacity Reservation for specific workloads. This enables you to explicitly control which instances are allowed to run in that reserved capacity.

Contents

- [Creating a Capacity Reservation \(p. 305\)](#)
- [Launching an Instance into an Existing Capacity Reservation \(p. 306\)](#)
- [Modifying a Capacity Reservation \(p. 307\)](#)

- [Modifying an Instance's Capacity Reservation Settings \(p. 307\)](#)
- [Viewing a Capacity Reservation \(p. 308\)](#)
- [Cancelling a Capacity Reservation \(p. 309\)](#)

Creating a Capacity Reservation

Creating a Capacity Reservation in your account creates a capacity reservation in a specific Availability Zone. After it is created, you can launch instances into the reserved capacity as needed.

Note

Your request to create a Capacity Reservation could fail if Amazon EC2 does not have sufficient capacity to fulfil the request. If your request fails due to Amazon EC2 capacity constraints, either try again at a later time, try in a different Availability Zone, or request a smaller capacity reservation. If your application is flexible across instance types and sizes, try to create a Capacity Reservation with different instance attributes.

Your request could also fail if the requested quantity exceeds your On-Demand Instance limit for the selected instance type. If your request fails due to limit constraints, increase your On-Demand Instance limit for the required instance type and try again. For more information about increasing your instance limits, see [Amazon EC2 Service Limits \(p. 899\)](#).

After you create the Capacity Reservation, the capacity is available immediately. The capacity remains reserved for your use as long as the Capacity Reservation is active, and you can launch instances into it at any time. If the Capacity Reservation is open, new instances and existing instances that have matching attributes automatically run in the Capacity Reservation's capacity. If the Capacity Reservation is targeted, instances must specifically target it to run in the reserved capacity.

You can create a Capacity Reservation using the Amazon EC2 console or the AWS CLI.

To create a Capacity Reservation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Capacity Reservations**, choose **Create Capacity Reservation**.
3. On the Create a Capacity Reservation page, configure the following settings in the **Instance details** section:
 - a. **Instance Type**—Specify the type of instance to launch into the reserved capacity.
 - b. **Launch EBS-optimized instances**—Specify whether to reserve the capacity for EBS-optimized instances. This option is selected by default for some instance types. For more information about EBS-optimized instances, see [Amazon Elastic Block Store \(p. 732\)](#).
 - c. **Attach instance store at launch**—Indicate whether instances launched into the Capacity Reservation use temporary block-level storage. The data on an instance store volume persists only during the life of the associated instance.
 - d. **Platform**—Specify the operating system for your intended instances.
 - e. **Availability Zone**—Specify the Availability Zone in which to reserve the capacity.
 - f. **Quantity**—Specify the number instances for which to reserve capacity. If you specify a quantity that exceeds your remaining On-Demand Instance limit for the selected instance type, the request will be denied.
4. Configure the following settings in the **Reservation details** section:
 - a. **Reservation Ends**—Choose one of the following options:
 - **Manually**—Reserve the capacity until you explicitly cancel it.
 - **Specific time**—Cancels the capacity reservation automatically. The capacity reservation is released automatically at the specified date and time. The Capacity Reservation is cancelled within an hour from the specified time. For example, if you specify 5/31/2019, 13:30:55, the Capacity Reservation is guaranteed to end between 13:30:55 and 14:30:55 on 5/31/2019.

Note

After the reservation ends, you can no longer target instances to the Capacity Reservation. Instances running in the reserved capacity continue to run uninterrupted. If instances targeting a Capacity Reservation are stopped, you cannot restart them until you remove their Capacity Reservation targeting preference or configure them to target a different Capacity Reservation.

- b. **Instance eligibility**—Choose one of the following options:

- **open**—(Default) The Capacity Reservation matches any instance that has matching attributes (instance type, platform, and Availability Zone). If you launch an instance with matching attributes, it is placed into the reserved capacity automatically.
- **targeted**—The Capacity Reservation only accepts instances that have matching attributes (instance type, platform, and Availability Zone), and explicitly target the reservation.

5. Choose **Request reservation**.

To create a Capacity Reservation using the AWS CLI

Use the [create-capacity-reservation](#) command:

```
C:\> aws ec2 create-capacity-reservation --instance-type instance_type --instance-platform platform_type --availability-zone az --instance-count quantity
```

Launching an Instance into an Existing Capacity Reservation

You can launch an instance into a Capacity Reservation if it has matching attributes (instance type, platform, and Availability Zone) and sufficient capacity. Launching an instance into a Capacity Reservation reduces its available capacity by the number of instances launched. For example, if you launch three instances, the Capacity Reservation's available capacity is reduced by three.

You can launch an instance into a Capacity Reservation that you previously created using the Amazon EC2 console or the command line.

To launch an instance into an existing Capacity Reservation using the console

1. Open the Launch Instance wizard by doing one of the following:
 - Choose **Instances, Launch Instance**.
 - Choose **Capacity Reservations, Launch Instance**.
2. Complete the instance details to suit your requirements.
3. On the **Configure Instance Details** page, for **Capacity Reservation**, do one of the following:
 - Choose **Open** to launch the instance into any open Capacity Reservation that has matching attributes (instance type, platform, and Availability Zone) and sufficient capacity.

Note

If you do not have a matching open Capacity Reservation with sufficient capacity, the instance launches into On-Demand capacity.

- Choose **None** to prevent the instance from launching into a Capacity Reservation.
- Choose the specific Capacity Reservation into which to launch the instance.

Note

If the selected Capacity Reservation does not have sufficient capacity, the instance launch fails.

4. Choose **Review and Launch, Launch**.

- When prompted, select an existing key pair or create a new one, and choose **Launch Instances**.

To launch an instance into an existing Capacity Reservation using the AWS CLI

Use the `run-instances` command and specify the `--capacity-reservation-specification` parameter.

The following example launches a `t2.micro` instance into any open Capacity Reservation that has matching attributes and available capacity:

```
C:\> aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro  
--key-name MyKeyPair --availability-zone us-east-1b --capacity-reservation-specification  
CapacityReservationPreference=open
```

The following example launches a `t2.micro` instance into a targeted Capacity Reservation:

```
C:\> aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro  
--key-name MyKeyPair --availability-zone us-east-1b --capacity-reservation-specification  
CapacityReservationTarget=[{CapacityReservationId=cr-a1234567}]
```

Modifying a Capacity Reservation

You can change an active Capacity Reservation's attributes after you have created it. You cannot modify a Capacity Reservation after it has expired or after you have explicitly cancelled it.

When modifying a Capacity Reservation, you can only increase or decrease the quantity and change the way in which it is released. You cannot change a Capacity Reservation's instance type, EBS optimization, instance store settings, platform, Availability Zone, or instance eligibility. If you need to modify any of these attributes, we recommend that you cancel the reservation, and then create a new one with the required attributes.

You can modify a Capacity Reservation using the Amazon EC2 console and the AWS CLI.

To modify a Capacity Reservation using the console

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- Choose **Capacity Reservations**, select the Capacity Reservation to modify, and then choose **Edit**.
- Modify the **Quantity** or **Reservation ends** options as needed, and choose **Save changes**.

Note

If you specify a new quantity that exceeds your remaining On-Demand Instance limit for the selected instance type, the update fails.

To modify a Capacity Reservation using the AWS CLI

Use the `modify-capacity-reservations` command:

```
C:\> aws ec2 modify-capacity-reservation --capacity-reservation-id reservation_id --  
instance-count quantity --end-date-type limited/unlimited --end-date expiration_date
```

Modifying an Instance's Capacity Reservation Settings

You can modify an existing instance's Capacity Reservation settings at any time. You can modify a stopped instance to do the following:

- Target a specific Capacity Reservation. The instance cannot launch outside of the targeted Capacity Reservation.

- Launch on any Capacity Reservation that has matching attributes (instance type, platform, and Availability Zone) and available capacity.
- Avoid launching in a Capacity Reservation. The instance is prevented from launching in any Capacity Reservation, even if the reservation is open and has matching attributes (instance type, platform, and Availability Zone).

Note

You can only modify an instance's Capacity Reservation settings while it is stopped.

You can modify an instance's Capacity Reservation settings using the Amazon EC2 console and the AWS CLI.

To modify an instance's Capacity Reservation settings using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Instances**, select the instance to modify, and then choose **Actions, Modify Capacity Reservation Settings**.
3. For **Capacity Reservation**, do one of the following:
 - Choose **Open** to configure the instance to run in any open Capacity Reservation that has matching attributes (instance type, platform, and Availability Zone) and sufficient capacity.

Note

If you do not have a matching open Capacity Reservation with sufficient capacity, the instance launches into On-Demand capacity.

- Choose **None** to prevent the instance from launching into a Capacity Reservation.
- Choose the specific Capacity Reservation in which the instance should run.

Note

If the instance attributes (instance type, platform, and Availability Zone) do not match those of the selected Capacity Reservation, or if the selected Capacity Reservation does not have sufficient capacity, the instance launch fails.

To modify an instance's Capacity Reservation settings using the AWS CLI

Use the [modify-instance-capacity-reservation-attributes](#) command:

```
C:\> aws ec2 modify-instance-capacity-reservation-attributes --instance-id instance_id --capacity-reservation-specification 'CapacityReservationPreference=none|open'
```

Viewing a Capacity Reservation

Capacity Reservations have three possible states:

- **active**—The Capacity Reservation is active and the capacity is available for your use.
- **expired**—The Capacity Reservation expired automatically at the date and time specified in your reservation request. The reserved capacity is no longer available for your use.
- **cancelled**—The Capacity Reservation was manually cancelled. The reserved capacity is no longer available for your use.
- **pending**—The Capacity Reservation request was successful but the capacity provisioning is still pending.
- **failed**—The Capacity Reservation request has failed. A request might fail due to invalid request parameters, capacity constraints, or instance limit constraints. Failed requests are retained for 60 minutes.

You can view your active Capacity Reservations using the Amazon EC2 console and the AWS CLI.

To view your Capacity Reservations using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Capacity Reservations** and select a Capacity Reservation to view.
3. Choose **View launched instances for this reservation**.

To view your Capacity Reservations using the AWS CLI

Use the [describe-capacity-reservations](#) command:

```
C:\> aws ec2 describe-capacity-reservations
```

Cancelling a Capacity Reservation

You can cancel a Capacity Reservation at any time if you no longer need the reserved capacity. When you cancel a Capacity Reservation, the capacity is released immediately, and it is no longer reserved for your use.

You can cancel empty Capacity Reservations and Capacity Reservations that have running instances. If you cancel a Capacity Reservation that has running instances, the instances continue to run normally outside of the capacity reservation at standard On-Demand Instance rates or at a discounted rate if you have an active matching Regional RI.

After you cancel a Capacity Reservation, instances that target it can no longer launch. Modify these instances so that they either target a different Capacity Reservation, launch into any 'open' Capacity Reservation with matching attributes and sufficient capacity, or avoid launching into a Capacity Reservation. For more information, see [Modifying an Instance's Capacity Reservation Settings \(p. 307\)](#).

You can cancel a Capacity Reservation using the Amazon EC2 console and the AWS CLI.

To cancel a Capacity Reservation using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Capacity Reservations** and select the Capacity Reservation to cancel.
3. Choose **Cancel reservation**, **Cancel reservation**.

To cancel a Capacity Reservation using the AWS CLI

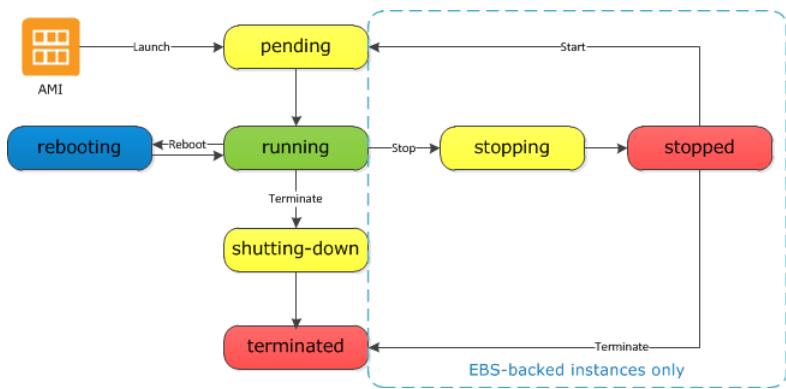
Use the [cancel-capacity-reservation](#) command:

```
C:\> aws ec2 cancel-capacity-reservation --capacity-reservation-id reservation_id
```

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.



The following table provides a brief description of each instance state and indicates whether it is billed or not.

Note

The table indicates billing for instance usage only. Some AWS resources, such as Amazon EBS volumes and Elastic IP addresses, incur charges regardless of the instance's state. For more information, see [Avoiding Unexpected Charges](#) in the *AWS Billing and Cost Management User Guide*.

Instance state	Description	Instance usage billing
pending	The instance is preparing to enter the running state. An instance enters the pending state when it launches for the first time, or when it is restarted after being in the stopped state.	Not billed
running	The instance is running and ready for use.	Billed
stopping	The instance is preparing to be stopped or stop-hibernated.	Not billed if preparing to stop Billed if preparing to hibernate
stopped	The instance is shut down and cannot be used. The instance can be 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 Reserved Instances that applied to terminated instances are billed until the end of their term according to their payment option. For more information, see Reserved Instances (p. 186)

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 hour or partial hour 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 \(p. 314\)](#) and [Connecting to Your Windows Instance \(p. 359\)](#).

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 hourly 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.

Your instance retains its private IPv4 address, which means that an Elastic IP address associated with the private IPv4 address or network interface is still associated with your instance. If your instance has an IPv6 address, it retains its IPv6 address.

Each time you transition an instance from `stopped` to `running`, we charge a full instance hour, even if these transitions happen multiple times within a single hour.

For more information, see [Stop and Start Your Instance \(p. 362\)](#).

Instance Hibernate (Amazon EBS-Backed Instances Only)

When you hibernate an instance, we signal the operating system to perform hibernation (suspend-to-disk), which saves the contents from the instance memory (RAM) to your Amazon EBS root volume. We persist the instance's Amazon EBS root volume and any attached Amazon EBS data volumes. When you restart your instance, the Amazon EBS root volume is restored to its previous state and the RAM contents are reloaded. Previously attached data volumes are reattached and the instance retains its instance ID.

When you hibernate your instance, it enters the `stopping` state, and then the `stopped` state. We don't charge hourly usage for a hibernated instance when it is in the `stopped` state, but we do charge while

it is in the stopping state, unlike when you [stop an instance \(p. 311\)](#) without hibernating it. We don't charge usage for data transfer fees, but we do charge for the storage for any Amazon EBS volumes, including storage for the RAM data.

When you restart your hibernated 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.

Your instance retains its private IPv4 address, which means that an Elastic IP address associated with the private IPv4 address or network interface is still associated with your instance. If your instance has an IPv6 address, it retains its IPv6 address.

For more information, see [Hibernate Your Instance \(p. 365\)](#).

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 hour.

For more information, see [Reboot Your Instance \(p. 365\)](#).

Instance Retirement

An instance is scheduled to be retired when AWS detects the irreparable failure of the underlying hardware hosting the instance. When an instance reaches its scheduled retirement date, it is stopped or terminated by AWS. If your instance root device is an Amazon EBS volume, the instance is stopped, and you can start it again at any time. If your instance root device is an instance store volume, the instance is terminated, and cannot be used again.

For more information, see [Instance Retirement \(p. 366\)](#).

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. 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 \(p. 368\)](#).

Differences Between Reboot, Stop, Hibernate, and Terminate

The following table summarizes the key differences between rebooting, stopping, hibernating, and terminating your instance.

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Hibernate (Amazon EBS-backed instances only)	Terminate
Host computer	The instance stays on the same host computer	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.	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	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.	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	The Elastic IP address remains associated with the instance	The Elastic IP address remains associated with the instance	The Elastic IP address is disassociated from the instance
IPv6 address	The address stays the same	The instance keeps its IPv6 address	The instance keeps its IPv6 address	None
Instance store volumes	The data is preserved	The data is erased	The data is erased	The data is erased
Root device volume	The volume is preserved	The volume is preserved	The volume is preserved	The volume is deleted by default
RAM (contents of memory)	The RAM is erased	The RAM is erased	The RAM is saved to a file on the root volume	The RAM is erased
Billing	The instance billing hour doesn't change.	You stop incurring charges for an instance as soon as its state changes	You incur charges while the instance is in the stopping state, but stop	You stop incurring charges for an instance as soon as its

Characteristic	Reboot	Stop/start (Amazon EBS-backed instances only)	Hibernate (Amazon EBS-backed instances only)	Terminate
		to stopping. Each time an instance transitions from stopped to running, we start a new instance billing hour.	incurring charges when the instance is in the stopped state. Each time an instance transitions from stopped to running, we start a new instance billing hour.	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 \(p. 370\)](#).

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. 315)
[Amazon EC2 console] Create a launch template and launch the instance from the launch template	Launching an Instance from a Launch Template (p. 321)
[Amazon EC2 console] Use an existing instance as the base	Launching an Instance Using Parameters from an Existing Instance (p. 331)
[Amazon EC2 console] Use an AMI that you purchased from the AWS Marketplace	Launching an AWS Marketplace Instance (p. 332)
[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. 333)

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 VPC can use to contact the instance. For more information about connecting to your instance, see [Connecting to Your Windows Instance \(p. 359\)](#).

When you are finished with an instance, be sure to terminate it. For more information, see [Terminate Your Instance \(p. 368\)](#).

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 \(p. 12\)](#).

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

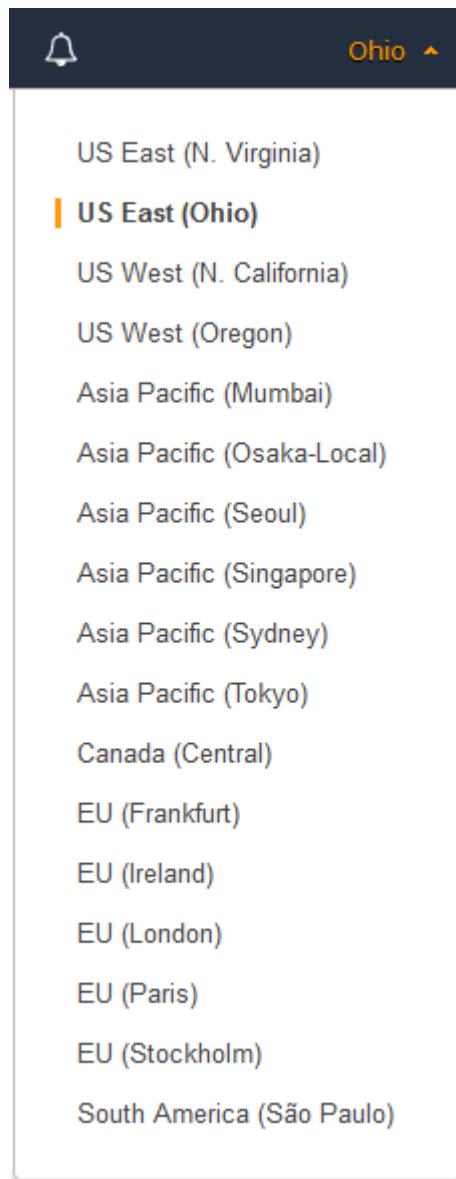
When you launch an instance, you must select a configuration, known as an Amazon Machine Image (AMI). An AMI contains the information required to create a new instance. For example, an AMI might contain the software required to act as a web server: for example, Windows, Apache, and your website.

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 \(p. 881\)](#).



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 \(p. 332\)](#).

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 **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.
- c. 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 \(p. 110\)](#).

To remain eligible for the free tier, choose the **t2.micro** instance type. For more information, see [Burstable Performance Instances \(p. 116\)](#).

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.
 - (Optional) To help ensure that you maintain the correct number of instances to handle demand on your application, you can choose **Launch into Auto Scaling Group** to create a launch configuration and an Auto Scaling group. Auto Scaling scales the number of instances in the group according to your specifications. For more information, see the [Amazon EC2 Auto Scaling User Guide](#).
 - **Purchasing option:** Choose **Request Spot instances** to launch a Spot Instance. 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 \(p. 241\)](#).
 - **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. You can select **No preference** to let AWS choose a default subnet in any Availability Zone. To create a new subnet, choose **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and choose **Refresh** to load your subnet in the list.
 - **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 \(p. 644\)](#).
 - **Auto-assign IPv6 IP:** Specify whether your instance receives an IPv6 address from the range of the subnet. Select **Enable** or **Disable** to override the subnet's default setting. This option is only

available if you've associated an IPv6 CIDR block with your VPC and subnet. For more information, see [Your VPC and Subnets](#) in the *Amazon VPC User Guide*.

- **Domain join directory:** Select the AWS Directory Service directory (domain) to which your Windows instance is joined. The directory must be in the same VPC that you selected for your instance. If you select a domain, you must select an IAM role. For more information, see [Launching an Instance \(Simple AD and Microsoft AD\)](#).
- **Capacity Reservation:** Specify whether to launch the instance into shared capacity or an existing Capacity Reservation. For more information, see [Launching an Instance into an Existing Capacity Reservation \(p. 306\)](#).
- **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 \(p. 633\)](#).
- **CPU options:** Choose **Specify CPU options** to specify a custom number of vCPUs during launch. Set the number of CPU cores and threads per core. For more information, see [Optimizing CPU Options \(p. 425\)](#).
- **Shutdown behavior:** Select whether the instance should stop or terminate when shut down. For more information, see [Changing the Instance Initiated Shutdown Behavior \(p. 370\)](#).
- **Enable termination protection:** To prevent accidental termination, select this check box. For more information, see [Enabling Termination Protection for an Instance \(p. 369\)](#).
- **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 \(p. 498\)](#).
- **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 \(p. 807\)](#).
- **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 \(p. 297\)](#) and [Dedicated Hosts \(p. 283\)](#).
- **T2/T3 Unlimited:** Select this check box to enable applications to burst beyond the baseline for as long as needed. Additional charges may apply. For more information, see [Burstable Performance Instances \(p. 116\)](#).
- **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 \(p. 665\)](#). 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 \(p. 647\)](#).

- **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 \(p. 692\)](#).
 - **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 \(p. 843\)](#) and [Amazon EBS Volumes \(p. 734\)](#).
 - **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

The following Amazon EBS volume considerations apply to Windows boot volumes:

- Windows boot volumes must use an MBR partition table, which limits the usable space to 2 TiB, regardless of volume size.
- Windows boot volumes of 2 TiB (2048 GiB) that have been converted to use a dynamic MBR partition table display an error when examined with Disk Manager.

The following Amazon EBS volume considerations apply to Windows data (non-boot) volumes:

- Windows volumes 2 TiB (2048 GiB) or greater must use a GPT partition table to access the entire volume.
- Amazon EBS volumes over 2048 GiB that are attached to Windows instances at launch are automatically formatted with a GPT partition table.
- Amazon EBS volumes attached to Windows instances after launch must be manually initialized with a GPT partition table. For more information, see [Making an Amazon EBS Volume Available for Use](#).

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 Windows \(p. 773\)](#).

- **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 \(p. 736\)](#).

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 \(p. 736\)](#).

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 \(p. 371\)](#).
- **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 \(p. 816\)](#).

When done configuring your volumes, choose **Next: Add Tags**.

8. On the **Add Tags** page, specify [tags \(p. 890\)](#) 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 Windows Instances \(p. 548\)](#).) 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.

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 RDP (port 3389).
- 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.

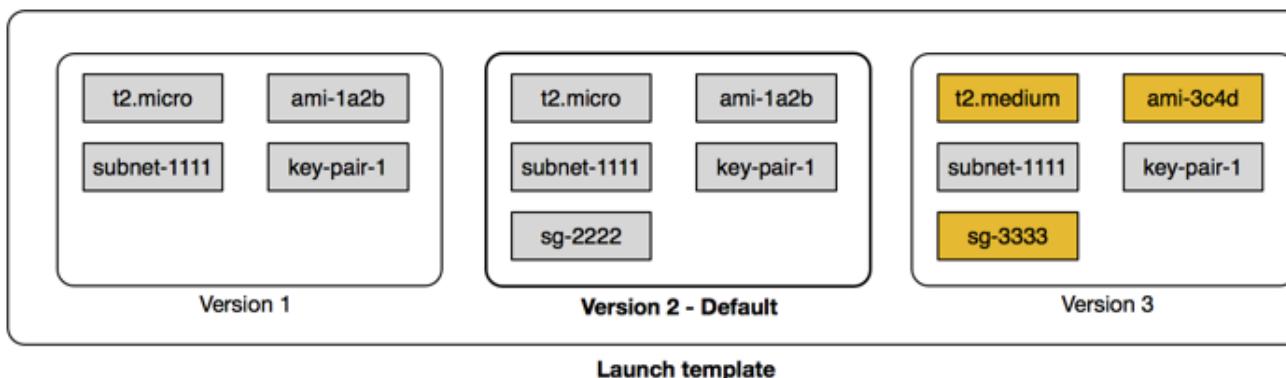
12. (Optional) You can create a status check alarm for the instance (additional fees may apply). (If you're not sure, you can always add one later.) On the confirmation screen, choose **Create status check alarms** and follow the directions. For more information, see [Creating and Editing Status Check Alarms \(p. 493\)](#).
13. If the instance fails to launch or the state immediately goes to `terminated` instead of `running`, see [Troubleshooting Instance Launch Issues \(p. 965\)](#).

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 \(p. 322\)](#)
- [Using Launch Templates to Control Launch Parameters \(p. 322\)](#)
- [Controlling the Use of Launch Templates \(p. 322\)](#)
- [Creating a Launch Template \(p. 323\)](#)
- [Managing Launch Template Versions \(p. 327\)](#)
- [Launching an Instance from a Launch Template \(p. 329\)](#)
- [Using Launch Templates with Amazon EC2 Auto Scaling \(p. 330\)](#)
- [Using Launch Templates with EC2 Fleet \(p. 330\)](#)
- [Using Launch Templates with Spot Fleet \(p. 330\)](#)
- [Deleting a Launch Template \(p. 331\)](#)

Launch Template Restrictions

The following rules apply to launch templates and launch template versions:

- You are limited to creating 5,000 launch templates per Region and 10,000 versions per launch template.
- Launch 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 permissions to use the `ec2:RunInstances` action. They must also have permissions to create or use the resources that are created or associated with the instance. You can use resource-level permissions for the `ec2:RunInstances` action to control the launch parameters that users can specify. Alternatively, you can grant users permissions to launch an instance using a launch template. This enables you to manage launch parameters in a launch template rather than in an IAM policy, and to use a launch template as an authorization vehicle for launching instances. For example, you can specify that users can only launch instances using a launch template, and that they can only use a specific launch template. You can also control the launch parameters that users can override in the launch template. For example policies, see [Launch Templates \(p. 618\)](#).

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 \(p. 573\)](#) and the following example policies: [Example: Working with Launch Templates \(p. 624\)](#).

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

Create a new launch template using parameters that you define, or use an existing launch template or an instance as the basis for a new launch template.

To create a new launch template using defined parameters (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** and provide a name and description.
4. For **Launch template contents**, provide the following information:
 - **AMI ID:** An AMI from which to launch the instance. To search through all available AMIs, choose **Search for AMI**. To select a commonly used AMI, choose **Quick Start**. Or, choose **AWS Marketplace or Community AMIs**. You can use an AMI that you own or [find a suitable AMI \(p. 53\)](#).
 - **Instance type:** Ensure that the instance type is compatible with the AMI that you've specified. For more information, see [Instance Types \(p. 110\)](#).
 - **Key pair name:** The key pair for the instance. For more information, see [Amazon EC2 Key Pairs and Windows Instances \(p. 543\)](#).
 - **Network type:** If applicable, whether to launch the instance into a VPC or EC2-Classic. If you choose **VPC**, specify the subnet in the **Network interfaces** section. If you choose **Classic**, ensure that the specified instance type is supported in EC2-Classic and specify the Availability Zone for the instance.
 - **Security Groups:** One or more security groups to associate with the instance. For more information, see [Amazon EC2 Security Groups for Windows Instances \(p. 548\)](#).
5. For **Network interfaces**, you can specify up to two [network interfaces \(p. 665\)](#) for the instance.
 - **Device:** The device number for the network interface, for example, `eth0` for the primary network interface. If you leave the field blank, AWS creates the primary network interface.
 - **Network interface:** The ID of the network interface, or leave blank to let AWS create a new network interface.
 - **Description:** (Optional) A description for the new network interface.
 - **Subnet:** The subnet in which to create a new network interface. For the primary network interface (`eth0`), this is the subnet in which the instance is launched. If you've entered an existing network interface for `eth0`, the instance is launched in the subnet in which the network interface is located.
 - **Auto-assign public IP:** Whether to automatically assign a public IP address to the network interface with the device index of `eth0`. This setting can only be enabled for a single, new network interface.
 - **Primary IP:** A private IPv4 address from the range of your subnet. Leave blank to let AWS choose a private IPv4 address for you.
 - **Secondary IP:** A secondary private IPv4 address from the range of your subnet. Leave blank to let AWS choose one for you.
 - (IPv6-only) **IPv6 IPs:** An IPv6 address from the range of the subnet.

- **Security group ID:** The ID of a security group in your VPC with which to associate the network interface.
 - **Delete on termination:** Whether the network interface is deleted when the instance is deleted.
6. For **Storage (Volumes)**, specify volumes to attach to the instance besides the volumes specified by the AMI.
- **Volume type:** The instance store or Amazon EBS volumes with which to associate your instance. The type of volume depends on the instance type that you've chosen. For more information, see [Amazon EC2 Instance Store \(p. 843\)](#) and [Amazon EBS Volumes \(p. 734\)](#).
 - **Device name:** A device name for the volume.
 - **Snapshot:** The ID of the snapshot from which to create the volume.
 - **Size:** For Amazon EBS volumes, the storage size.
 - **Volume type:** For Amazon EBS volumes, the volume type. For more information, see [Amazon EBS Volume Types \(p. 736\)](#).
 - **IOPS:** For the Provisioned IOPS SSD volume type, the number of I/O operations per second (IOPS) that the volume can support.
 - **Delete on termination:** For Amazon EBS volumes, whether to delete the volume when the instance is terminated. For more information, see [Preserving Amazon EBS Volumes on Instance Termination \(p. 371\)](#).
 - **Encrypted:** Whether 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 \(p. 816\)](#).
 - **Key:** For encrypting new Amazon EBS volumes, the master key to use when encrypting the volumes. Enter the default master key for your account, or any customer master key (CMK) that you have previously created using the AWS Key Management Service. You can paste the full ARN of any key to which you have access. For more information, see the [AWS Key Management Service Developer Guide](#).
7. For **Tags**, specify [tags \(p. 890\)](#) by providing key and value combinations. You can tag the instance, the volumes, or both.
8. For **Advanced Details**, expand the section to view the fields and specify any additional parameters for the instance.
- **Purchasing option:** The purchasing model. Choose **Request Spot instances** to request Spot Instances at the Spot price, capped at the On-Demand price, and choose **Customize Spot parameters** to change the default Spot Instance settings. If you do not request a Spot Instance, EC2 launches an On-Demand Instance by default. For more information, see [Spot Instances \(p. 225\)](#).
 - **IAM instance profile:** An AWS Identity and Access Management (IAM) instance profile to associate with the instance. For more information, see [IAM Roles for Amazon EC2 \(p. 633\)](#).
 - **Shutdown behavior:** Whether the instance should stop or terminate when shut down. For more information, see [Changing the Instance Initiated Shutdown Behavior \(p. 370\)](#).
 - **Stop - Hibernate behavior:** Whether the instance is enabled for hibernation. This field is only valid for instances that meet the hibernation prerequisites. For more information, see [Hibernate Your Instance \(p. 365\)](#).
 - **Termination protection:** Whether to prevent accidental termination. For more information, see [Enabling Termination Protection for an Instance \(p. 369\)](#).
 - **Monitoring:** Whether to enable detailed monitoring of the instance using Amazon CloudWatch. Additional charges apply. For more information, see [Monitoring Your Instances Using CloudWatch \(p. 498\)](#).
 - **Elastic Graphics:** An Elastic Graphics accelerator to attach to the instance. Not all instance types support Elastic Graphics. For more information, see [Amazon Elastic Graphics \(p. 475\)](#).

- **T2/T3 Unlimited:** Whether to enable applications to burst beyond the baseline for as long as needed. This field is only valid for T2 and T3 instances. Additional charges may apply. For more information, see [Burstable Performance Instances \(p. 116\)](#).
- **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 \(p. 692\)](#).
- **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 \(p. 807\)](#).
- **Tenancy:** Choose whether to run your instance on shared hardware (**Shared**), isolated, dedicated hardware (**Dedicated**), or on a Dedicated Host (**Dedicated host**). Additional charges may apply. For more information, see [Dedicated Instances \(p. 297\)](#) and [Dedicated Hosts \(p. 283\)](#). 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 specified a kernel, you may need to specify a specific RAM disk with the drivers to support it. Only valid for paravirtual (PV) AMIs.
- **Kernel ID:** A kernel for the instance. Only valid for paravirtual (PV) AMIs.
- **User data:** You can specify user data to configure an instance during launch, or to run a configuration script. For more information, see [Running Commands on Your Windows Instance at Launch \(p. 447\)](#).

9. Choose **Create launch template**.

To create a launch template from an existing launch template (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. 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 from an instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, and choose **Actions, Create Template From Instance**.
4. Provide a name and description, and adjust the launch parameters as required.

Note

When you create a launch template from an instance, the instance's network interface IDs and IP addresses are not included in the template.

5. Choose **Create Template From Instance**.

To create a launch template (AWS CLI)

- Use the `create-launch-template` (AWS CLI) command. The following example creates a launch template that specifies the following:
 - The instance type (`r4.4xlarge`) and AMI (`ami-8c1be5f6`) to launch
 - The number of cores (4) and threads per core (2) for a total of 8 vCPUs (4 cores x 2 threads)
 - The subnet in which to launch the instance (`subnet-7b16de0c`)

The template assigns a public IP address and an IPv6 address to the instance and creates a tag for the instance (Name=webserver).

```
aws ec2 create-launch-template --launch-template-name TemplateForWebServer --version-description WebVersion1 --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 (AWS CLI)

- 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-0123d646e8048bab --query  
"LaunchTemplateData"
```

The following is example output.

```
{
```

```
"Monitoring": {},
"ImageId": "ami-8c1be5f6",
"BlockDeviceMappings": [
    {
        "DeviceName": "/dev/xvda",
        "Ebs": {
            "DeleteOnTermination": true
        }
    }
],
"EbsOptimized": false,
"Placement": {
    "Tenancy": "default",
    "GroupName": "",
    "AvailabilityZone": "us-east-1a"
},
"InstanceType": "t2.micro",
"NetworkInterfaces": [
    {
        "Description": "",
        "NetworkInterfaceId": "eni-35306abc",
        "PrivateIpAddresses": [
            {
                "Primary": true,
                "PrivateIpAddress": "10.0.0.72"
            }
        ],
        "SubnetId": "subnet-7b16de0c",
        "Groups": [
            "sg-7c227019"
        ],
        "Ipv6Addresses": [
            {
                "Ipv6Address": "2001:db8:1234:1a00::123"
            }
        ],
        "PrivateIpAddress": "10.0.0.72"
    }
]
}
```

You can write the output directly to a file, for example:

```
aws ec2 get-launch-template-data --instance-id i-0123d646e8048babc --query
"LaunchTemplateData" >> instance-data.json
```

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 \(p. 328\)](#)
- [Setting the Default Launch Template Version \(p. 328\)](#)
- [Deleting a Launch Template Version \(p. 329\)](#)

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 \(p. 323\)](#).

To create a launch template version (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 (AWS CLI)

- 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 (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 (AWS CLI)

- 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 (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 (AWS CLI)

- 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 (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 (AWS CLI)

- 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 \(p. 965\)](#).

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 (AWS CLI)

- 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 EC2 Fleet

You can create an EC2 Fleet request and specify a launch template in the instance configuration. When Amazon EC2 fulfills the EC2 Fleet request, it uses the launch parameters defined in the associated launch template. You can override some of the parameters that are specified in the launch template.

For more information, see [Creating an EC2 Fleet \(p. 349\)](#).

To create an EC2 Fleet with a launch template (AWS CLI)

- Use the [create-fleet](#) AWS CLI command. Use the `--launch-template-configs` parameter to specify the launch template and any overrides for the launch template.

Using Launch Templates with Spot Fleet

You can create a Spot Fleet request and specify a launch template in the instance configuration. When Amazon EC2 fulfills the Spot Fleet request, it uses the launch parameters defined in the associated launch template. You can override some of the parameters that are specified in the launch template.

For more information, see [Spot Fleet Requests \(p. 246\)](#).

To create a Spot Fleet request with a launch template (AWS CLI)

- 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 (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Launch Templates**.
3. Select the launch template and choose **Actions, Delete template**.
4. Choose **Delete launch template**.

To delete a launch template (AWS CLI)

- Use the [delete-launch-template](#) (AWS CLI) command and specify the launch template.

```
aws ec2 delete-launch-template --launch-template-id lt-01238c059e3466abc
```

Launching an Instance Using Parameters from an Existing Instance

The Amazon EC2 console provides a **Launch More Like This** wizard option that enables you to use a current instance as a base for launching other instances. This option automatically populates the Amazon EC2 launch wizard with certain configuration details from the selected instance.

Note

The **Launch More Like This** wizard option does not clone your selected instance; it only replicates some configuration details. To create a copy of your instance, first create an AMI from it, then launch more instances from the AMI.

Alternatively, create a [launch template \(p. 321\)](#) 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 \(p. 644\)](#).
- 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
- Association information. If the selected instance is associated with a configuration file, the same file is automatically associated with the new instance. If the configuration file includes a joined domain configuration, the new instance is joined to the same domain. For more information about joining a domain, see [Launching an Instance \(Simple AD and Microsoft AD\)](#).

The following configuration details are not copied from your selected instance; instead, the wizard applies their default settings or behavior:

- Number of network interfaces: The default is one network interface, which is the primary network interface (eth0).
- Storage: The default storage configuration is determined by the AMI and the instance type.

To use your current instance as a template

1. 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 \(p. 965\)](#).

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 \(p. 61\)](#). 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 \(p. 65\)](#).

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 \(p. 315\)](#). 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 [Connecting to Your Windows Instance \(p. 359\)](#).

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 \(p. 65\)](#).

10. If the instance fails to launch or the state immediately goes to **terminated** instead of **running**, see [Troubleshooting Instance Launch Issues \(p. 965\)](#).

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:

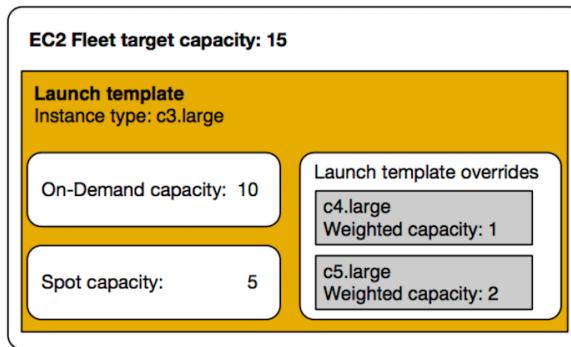
Method	Documentation
AWS CLI	Use the run-instances command, or see the following topic for more information: Launching an Instance .
AWS Tools for Windows PowerShell	Use the New-EC2Instance command, or see the following topic for more information: Launch an Amazon EC2 Instance Using Windows PowerShell
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 can also attempt 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 \(p. 228\)](#).



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 \(p. 334\)](#)
- [EC2 Fleet Configuration Strategies \(p. 334\)](#)
- [Managing an EC2 Fleet \(p. 342\)](#)

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. 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 \(p. 335\)](#)
- [EC2 Fleet Request Types \(p. 336\)](#)
- [Allocation Strategies for Spot Instances \(p. 336\)](#)
- [Configuring EC2 Fleet for On-Demand Backup \(p. 337\)](#)
- [Maximum Price Overrides \(p. 338\)](#)
- [EC2 Fleet Instance Weighting \(p. 338\)](#)
- [Walkthrough: Using EC2 Fleet with Instance Weighting \(p. 339\)](#)
- [Walkthrough: Using EC2 Fleet with On-Demand as the Primary Capacity \(p. 341\)](#)

Planning an EC2 Fleet

When planning your EC2 Fleet, we recommend that you do the following:

- Determine whether you want to create an EC2 Fleet that submits a synchronous or asynchronous one-time request for the desired target capacity, or one that maintains a target capacity over time. For more information, see [EC2 Fleet Request Types \(p. 336\)](#).
- 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 \(p. 338\)](#).
- 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 \(p. 347\)](#). For EC2 Fleet configuration examples, see [EC2 Fleet Example Configurations \(p. 356\)](#).

EC2 Fleet Request Types

There are three types of EC2 Fleet requests:

`instant`

If you configure the request type as `instant`, EC2 Fleet places a synchronous one-time request for your desired capacity. In the API response, it returns the instances that launched, along with errors for those instances that could not be launched.

`request`

If you configure the request type as `request`, EC2 Fleet places an asynchronous one-time request for your desired capacity. Thereafter, if capacity is diminished because of Spot interruptions, the fleet does not attempt to replenish Spot Instances, nor does it submit requests in alternative Spot Instance pools if capacity is unavailable.

`maintain`

(Default) If you configure the request type as `maintain`, EC2 Fleet places an asynchronous request for your desired capacity, and maintains capacity by automatically replenishing any interrupted Spot Instances.

You cannot modify the target capacity of an `instant` or `request` EC2 Fleet request after it's been submitted. To change the target capacity of an `instant` or `request` fleet request, delete the fleet and create a new one.

All three types of requests benefit from an allocation strategy. For more information, see [Allocation Strategies for Spot Instances \(p. 336\)](#).

Allocation Strategies for Spot Instances

The allocation strategy for your EC2 Fleet determines how it fulfills your request for Spot Instances from the possible Spot Instance pools represented by its launch specifications. The following are the allocation strategies that you can specify in your fleet:

`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 \(p. 344\)](#).

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": "MyFleetLaunchTemplate",  
                "LaunchTemplateVersion": "1",  
                "Type": "ON_DEMAND",  
                "Weight": 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 \(p. 349\)](#).

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 \(p. 344\)](#).

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 \(p. 349\)](#).

Fulfillment

The allocation strategy determines that the On-Demand capacity is always fulfilled, while the balance of the target capacity is fulfilled as Spot if there is capacity and availability.

Managing an EC2 Fleet

To use an EC2 Fleet, you create a request that includes the total target capacity, On-Demand capacity, Spot capacity, one or more launch specifications for the instances, and the maximum price that you are willing to pay. The fleet request must include a launch template that defines the information that the fleet needs to launch an instance, such as an AMI, instance type, subnet or Availability Zone, and one or more security groups. You can specify launch specification overrides for the instance type, subnet, Availability Zone, and maximum price you're willing to pay, and you can assign weighted capacity to each launch specification override.

If your fleet includes Spot Instances, Amazon EC2 can attempt to maintain your fleet target capacity as Spot prices change.

An EC2 Fleet request remains active until it expires or you delete it. When you delete a fleet, you may specify whether deletion terminates the instances in that fleet.

Contents

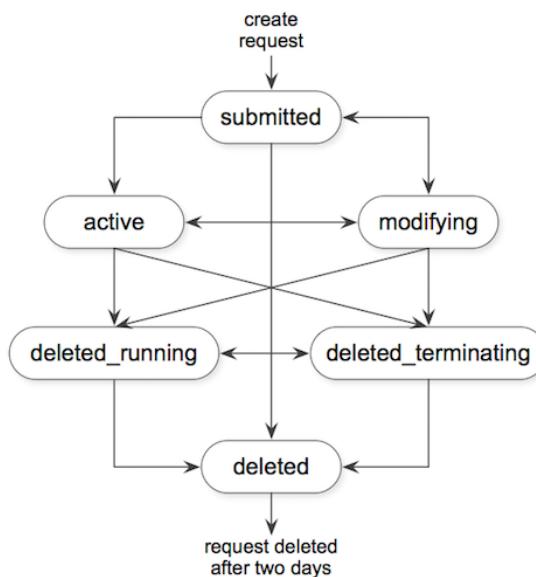
- [EC2 Fleet Request States \(p. 343\)](#)
- [EC2 Fleet Prerequisites \(p. 344\)](#)
- [EC2 Fleet Health Checks \(p. 345\)](#)
- [Generating an EC2 Fleet JSON Configuration File \(p. 346\)](#)
- [Creating an EC2 Fleet \(p. 349\)](#)
- [Tagging an EC2 Fleet \(p. 352\)](#)
- [Monitoring Your EC2 Fleet \(p. 342\)](#)
- [Modifying an EC2 Fleet \(p. 354\)](#)
- [Deleting an EC2 Fleet \(p. 355\)](#)
- [EC2 Fleet Example Configurations \(p. 356\)](#)

EC2 Fleet Request States

An EC2 Fleet request can be in one of the following states:

- **submitted** – The EC2 Fleet request is being evaluated and Amazon EC2 is preparing to launch the target number of instances, which can include On-Demand Instances, Spot Instances, or both.
- **active** – The EC2 Fleet request has been validated and Amazon EC2 is attempting to maintain the target number of running instances. The request remains in this state until it is modified or deleted.
- **modifying** – The EC2 Fleet request is being modified. The request remains in this state until the modification is fully processed or the request is deleted. Only a `maintain` request type can be modified. This state does not apply to other request types.
- **deleted_running** – The EC2 Fleet request is deleted and does not launch additional instances. Its existing instances continue to run until they are interrupted or terminated. The request remains in this state until all instances are interrupted or terminated.
- **deleted_terminating** – The EC2 Fleet request is deleted and its instances are terminating. The request remains in this state until all instances are terminated.
- **deleted** – The EC2 Fleet is deleted and has no running instances. The request is deleted two days after its instances are terminated.

The following illustration represents the transitions between the EC2 Fleet request states. If you exceed your fleet limits, the request is deleted immediately.



EC2 Fleet Prerequisites

To create an EC2 Fleet, the following prerequisites must be in place.

Launch Template

A launch template includes information about the instances to launch, such as the instance type, Availability Zone, and the maximum price that you are willing to pay. For more information, see [Launching an Instance from a Launch Template \(p. 321\)](#).

Service-Linked Role for EC2 Fleet

The **AWSServiceRoleForEC2Fleet** role grants the EC2 Fleet permission to request, launch, terminate, and tag instances on your behalf. Amazon EC2 uses this service-linked role to complete the following actions:

- **ec2: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.

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. Choose **Create role**.
4. On the **Select type of trusted entity** page, choose **EC2 - Fleet** and then choose **Next: Permissions**.
5. On the next page, choose **Next:Review**.
6. On the **Review** page, choose **Create role**.

If you no longer need to use 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 and IAM Users

If your IAM users will create or manage an EC2 Fleet, be sure to grant them the required permissions as follows.

To grant an IAM user permissions for EC2 Fleet

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Policies**.
3. Choose **Create policy**.
4. On the **Create policy** page, choose the **JSON** tab, replace the text with the following, and choose **Review policy**.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "ec2:*"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "iam>ListRoles",
            "iam>PassRole",
            "iam>ListInstanceProfiles"
        ],
        "Resource": "*"
    }
]
```

The `ec2:*` grants an IAM user permission to call all Amazon EC2 API actions. To limit the user to specific Amazon EC2 API actions, specify those actions instead.

An IAM user must have permission to call the `iam>ListRoles` action to enumerate existing IAM roles, the `iam>PassRole` action to specify the EC2 Fleet role, and the `iam>ListInstanceProfiles` action to enumerate existing instance profiles.

(Optional) To enable an IAM user to create roles or instance profiles using the IAM console, you must also add the following actions to the policy:

- `iam>AddRoleToInstanceProfile`
- `iam>AttachRolePolicy`
- `iam>CreateInstanceProfile`
- `iam>CreateRole`
- `iam>GetRole`
- `iam>ListPolicies`

5. On the **Review policy** page, enter a policy name and description, and choose **Create policy**.
6. In the navigation pane, choose **Users** and select the user.
7. On the **Permissions** tab, choose **Add permissions**.
8. Choose **Attach existing policies directly**. Select the policy that you created earlier and choose **Next: Review**.
9. Choose **Add permissions**.

EC2 Fleet Health Checks

EC2 Fleet checks the health status of the instances in the fleet every two minutes. The health status of an instance is either healthy or unhealthy. The fleet determines the health status of an instance using the status checks provided by Amazon EC2. If the status of either the instance status check or the system status check is impaired for three consecutive health checks, the health status of the instance is unhealthy. Otherwise, the health status is healthy. For more information, see [Status Checks for Your Instances \(p. 490\)](#).

You can configure your EC2 Fleet to replace unhealthy instances. After enabling health check replacement, an instance is replaced after its health status is reported as unhealthy. The fleet could go below its target capacity for up to a few minutes while an unhealthy instance is being replaced.

Requirements

- Health check replacement is supported only with EC2 Fleets that maintain a target capacity, not with one-time fleets.
- You can configure your EC2 Fleet to replace unhealthy instances only when you create it.
- IAM users can use health check replacement only if they have permission to call the `ec2:DescribeInstanceStatus` action.

Generating an EC2 Fleet JSON Configuration File

To create an EC2 Fleet, you need only specify the launch template, total target capacity, and whether the default purchasing 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": "lowestPrice",  
        "InstanceInterruptionBehavior": "hibernate",  
        "InstancePoolsToUseCount": 0  
    },  
    "OnDemandOptions": {  
        "AllocationStrategy": "prioritized"  
    },  
    "ExcessCapacityTerminationPolicy": "termination",  
    "LaunchTemplateConfigs": [  
        {  
            "LaunchTemplateSpecification": {  
                "LaunchTemplateId": "",  
                "LaunchTemplateName": "",  
                "Version": ""  
            },  
            "Overrides": [  
                {  
                    "InstanceType": "t2.micro",  
                    "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 `lowestPrice` and `diversified`. The default is `lowestPrice`. Specify the allocation strategy that meets your needs. For more information, see [Allocation Strategies for Spot Instances \(p. 336\)](#).

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 `lowestPrice`, 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 `lowestPrice`.

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 \(p. 321\)](#).

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 \(p. 321\)](#).

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.

To create a new VPC, go the Amazon VPC console. When you are done, return to the JSON file and enter the new subnet ID.

AvailabilityZone

(Optional) The Availability Zone in which to launch the instances. The default is to let AWS choose the zones for your instances. If you prefer, you can specify specific zones. If entered, this value overrides the launch template.

Specify one or more Availability Zones. If you have more than one subnet in a zone, specify the appropriate subnet. To add subnets, go to the Amazon VPC console. When you are done, return to the JSON file and enter the new subnet ID.

WeightedCapacity

(Optional) The number of units provided by the specified instance type. If entered, this value overrides the launch template.

Priority

The priority for the launch template override. If **AllocationStrategy** is set to prioritized, EC2 Fleet uses priority to determine which launch template override to use first in fulfilling On-Demand capacity. The highest priority is launched first. Valid values are whole numbers starting at 0. The lower the number, the higher the priority. If no number is set, the override has the lowest priority.

TotalTargetCapacity

The number of instances to launch. You can choose instances or performance characteristics that are important to your application workload, such as vCPUs, memory, or storage. If the request type is maintain, you can specify a target capacity of 0 and add capacity later.

OnDemandTargetCapacity

(Optional) The number of On-Demand Instances to launch. This number must be less than the **TotalTargetCapacity**.

SpotTargetCapacity

(Optional) The number of Spot Instances to launch. This number must be less than the `TotalTargetCapacity`.

DefaultTargetCapacityType

If the value for `TotalTargetCapacity` is higher than the combined values for `OnDemandTargetCapacity` and `SpotTargetCapacity`, the difference is launched as the instance purchasing 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 the EC2 Fleet submits a synchronous one-time request for your desired capacity (`instant`), or an asynchronous one-time request for your desired capacity, but with no attempt to maintain the capacity or to submit requests in alternative capacity pools if capacity is unavailable (`request`), or submits an asynchronous request for your desired capacity and continues to maintain your desired capacity by replenishing interrupted Spot Instances (`maintain`). Valid values are `instant`, `request`, and `maintain`. The default value is `maintain`. For more information, see [EC2 Fleet Request Types \(p. 336\)](#).

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 \(p. 323\)](#). For information about tagging after launch, see [Tagging Your Resources \(p. 892\)](#).

[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 \(p. 347\)](#).

EC2 Fleets can only be created using the AWS CLI.

To create an EC2 Fleet (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 \(p. 356\)](#).

The following is example output for a fleet of type `request` or `maintain`:

```
{  
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"  
}
```

The following is example output for a fleet of type `instant` that launched the target capacity:

```
{  
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",  
    "Errors": [],  
    "Instances": [  
        {  
            "LaunchTemplateAndOverrides": {  
                "LaunchTemplateSpecification": {  
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",  
                    "Version": "1"  
                },  
                "Overrides": {  
                    "InstanceType": "c5.large",  
                    "AvailabilityZone": "us-east-1a"  
                }  
            },  
            "Lifecycle": "on-demand",  
            "InstanceIds": [  
                "i-1234567890abcdef0",  
                "i-9876543210abcdef9"  
            ],  
            "InstanceType": "c5.large",  
            "Platform": null  
        },  
        {  
            "LaunchTemplateAndOverrides": {  
                "LaunchTemplateSpecification": {  
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",  
                    "Version": "1"  
                },  
                "Overrides": {  
                    "InstanceType": "c4.large",  
                    "AvailabilityZone": "us-east-1a"  
                }  
            },  
            "Lifecycle": "on-demand",  
            "InstanceIds": [  
                "i-5678901234abcdef0",  
                "i-5432109876abcdef9"  
            ],  
            "InstanceType": "c4.large",  
            "Platform": null  
        },  
    ]  
}
```

The following is example output for a fleet of type `instant` that launched part of the target capacity with errors for instances that were not launched:

```
{  
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",  
    "Errors": [  
        {  
            "LaunchTemplateAndOverrides": {  
                "LaunchTemplateSpecification": {  
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",  
                    "Version": "1"  
                },  
                "Overrides": {  
                    "InstanceType": "c4.xlarge",  
                    "AvailabilityZone": "us-east-1a",  
                }  
            },  
            "Lifecycle": "on-demand",  
            "ErrorCode": "InsufficientInstanceCapacity",  
            "ErrorMessage": "",  
            "InstanceType": "c4.xlarge",  
            "Platform": null  
        },  
    ],  
    "Instances": [  
        {  
            "LaunchTemplateAndOverrides": {  
                "LaunchTemplateSpecification": {  
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",  
                    "Version": "1"  
                },  
                "Overrides": {  
                    "InstanceType": "c5.large",  
                    "AvailabilityZone": "us-east-1a"  
                }  
            },  
            "Lifecycle": "on-demand",  
            "InstanceIds": [  
                "i-1234567890abcdef0",  
                "i-9876543210abcdef9"  
            ],  
            "InstanceType": "c5.large",  
            "Platform": null  
        },  
    ]  
}
```

The following is example output for a fleet of type `instant` that launched no instances:

```
{  
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",  
    "Errors": [  
        {  
            "LaunchTemplateAndOverrides": {  
                "LaunchTemplateSpecification": {  
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",  
                    "Version": "1"  
                },  
                "Overrides": {  
                    "InstanceType": "c4.xlarge",  
                    "AvailabilityZone": "us-east-1a",  
                }  
            },  
            "Lifecycle": "on-demand",  
            "Message": "No instances were launched because there was not enough capacity available."  
        }  
    ]  
}
```

```
"ErrorCode": "InsufficientCapacity",
"ErrorMessage": "",
"InstanceType": "c4.xlarge",
"Platform": null
},
{
"LaunchTemplateAndOverrides": {
"LaunchTemplateSpecification": {
"LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
"Version": "1"
},
"Overrides": {
"InstanceType": "c5.large",
"AvailabilityZone": "us-east-1a",
}
},
"Lifecycle": "on-demand",
"ErrorCode": "InsufficientCapacity",
"ErrorMessage": "",
"InstanceType": "c5.large",
"Platform": null
},
],
"Instances": []
}
```

Tagging an EC2 Fleet

To help categorize and manage your EC2 Fleet requests, you can tag them with custom metadata. For more information, see [Tagging Your Amazon EC2 Resources \(p. 890\)](#).

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 \(p. 346\)](#) 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 \(p. 323\)](#) referenced in the EC2 Fleet request.

To tag an existing EC2 Fleet request and instance (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 (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 (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 (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 \(p. 347\)](#).

1. [Launch Spot Instances as the default purchasing model \(p. 356\)](#)
2. [Launch On-Demand Instances as the default purchasing model \(p. 356\)](#)
3. [Launch On-Demand Instances as the primary capacity \(p. 357\)](#)
4. [Launch Spot Instances using the lowestPrice allocation strategy \(p. 357\)](#)

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 `lowestPrice`, is used. The following example uses the `lowestPrice` 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"  
}
```

Connecting to Your Windows Instance

Amazon EC2 instances created from most Windows Amazon Machine Images (AMIs) enable you to connect using Remote Desktop. Remote Desktop uses the Remote Desktop Protocol (RDP) and enables you to connect to and use your instance in the same way you use a computer sitting in front of you. It is available on most editions of Windows and available for Mac OS.

For information about connecting to a Linux instance, see [Connect to Your Linux Instance](#) in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- [Prerequisites \(p. 359\)](#)
- [Connect to Your Windows Instance \(p. 360\)](#)
- [Connect to a Windows Instance Using Its IPv6 Address \(p. 361\)](#)
- [Transfer Files to Windows Instances \(p. 362\)](#)

Prerequisites

- **Install an RDP client**

- [Windows] Windows includes an RDP client by default. To verify, type `mstsc` at a Command Prompt window. If your computer doesn't recognize this command, see the [Windows home page](#) and search for the download for the Microsoft Remote Desktop app.
- [Mac OS X] Download the Microsoft Remote Desktop app from the Mac App Store.
- [Linux] Use `rdesktop`.

- **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 \(p. 645\)](#).

- **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.

- **Enable inbound RDP traffic from your IP address to your instance**

Ensure that the security group associated with your instance allows incoming RDP traffic from your IP address. The default security group does not allow incoming RDP traffic by default. For more information, see [Authorizing Inbound Traffic for Your Windows Instances \(p. 640\)](#).

- For the best experience using Internet Explorer, run the latest version.

Connect to Your Windows Instance

To connect to a Windows instance, you must retrieve the initial administrator password and then specify this password when you connect to your instance using Remote Desktop.

The name of the administrator account depends on the language of the operating system. For example, for English, it's Administrator, for French it's Administrateur, and for Portuguese it's Administrador. For more information, see [Localized Names for Administrator Account in Windows](#) in the Microsoft TechNet Wiki.

If you've joined your instance to a domain, you can connect to your instance using domain credentials you've defined in AWS Directory Service. On the Remote Desktop login screen, instead of using the local computer name and the generated password, use the fully-qualified user name for the administrator (for example, `corp.example.com\Admin`) and the password for this account.

The license for the Windows Server operating system (OS) allows two simultaneous remote connections for administrative purposes. The license for Windows Server is included in the price of your Windows instance. If you need more than two simultaneous remote connections, you must purchase a Remote Desktop Services (RDS) license. If you attempt a third connection, an error occurs. For more information, see [Configure the Number of Simultaneous Remote Connections Allowed for a Connection](#).

To connect to your Windows instance using an RDP client

1. In the Amazon EC2 console, select the instance, and then choose **Connect**.
2. In the **Connect To Your Instance** dialog box, choose **Get Password** (it will take a few minutes after the instance is launched before the password is available).
3. Choose **Browse** and navigate to the private key file you created when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file into the **Contents** field.
4. Choose **Decrypt Password**. The console displays the default administrator password for the instance in the **Connect To Your Instance** dialog box, replacing the link to **Get Password** shown previously with the actual password.
5. Record the default administrator password, or copy it to the clipboard. You need this password to connect to the instance.
6. Choose **Download Remote Desktop File**. Your browser prompts you to either open or save the .rdp file. Either option is fine. When you have finished, you can choose **Close** to dismiss the **Connect To Your Instance** dialog box.
 - If you opened the .rdp file, you'll see the **Remote Desktop Connection** dialog box.
 - If you saved the .rdp file, navigate to your downloads directory, and open the .rdp file to display the dialog box.
7. You may get a warning that the publisher of the remote connection is unknown. You can continue to connect to your instance.
8. When prompted, log in to the instance, using the administrator account for the operating system and the password that you recorded or copied previously. If your **Remote Desktop Connection** already has an administrator account set up, you might have to choose the **Use another account** option and type the user name and password manually.

Note

Sometimes copying and pasting content can corrupt data. If you encounter a "Password Failed" error when you log in, try typing in the password manually.

9. Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. Use the following steps to verify the identity of the remote computer, or simply choose **Yes** or **Continue** to continue if you trust the certificate.

- a. If you are using **Remote Desktop Connection** from a Windows PC, choose **View certificate**. If you are using **Microsoft Remote Desktop** on a Mac, choose **Show Certificate**.
- b. Choose the **Details** tab, and scroll down to the **Thumbprint** entry on a Windows PC, or the **SHA1 Fingerprints** entry on a Mac. This is the unique identifier for the remote computer's security certificate.
- c. In the Amazon EC2 console, select the instance, choose **Actions**, and then choose **Get System Log**.
- d. In the system log output, look for an entry labeled **RDPCERTIFICATE-THUMBPRINT**. If this value matches the thumbprint or fingerprint of the certificate, you have verified the identity of the remote computer.
- e. If you are using **Remote Desktop Connection** from a Windows PC, return to the **Certificate** dialog box and choose **OK**. If you are using **Microsoft Remote Desktop** on a Mac, return to the **Verify Certificate** and choose **Continue**.
- f. [Windows] Choose **Yes** in the **Remote Desktop Connection** window to connect to your instance.

[Mac OS] Log in as prompted, using the default administrator account and the default administrator password that you recorded or copied previously. Note that you might need to switch spaces to see the login screen. For more information about spaces, see support.apple.com/en-us/HT204100.

- g. If you receive an error while attempting to connect to your instance, see [Remote Desktop can't connect to the remote computer \(p. 986\)](#).

After you connect, we recommend that you do the following:

- Change the administrator password from the default value. You change the password while logged on to the instance itself, just as you would on any other Windows Server.
- Create another user account with administrator privileges on the instance. Another account with administrator privileges is a safeguard if you forget the administrator password or have a problem with the administrator account. The user account must have permission to access the instance remotely. Open **System Properties**, choose **Remote**, and add the user to the **Remote Desktop Users** group.

Connect to a Windows Instance Using Its IPv6 Address

If you've enabled your VPC for IPv6 and assigned an IPv6 address to your Windows instance, you can use an RDP client to connect to your instance using its IPv6 address instead of a public IPv4 address or public DNS hostname. For more information, see [IPv6 Addresses \(p. 645\)](#).

To connect to your Windows instance using its IPv6 address

1. In the Amazon EC2 console, select the instance, and then choose **Connect**.
2. In the **Connect To Your Instance** dialog box, choose **Get Password** (it will take a few minutes after the instance is launched before the password is available).
3. Choose **Browse** and navigate to the private key file you created when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file into the **Contents** field.
4. Choose **Decrypt Password**.
5. Copy the default administrator password. You need this password to connect to the instance.
6. Open the RDP client on your computer.
7. [Windows] For the RDP client on a Windows computer, choose **Show Options** and do the following:
 - For **Computer**, type the IPv6 address of your Windows instance, for example, 2001:db8:1234:1a00:9691:9503:25ad:1761.

- For **User name**, enter **Administrator**.
- Choose **Connect**.

[Mac OS X] For the Microsoft Remote Desktop app, choose **New** and do the following:

- For **PC Name**, enter the IPv6 address of your Windows instance; for example, 2001:db8:1234:1a00:9691:9503:25ad:1761.
 - For **User name**, enter **Administrator**.
 - Close the dialog box. Under **My Desktops**, select the connection and choose **Start**.
8. Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. Use the following steps to verify the identity of the remote computer, or simply choose **Yes** or **Continue** to continue if you trust the certificate.
 9. When prompted, enter the password that you recorded or copied previously.

Transfer Files to Windows Instances

You can work with your Windows instance the same way that you would work with any Windows server. For example, you can transfer files between a Windows instance and your local computer using the local file sharing feature of the Microsoft Remote Desktop Connection software. If you enable this option, you can access your local files from your Windows instances. You can access local files on hard disk drives, DVD drives, portable media drives, and mapped network drives.

To make local devices and resources available to a remote session on Windows, map the remote session drive to your local drive:

1. Open the Remote Desktop Connection client.
2. Choose **Show Options**.
3. Select the **Local Resources** tab.
4. Under the Local Devices and resources section, choose **More...**
5. Open Drives and select the local drive to map to your Windows instance.
6. Choose **OK**.
7. Choose **Connect** to connect to your Windows instance.

For more information on making local devices available to a remote session on a Mac computer, see:

[Get Started with Remote Desktop on Mac](#)

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 \(p. 363\)](#) 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 full instance hour, even if you make this transition multiple times within a single hour.

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 \(p. 368\)](#). If you'd rather hibernate the instance, see [Hibernate Your Instance \(p. 365\)](#). For more information, see [Differences Between Reboot, Stop, Hibernate, and Terminate \(p. 313\)](#).

Contents

- [Overview \(p. 363\)](#)
- [Stopping and Starting Your Instances \(p. 364\)](#)
- [Modifying a Stopped Instance \(p. 364\)](#)
- [Troubleshooting \(p. 365\)](#)

Overview

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.
- 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.
- 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 \(p. 383\)](#) 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 \(p. 708\)](#).

For more information, see [Differences Between Reboot, Stop, Hibernate, and Terminate \(p. 313\)](#).

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 \(p. 370\)](#).

To stop and start an Amazon EBS-backed instance using the console

1. In the navigation pane, choose **Instances**, and select the instance.
2. 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. To keep data from instance store volumes, be sure to back it up to persistent storage.

3. In the confirmation dialog box, choose **Yes, Stop**. It can take a few minutes for the instance to stop.
4. While your instance is stopped, you can modify certain instance attributes. For more information, see [Modifying a Stopped Instance \(p. 364\)](#).
5. To restart the stopped instance, select the instance, and choose **Actions, Instance State, Start**.
6. In the confirmation dialog box, choose **Yes, Start**. It can take a few minutes for the instance to enter the `running` state.

To stop and start an Amazon EBS-backed instance using the command line

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

- `stop-instances` and `start-instances` (AWS CLI)
- `Stop-EC2Instance` and `Start-EC2Instance` (AWS Tools for Windows PowerShell)

Modifying a Stopped Instance

You can change the instance type, user data, and EBS-optimization attributes of a stopped instance using the AWS Management Console or the command line interface. You can't use the AWS Management Console to modify the `DeleteOnTermination`, kernel, or RAM disk attributes.

To modify an instance attribute

- To change the instance type, see [Changing the Instance Type \(p. 182\)](#).
- To change the user data for your instance, see [Working with Instance User Data \(p. 455\)](#).
- To enable or disable EBS-optimization for your instance, see [Modifying EBS-Optimization \(p. 815\)](#).
- To change the `DeleteOnTermination` attribute of the root volume for your instance, see [Updating the Block Device Mapping of a Running Instance \(p. 862\)](#).

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](#) in the *Amazon EC2 User Guide for Linux Instances*.

Hibernate Your Instance

When you hibernate an instance, we signal the operating system to perform hibernation (suspend-to-disk), which saves the contents from the instance memory (RAM) to your Amazon EBS root volume. We persist the instance's Amazon EBS root volume and any attached Amazon EBS data volumes. When you restart your instance, the Amazon EBS root volume is restored to its previous state, the RAM contents are reloaded, and the processes that were previously running on the instance are resumed. Previously attached data volumes are reattached and the instance retains its instance ID. For more information, see [Hibernate Your Instance](#) in the *Amazon EC2 User Guide for Linux Instances*.

Important

Hibernation is currently not supported on Windows instances.

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 hour, 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 \(p. 494\)](#).

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.

If Windows is installing updates on your instance, we recommend that you do not reboot or shut down your instance using the Amazon EC2 console or the command line until all the updates are installed. When you use the Amazon EC2 console or the command line to reboot or shut down your instance, there is a risk that your instance will be hard rebooted. A hard reboot while updates are being installed could throw your instance into an unstable state.

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.

Contents

- [Identifying Instances Scheduled for Retirement \(p. 366\)](#)
- [Working with Instances Scheduled for Retirement \(p. 367\)](#)

For more information about types of instance events, see [Scheduled Events for Your Instances \(p. 494\)](#).

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 \(p. 366\)](#).
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-EC2InstanceState](#) (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	Create an EBS-backed AMI from your instance so that you have a backup. Wait for the scheduled retirement date - when the instance is stopped - or stop the instance yourself before the retirement date. You can start the instance again at any time. For more information 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. 362) .
EBS	Create an EBS-backed AMI from your instance, and launch a replacement instance. For more information, see Creating a Custom Windows AMI (p. 65) .

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, or hibernate it, see [Stop and Start Your Instance \(p. 362\)](#) or [Hibernate Your Instance \(p. 365\)](#). For more information, see [Differences Between Reboot, Stop, Hibernate, and Terminate \(p. 313\)](#).

Contents

- [Instance Termination \(p. 368\)](#)
- [Terminating an Instance \(p. 368\)](#)
- [Enabling Termination Protection for an Instance \(p. 369\)](#)
- [Changing the Instance Initiated Shutdown Behavior \(p. 370\)](#)
- [Preserving Amazon EBS Volumes on Instance Termination \(p. 371\)](#)

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 \(p. 371\)](#).

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 \(p. 369\)](#).

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 \(p. 370\)](#).

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 \(p. 370\)](#).

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 \(p. 276\)](#).

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)

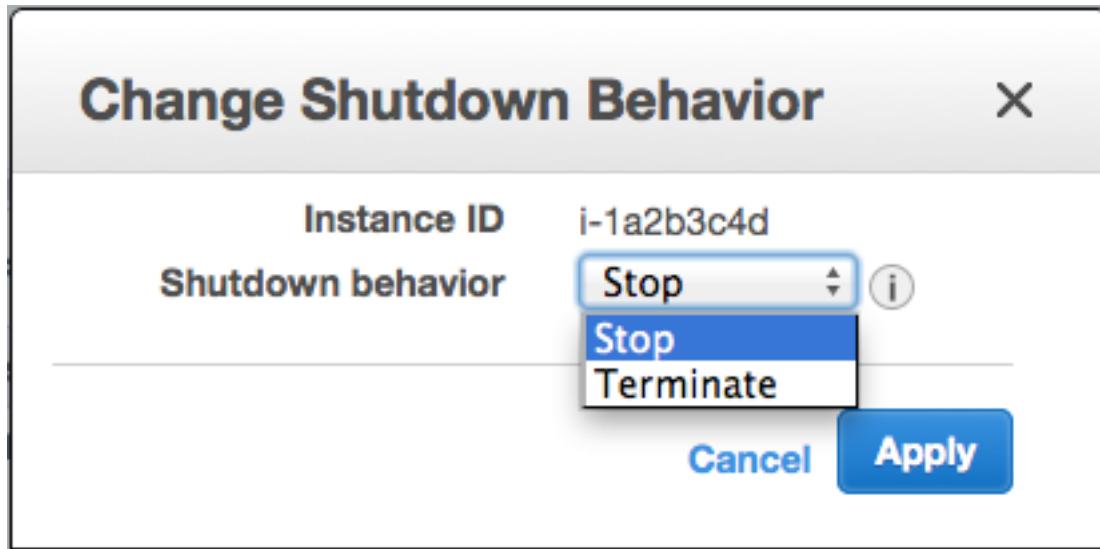
Changing the Instance Initiated Shutdown Behavior

By default, when you initiate a shutdown from an Amazon EBS-backed instance (using a command such as **shutdown** or **poweroff**), the instance stops (Note that **halt** does not issue a **poweroff** command and, if used, the instance will not terminate; instead, it will place the CPU into HLT and the instance will remain running). You can change this behavior using the `InstanceInitiatedShutdownBehavior` attribute for the instance so that it terminates instead. You can update this attribute while the instance is running or stopped.

You can update the `InstanceInitiatedShutdownBehavior` attribute using the Amazon EC2 console or the command line. The `InstanceInitiatedShutdownBehavior` attribute only applies when you perform a shutdown from the operating system of the instance itself; it does not apply when you stop an instance using the `StopInstances` API or the Amazon EC2 console.

To change the shutdown behavior of an instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, 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 \(p. 863\)](#).

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 \(p. 371\)](#)
- [Changing the Root Volume to Persist at Launch Using the Command Line \(p. 372\)](#)
- [Changing the Root Volume of a Running Instance to Persist Using the Command Line \(p. 372\)](#)

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  
    }  
  }  
]
```

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 \(p. 517\)](#). 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 \(p. 494\)](#).

The recover action is supported only on instances with the following characteristics:

- Use one of the following instance types: C3, C4, C5, C5n, M3, M4, M5, M5a, R3, R4, R5, R5a, T2, T3, X1, or X1e
- Use default or dedicated instance tenancy
- Use EBS volumes only (do not configure instance store volumes).

If your instance has a public IPv4 address, it retains the public IPv4 address after recovery.

Configuring Your Windows Instance

A Windows instance is a virtual server running Windows Server in the cloud.

After you have successfully launched and logged into your instance, you can make changes to it so that it's configured to meet the needs of a specific application. The following are some common tasks to help you get started.

Contents

- [Configuring a Windows Instance Using EC2Launch \(p. 374\)](#)
- [Configuring a Windows Instance Using the EC2Config Service \(p. 383\)](#)
- [Paravirtual Drivers for Windows Instances \(p. 409\)](#)
- [AWS NVMe Drivers for Windows Instances \(p. 424\)](#)
- [Optimizing CPU Options \(p. 425\)](#)
- [Setting the Time for a Windows Instance \(p. 435\)](#)
- [Setting Passwords for Windows Instances \(p. 439\)](#)
- [Adding Windows Components Using Installation Media \(p. 440\)](#)
- [Configuring a Secondary Private IPv4 Address for Your Windows Instance \(p. 443\)](#)
- [Running Commands on Your Windows Instance at Launch \(p. 447\)](#)
- [Instance Metadata and User Data \(p. 452\)](#)

Configuring a Windows Instance Using EC2Launch

EC2Launch is a set of Windows PowerShell scripts that replaces the EC2Config service on Windows Server 2016 AMIs.

Contents

- [EC2Launch Tasks \(p. 374\)](#)
- [Installing the Latest Version of EC2Launch \(p. 375\)](#)
- [Verify the EC2Launch Version \(p. 375\)](#)
- [EC2Launch Directory Structure \(p. 375\)](#)
- [Configuring EC2Launch \(p. 376\)](#)
- [Using Sysprep with EC2Launch \(p. 379\)](#)
- [EC2Launch Version History \(p. 382\)](#)

EC2Launch Tasks

EC2Launch performs the following tasks by default during the initial instance boot:

- Sets up new wallpaper that renders information about the instance.
- Sets the computer name.
- Sends instance information to the Amazon EC2 console.
- Sends the RDP certificate thumbprint to the EC2 console.
- Sets a random password for the administrator account.
- Adds DNS suffixes.
- Dynamically extends the operating system partition to include any unpartitioned space.
- Executes user data (if specified). For more information about specifying user data, see [Working with Instance User Data \(p. 455\)](#).
- Sets persistent static routes to reach the metadata service and KMS servers.

Important

If a custom AMI is created from this instance, these routes are captured as part of the OS configuration and any new instances launched from the AMI will retain the same routes,

regardless of subnet placement. In order to update the routes, see [Updating metadata/KMS routes for Server 2016 when launching a custom AMI \(p. 381\)](#).

The following tasks help to maintain backward compatibility with the EC2Config service. You can also configure EC2Launch to perform these tasks during startup:

- Initialize secondary EBS volumes.
- Send Windows Event logs to the EC2 console logs.
- Send the *Windows is ready to use* message to the EC2 console.

For more information about Windows Server 2016, see [What's New with Windows Server 2016](#) on Microsoft.com.

Installing the Latest Version of EC2Launch

Use the following procedure to download and install the latest version of EC2Launch on your instances.

To download and install the latest version of EC2Launch

1. If you have already installed and configured EC2Launch on an instance, make a backup of the EC2Launch configuration file. The installation process does not preserve changes in this file. By default, the file is located in the C:\ProgramData\Amazon\EC2-Windows\Launch\Config directory.
2. Download [EC2-Windows-Launch.zip](#) to a directory on the instance.
3. Download [install.ps1](#) to the same directory where you downloaded EC2-Windows-Launch.zip.
4. Run `install.ps1`
5. If you made a backup of the EC2Launch configuration file, copy it to the C:\ProgramData\Amazon\EC2-Windows\Launch\Config directory.

Verify the EC2Launch Version

Use the following Windows PowerShell command to verify the installed version of EC2Launch.

```
PS C:\> Test-ModuleManifest -Path "C:\ProgramData\Amazon\EC2-Windows\Launch\Module\Ec2Launch.psd1" | Select Version
```

EC2Launch Directory Structure

EC2Launch is installed by default on Windows Server 2016 AMIs in the root directory C:\ProgramData\Amazon\EC2-Windows\Launch.

Note

By default, Windows hides files and folders under C:\ProgramData. To view EC2Launch directories and files, you must either type the path in Windows Explorer or change the folder properties to show hidden files and folders.

The Launch directory contains the following subdirectories.

- Scripts — Contains the PowerShell scripts that make up EC2Launch.
- Module — Contains the module for building scripts related to Amazon EC2.
- Config — Contains script configuration files that you can customize.
- Sysprep — Contains Sysprep resources.

- **Settings** — Contains an application for the Sysprep graphical user interface.
- **Logs** — Contains log files generated by scripts.

Configuring EC2Launch

After your instance has been initialized the first time, you can configure EC2Launch to run again and perform different start-up tasks.

Tasks

- [Configure Initialization Tasks \(p. 376\)](#)
- [Schedule EC2Launch to Run on Every Boot \(p. 377\)](#)
- [Initialize Drives and Map Drive Letters \(p. 377\)](#)
- [Send Windows Event Logs to the EC2 Console \(p. 378\)](#)
- [Send Windows Is Ready Message After A Successful Boot \(p. 378\)](#)

Configure Initialization Tasks

Specify settings in the `LaunchConfig.json` file to enable or disable the following initialization tasks:

- Set the computer name.
- Set up new wallpaper.
- Add DNS suffix list.
- Extend the boot volume size.
- Set the administrator password.

To configure initialization settings

1. On the instance to configure, open the following file in a text editor: `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\LaunchConfig.json`.
2. Update the following settings as needed and save your changes. Provide a password in `adminPassword` only if `adminPasswordType` is `Specify`.

```
{  
    "setComputerName": false,  
    "setWallpaper": true,  
    "addDnsSuffixList": true,  
    "extendBootVolumeSize": true,  
    "handleUserData": true,  
    "adminPasswordType": "Random | Specify | DoNothing",  
    "adminPassword": "password that adheres to your security policy (optional)"  
}
```

The password types are defined as follows:

Random

EC2Launch generates a password and encrypts it using the user's key. The system disables this setting after the instance is launched so that this password persists if the instance is rebooted or stopped and started.

Specify

EC2Launch uses the password you specify in `adminPassword`. If the password does not meet the system requirements, EC2Launch generates a random password instead. The password is

stored in `LaunchConfig.json` as clear text and is deleted after Sysprep sets the administrator password. EC2Launch encrypts the password using the user's key.

DoNothing

EC2Launch uses the password you specify in the `unattend.xml` file. If you don't specify a password in `unattend.xml`, the administrator account is disabled.

3. In Windows PowerShell, run the following command to schedule the script to run as a Windows Scheduled Task. The script runs one time during the next boot and then disables these tasks from running again.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -  
Schedule
```

Schedule EC2Launch to Run on Every Boot

You can schedule EC2Launch to run on every boot instead of only the initial boot.

To enable EC2Launch to run on every boot:

1. Open Windows PowerShell and run the following command:

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -  
SchedulePerBoot
```

2. Or, run the executable with the following command:

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Settings\Ec2LaunchSettings.exe
```

Then select Run EC2Launch on every boot. You can specify that your EC2 instance Shutdown without Sysprep or Shutdown with Sysprep.

Note

When you enable EC2Launch to run on every boot, the following changes will be made to the `LaunchConfig.json` the next time EC2Launch runs:

- `AdminPasswordType` will be set back to `DoNothing` so that the password does not change on each boot.
- `HandleUserData` will be set back to `false` unless the user data has `persist` set to `true`. For more information about user data scripts, see [User Data Scripts](#) in the Amazon EC2 User Guide.

Similarly, if you do not want your password reset on the next boot, you should set `AdminPasswordType` to `DoNothing` before rebooting.

Initialize Drives and Map Drive Letters

Specify settings in the `DriveLetterMappingConfig.json` file to map drive letters to volumes on your EC2 instance. The script performs this operation if the drives have not already been initialized and partitioned.

To map drive letters to volumes

1. Open the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\DriveLetterMappingConfig.json` file in a text editor.

2. Specify the following volume settings and save your changes:

```
{  
    "driveLetterMapping": [  
        {  
            "volumeName": "sample volume",  
            "driveLetter": "H"  
        }  
    ]  
}
```

3. Open Windows PowerShell and use the following command to run the EC2Launch script that initializes the disks:

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
```

To initialize the disks each time the instance boots, add the `-Schedule` flag as follows:

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1 -Schedule
```

Send Windows Event Logs to the EC2 Console

Specify settings in the `EventLogConfig.json` file to send Windows Event logs to EC2 console logs.

To configure settings to send Windows Event logs

1. On the instance, open the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\EventLogConfig.json` file in a text editor.
2. Configure the following log settings and save your changes:

```
{  
    "events": [  
        {  
            "logName": "System",  
            "source": "An event source (optional)",  
            "level": "Error | Warning | Information",  
            "numEntries": 3  
        }  
    ]  
}
```

3. In Windows PowerShell, run the following command so that the system schedules the script to run as a Windows Scheduled Task each time the instance boots.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\SendEventLogs.ps1 -Schedule
```

The logs can take three minutes or more to appear in the EC2 console logs.

Send Windows Is Ready Message After A Successful Boot

The EC2Config service sent the "Windows is ready" message to the EC2 console after every boot. EC2Launch sends this message only after the initial boot. For backwards compatibility with the EC2Config service, you can schedule EC2Launch to send this message after every boot. On the instance, open Windows PowerShell and run the following command. The system schedules the script to run as a Windows Scheduled Task.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\SendWindowsIsReady.ps1 -Schedule
```

Using Sysprep with EC2Launch

Sysprep simplifies the process of duplicating a customized installation of Windows Server 2016. EC2Launch offers a default answer file and batch files for Sysprep that automate and secure the image-preparation process on your AMI. Modifying these files is optional. These files are located in the following directory by default: C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep.

Important

Do not use Sysprep to create an instance backup. Sysprep removes system-specific information. If you remove this information there might be unintended consequences for an instance backup.

The EC2Launch answer file and batch files for Sysprep include the following:

`Unattend.xml`

This is the default answer file. If you run `SysprepInstance.ps1` or choose **ShutdownWithSysprep** in the user interface, the system reads the setting from this file.

`BeforeSysprep.cmd`

Customize this batch file to run commands before EC2Launch runs Sysprep.

`SysprepSpecialize.cmd`

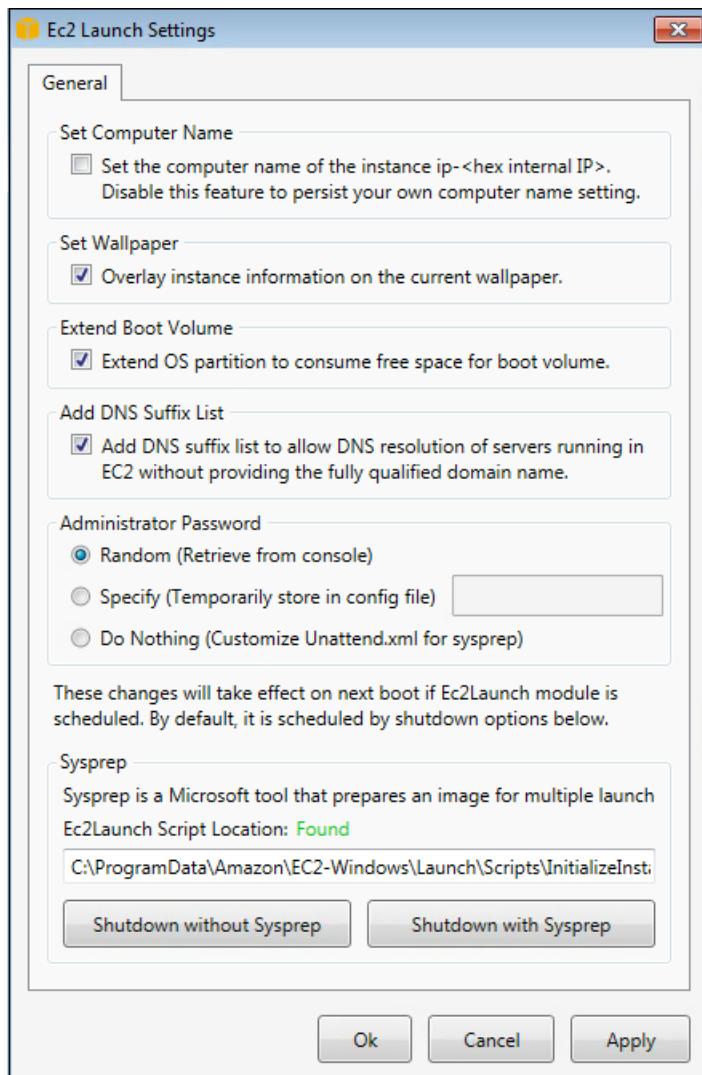
Customize this batch file to run commands during the Sysprep specialize phase.

Running Sysprep with EC2Launch

On the full installation of Windows Server 2016 (with a desktop experience), you can run Sysprep with EC2Launch manually or by using the **EC2 Launch Settings** application.

To run Sysprep using the EC2Launch Settings application

1. In the Amazon EC2 console, locate or create a Windows Server 2016 AMI.
2. Launch a Windows instance from the AMI.
3. Connect to your Windows instance and customize it.
4. Search for and run the **EC2LaunchSettings** application. It is located in the following directory by default: C:\ProgramData\Amazon\EC2-Windows\Launch\Settings.



5. Select or clear options as needed. These settings are stored in the `LaunchConfig.json` file.
6. For **Administrator Password**, do one of the following:
 - Choose **Random**. EC2Launch generates a password and encrypts it using the user's key. The system disables this setting after the instance is launched so that this password persists if the instance is rebooted or stopped and started.
 - Choose **Specify** and type a password that meets the system requirements. The password is stored in `LaunchConfig.json` as clear text and is deleted after Sysprep sets the administrator password. If you shut down now, the password is set immediately. EC2Launch encrypts the password using the user's key.
 - Choose **DoNothing** and specify a password in the `unattend.xml` file. If you don't specify a password in `unattend.xml`, the administrator account is disabled.
7. Choose **Shutdown with Sysprep**.

To manually run Sysprep using EC2Launch

1. In the Amazon EC2 console locate or create a Windows Server 2016, Datacenter edition AMI that you want to duplicate.

2. Launch and connect to your Windows instance.
3. Customize the instance.
4. Specify settings in the `LaunchConfig.json` file. This file is located in the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` directory by default.

For `adminPasswordType`, specify one of the following values:

`Random`

EC2Launch generates a password and encrypts it using the user's key. The system disables this setting after the instance is launched so that this password persists if the instance is rebooted or stopped and started.

`Specify`

EC2Launch uses the password you specify in `adminPassword`. If the password does not meet the system requirements, EC2Launch generates a random password instead. The password is stored in `LaunchConfig.json` as clear text and is deleted after Sysprep sets the administrator password. EC2Launch encrypts the password using the user's key.

`DoNothing`

EC2Launch uses the password you specify in the `unattend.xml` file. If you don't specify a password in `unattend.xml`, the administrator account is disabled.

5. (Optional) Specify settings in `unattend.xml` and other configuration files. If plan to attend to the installation, then you don't need to make changes in these files. The files are located in the following directory by default: `C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep`.
6. In Windows PowerShell, run `./InitializeInstance.ps1 -Schedule`. The script is located in the following directory, by default: `C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts`. This script schedules the instance to initialize during the next boot. You must run this script before you run the `SysprepInstance.ps1` script in the next step.
7. In Windows PowerShell, run `./SysprepInstance.ps1`. The script is located in the following directory by default: `C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts`.

You are logged off the instance and the instance shuts down. If you check the **Instances** page in the Amazon EC2 console, the instance state changes from `running` to `stopping`, and then to `stopped`. At this point, it is safe to create an AMI from this instance.

Updating metadata/KMS routes for Server 2016 when launching a custom AMI

To update metadata/KMS routes for Server 2016 when launching a custom AMI

1. Use EC2LaunchSettings GUI (`C:\ProgramData\Amazon\EC2-Windows\Launch\Settings\EC2LaunchSettings.exe`) to shut down with Sysprep.
2. Or, shut down without Sysprep before creating an AMI. This sets the EC2 Launch Initialize tasks to run at the next boot, which will set routes based on the subnet being launched into.
3. Or, manually reschedule EC2 Launch initialize tasks before creating an AMI from [PowerShell \(p. 376\)](#).

Important

Please take note of the default password reset behavior before rescheduling tasks.

EC2Launch Version History

Windows AMIs starting with Windows Server 2016 include a set of Windows Powershell scripts called EC2Launch. EC2Launch performs tasks during the initial instance boot. For information about the EC2Launch versions included in the Windows AMIs, see [see Managed AWS Windows AMIs \(p. 77\)](#).

To download and install the latest version of EC2Launch, see [Installing the Latest Version of EC2Launch \(p. 375\)](#).

The following table describes the released versions of EC2Launch.

Version	Details
1.3.2000760	<ul style="list-style-type: none">Added default configuration for RSS and Receive Queue settings for ENA devices.Disabled hibernation during Sysprep.
1.3.2000630	<ul style="list-style-type: none">Added route 169.254.169.253/32 for DNS server.Added filter of setting Admin user.Improvements made to instance hibernation.Added option to schedule EC2Launch to run on every boot.
1.3.2000430.0	<ul style="list-style-type: none">Added route 169.254.169.123/32 to AMZN time service.Added route 169.254.169.249/32 to GRID license service.Added timeout of 25 seconds when attempting to start SSM.
1.3.200039.0	<ul style="list-style-type: none">Fixed improper drive lettering for EBS NVME volumes.Added additional logging for NVME driver versions.
1.3.2000080	Fixed minor issues.
1.3.610	Fixed issue with redirecting output and errors to files from user data.
1.3.590	<ul style="list-style-type: none">Added missing instances types in the wallpaper.Fixed an issue with drive letter mapping and disk installation.
1.3.580	<ul style="list-style-type: none">Fixed Get-Metadata to use the default system proxy settings for web requests.Added a special case for NVMe in disk initialization.Fixed minor issues.
1.3.550	Added a <code>-NoShutdown</code> option to enable Sysprep with no shutdown.
1.3.540	Fixed minor issues.

Version	Details
1.3.530	Fixed minor issues.
1.3.521	Fixed minor issues.
1.3.0	<ul style="list-style-type: none"> • Fixed a hexadecimal length issue for computer name change. • Fixed a possible reboot loop for computer name change. • Fixed an issue in wallpaper setup.
1.2.0	<ul style="list-style-type: none"> • Update to display information about installed operating system (OS) in EC2 system log. • Update to display EC2Launch and SSM Agent version in EC2 system log. • Fixed minor issues.
1.1.2	<ul style="list-style-type: none"> • Update to display ENA driver information in EC2 system log. • Update to exclude Hyper-V from primary NIC filter logic. • Added KMS server and port into registry key for KMS activation. • Improved wallpaper setup for multiple users. • Update to clear routes from persistent store. • Update to remove the z from availability zone in DNS suffix list. • Update to address an issue with the <runAsLocalSystem> tag in user data.
1.1.1	Initial release.

Configuring a Windows Instance Using the EC2Config Service

Windows AMIs for Windows Server 2012 R2 and earlier include an optional service, the EC2Config service (`EC2Config.exe`). EC2Config starts when the instance boots and performs tasks during startup and each time you stop or start the instance. EC2Config can also perform tasks on demand. Some of these tasks are automatically enabled, while others must be enabled manually. Although optional, this service provides access to advanced features that aren't otherwise available. This service runs in the LocalSystem account.

Note

EC2Launch replaces EC2Config on Windows AMIs for Windows Server 2016. For more information, see [Configuring a Windows Instance Using EC2Launch \(p. 374\)](#).

EC2Config uses settings files to control its operation. You can update these settings files using either a graphical tool or by directly editing XML files. The service binaries and additional files are contained in the `%ProgramFiles%\Amazon\EC2ConfigService` directory.

Contents

- [EC2Config Tasks \(p. 384\)](#)

- [Installing the Latest Version of EC2Config \(p. 384\)](#)
- [Stopping, Restarting, Deleting, or Uninstalling EC2Config \(p. 385\)](#)
- [EC2Config and AWS Systems Manager \(p. 386\)](#)
- [EC2Config and Sysprep \(p. 386\)](#)
- [EC2 Service Properties \(p. 387\)](#)
- [EC2Config Settings Files \(p. 390\)](#)
- [Configure Proxy Settings for the EC2Config Service \(p. 393\)](#)
- [EC2Config Version History \(p. 395\)](#)
- [Troubleshooting Issues with the EC2Config Service \(p. 408\)](#)

EC2Config Tasks

EC2Config runs initial startup tasks when the instance is first started and then disables them. To run these tasks again, you must explicitly enable them prior to shutting down the instance, or by running Sysprep manually. These tasks are as follows:

- Set a random, encrypted password for the administrator account.
- Generate and install the host certificate used for Remote Desktop Connection.
- Dynamically extend the operating system partition to include any unpartitioned space.
- Execute the specified user data (and Cloud-Init, if it's installed). For more information about specifying user data, see [Working with Instance User Data \(p. 455\)](#).

EC2Config performs the following tasks every time the instance starts:

- Change the host name to match the private IP address in Hex notation (this task is disabled by default and must be enabled in order to run at instance start).
- Configure the key management server (AWS KMS), check for Windows activation status, and activate Windows as necessary.
- Mount all Amazon EBS volumes and instance store volumes, and map volume names to drive letters.
- Write event log entries to the console to help with troubleshooting (this task is disabled by default and must be enabled in order to run at instance start).
- Write to the console that Windows is ready.
- Add a custom route to the primary network adapter to enable the following IP addresses when multiple NICs are attached: 169.254.169.250, 169.254.169.251, and 169.254.169.254. These addresses are used by Windows Activation and when you access instance metadata.

EC2Config performs the following task every time a user logs in:

- Display wallpaper information to the desktop background.

While the instance is running, you can request that EC2Config perform the following task on demand:

- Run Sysprep and shut down the instance so that you can create an AMI from it. For more information, see [Create a Standard Amazon Machine Image Using Sysprep \(p. 104\)](#).

Installing the Latest Version of EC2Config

By default, the EC2Config service is included in AMIs prior to Windows Server 2016. When the EC2Config service is updated, new Windows AMIs from AWS include the latest version of the service. However, you need to update your own Windows AMIs and instances with the latest version of EC2Config.

Note

EC2Launch replaces EC2Config on Windows Server 2016 AMIs. For more information, see [Configuring a Windows Instance Using EC2Launch \(p. 374\)](#).

For information about how to receive notifications for EC2Config updates, see [Subscribing to EC2Config Service Notifications \(p. 407\)](#). For information about the changes in each version, see the [EC2Config Version History \(p. 395\)](#).

Before You Begin

- Verify that you have .NET framework 3.5 SP1 or greater.
- By default, Setup replaces your settings files with default settings files during installation and restarts the EC2Config service when the installation is completed. If you changed EC2Config service settings, copy the config.xml file from the %Program Files%\Amazon\Ec2ConfigService\Settings directory. After you update the EC2Config service, you can restore this file to retain your configuration changes.
- If your version of EC2Config is earlier than version 2.1.19 and you are installing version 2.2.12 or earlier, you must first install version 2.1.19. To install version 2.1.19, download [EC2Install_2.1.19.zip](#), unzip the file, and then run EC2Install.exe.

Note

If your version of EC2Config is earlier than version 2.1.19 and you are installing version 2.3.313 or later, you can install it directly without installing version 2.1.19 first.

Verify the EC2Config Version

Use the following procedure to verify the version of EC2Config that is installed on your instances.

To verify the installed version of EC2Config

1. Launch an instance from your AMI and connect to it.
2. In Control Panel, select **Programs and Features**.
3. In the list of installed programs, look for Ec2ConfigService. Its version number appears in the **Version** column.

Update EC2Config

Use the following procedure to download and install the latest version of EC2Config on your instances.

To download and install the latest version of EC2Config

1. Download and unzip the [EC2Config installer](#).
2. Run EC2Install.exe. For a complete list of options, run EC2Install with the /? option. By default, setup displays prompts. To run the command with no prompts, use the /quiet option.

Important

To keep the custom settings from the config.xml file that you saved, run EC2Install with the /norestart option, restore your settings, and then restart the EC2Config service manually.

3. If you are running EC2Config version 4.0 or later, you must restart the SSM Agent on the instance from the Microsoft Services snap-in.

Stopping, Restarting, Deleting, or Uninstalling EC2Config

You can manage the EC2Config service just as you would any other service.

To apply updated settings to your instance, you can stop and restart the service. If you're manually installing EC2Config, you must stop the service first.

To stop the EC2Config service

1. Launch and connect to your Windows instance.
2. On the **Start** menu, point to **Administrative Tools**, and then click **Services**.
3. In the list of services, right-click **EC2Config**, and select **Stop**.

To restart the EC2Config service

1. Launch and connect to your Windows instance.
2. On the **Start** menu, point to **Administrative Tools**, and then click **Services**.
3. In the list of services, right-click **EC2Config**, and select **Restart**.

If you don't need to update the configuration settings, create your own AMI, or use Amazon EC2 Systems Manager (SSM), you can delete and uninstall the service. Deleting a service removes its registry subkey. Uninstalling a service removes the files, the registry subkey, and any shortcuts to the service.

To delete the EC2Config service

1. Start a command prompt window.
2. Run the following command:

```
sc delete ec2config
```

To uninstall EC2Config

1. Launch and connect to your Windows instance.
2. On the **Start** menu, click **Control Panel**.
3. Double-click **Programs and Features**.
4. On the list of programs, select **EC2ConfigService**, and click **Uninstall**.

EC2Config and AWS Systems Manager

The EC2Config service processes Systems Manager requests on instances created from AMIs for versions of Windows Server prior to Windows Server 2016 that were published before November 2016.

Instances created from AMIs for versions of Windows Server prior to Windows Server 2016 that were published after November 2016 include the EC2Config service and SSM Agent. EC2Config performs all of the tasks described earlier, and SSM Agent processes requests for Systems Manager capabilities like Run Command and State Manager.

You can use Run Command to upgrade your existing instances to use to the latest version of the EC2Config service and SSM Agent. For more information, see [Example: Update the SSM Agent](#) in the *AWS Systems Manager User Guide*.

EC2Config and Sysprep

The EC2Config service runs Sysprep, a Microsoft tool that enables you to create a customized Windows AMI that can be reused. When EC2Config calls Sysprep, it uses the files in %ProgramFiles%\Amazon\EC2ConfigService\Settings to determine which operations to perform. You can edit these files

indirectly using the **Ec2 Service Properties** dialog box, or directly using an XML editor or a text editor. However, there are some advanced settings that aren't available in the **Ec2 Service Properties** dialog box, so you must edit those entries directly.

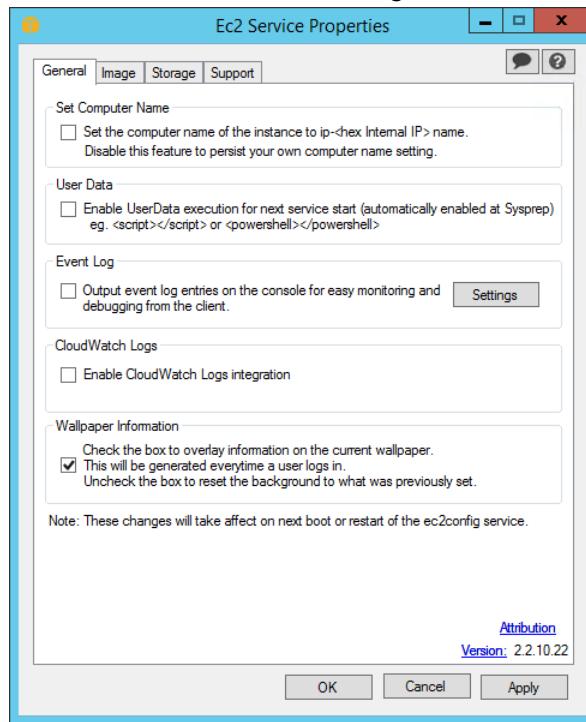
If you create an AMI from an instance after updating its settings, the new settings are applied to any instance that's launched from the new AMI. For information about creating an AMI, see [Creating a Custom Windows AMI \(p. 65\)](#).

EC2 Service Properties

The following procedure describes how to use the **Ec2 Service Properties** dialog box to enable or disable settings.

To change settings using the Ec2 Service Properties dialog box

1. Launch and connect to your Windows instance.
2. From the **Start** menu, click **All Programs**, and then click **EC2ConfigService Settings**.



3. On the **General** tab of the **Ec2 Service Properties** dialog box, you can enable or disable the following settings.

Set Computer Name

If this setting is enabled (it is disabled by default), the host name is compared to the current internal IP address at each boot; if the host name and internal IP address do not match, the host name is reset to contain the internal IP address and then the system reboots to pick up the new host name. To set your own host name, or to prevent your existing host name from being modified, do not enable this setting.

User Data

User data execution enables you to specify scripts in the instance metadata. By default, these scripts are run during the initial launch. You can also configure them to run the next time you reboot or start the instance, or every time you reboot or start the instance.

If you have a large script, we recommend that you use user data to download the script, and then execute it.

For more information, see [User Data Execution \(p. 449\)](#).

Event Log

Use this setting to display event log entries on the console during boot for easy monitoring and debugging.

Click **Settings** to specify filters for the log entries sent to the console. The default filter sends the three most recent error entries from the system event log to the console.

CloudWatch Logs

Starting with EC2Config version 2.2.5 (version 2.2.6 or later is recommended), you can export all Windows Server messages in the System log, Security log, Application log, and IIS log to CloudWatch Logs and monitor them using CloudWatch metrics. EC2Config version 2.2.10 or later adds the ability to export any event log data, Event Tracing (Windows) data, or text-based log files to CloudWatch Logs. In addition, you can also export performance counter data to CloudWatch. For more information, see [Monitoring System, Application, and Custom Log Files in the Amazon CloudWatch User Guide](#).

1. Select **Enable CloudWatch integration**, and then click **OK**.
2. Edit the `\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json` file and configure the types of logs you want to send to CloudWatch Logs. For more information, see [Sending Logs, Events, and Performance Counters to Amazon CloudWatch \(p. 526\)](#).

If your instance is running EC2Config version 4.x or later, this option is not available. SSM Agent sends log data to CloudWatch instead.

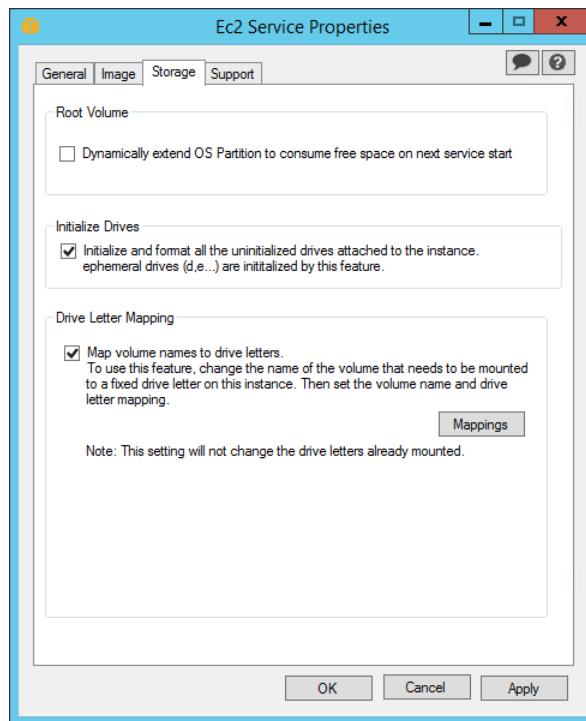
Wallpaper Information

Use this setting to display system information on the desktop background. The following is an example of the information displayed on the desktop background.

Hostname :	WIN-UORFOJCTPUU
Instance ID :	i-d583f76a
Public IP Address :	54.208.43.227
Private IP Address :	172.31.42.195
Availability Zone :	us-east-1b
Instance Size :	t2.micro
Architecture :	AMD64

The information displayed on the desktop background is controlled by the settings file `EC2ConfigService\Settings\WallpaperSettings.xml`.

4. Click the **Storage** tab. You can enable or disable the following settings.



Root Volume

This setting dynamically extends Disk 0/Volume 0 to include any unpartitioned space. This can be useful when the instance is booted from a root device volume that has a custom size.

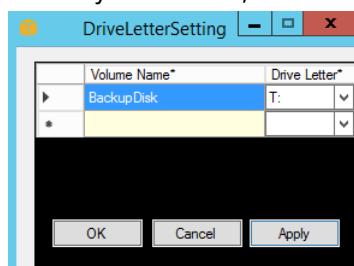
Initialize Drives

This setting formats and mounts all volumes attached to the instance during start.

Drive Letter Mapping

The system maps the volumes attached to an instance to drive letters. For Amazon EBS volumes, the default is to assign drive letters going from D: to Z:. For instance store volumes, the default depends on the driver. Citrix PV drivers assign instance store volumes drive letters going from Z: to A:. Red Hat drivers assign instance store volumes drive letters going from D: to Z:.

To choose the drive letters for your volumes, click **Mappings**. In the **DriveLetterSetting** dialog box, specify the **Volume Name** and **Drive Letter** values for each volume, click **Apply**, and then click **OK**. We recommend that you select drive letters that avoid conflicts with drive letters that are likely to be in use, such as drive letters in the middle of the alphabet.



After you specify a drive letter mapping and attach a volume with same label as one of the volume names that you specified, EC2Config automatically assigns your specified drive letter

to that volume. However, the drive letter mapping fails if the drive letter is already in use. Note that EC2Config doesn't change the drive letters of volumes that were already mounted when you specified the drive letter mapping.

5. To save your settings and continue working on them later, click **OK** to close the **Ec2 Service Properties** dialog box. If you have finished customizing your instance and want to create an AMI from that instance, see [Create a Standard Amazon Machine Image Using Sysprep \(p. 104\)](#).

EC2Config Settings Files

The settings files control the operation of the EC2Config service. These files are located in the C:\\Program Files\\Amazon\\Ec2ConfigService\\Settings directory:

- ActivationSettings.xml—Controls product activation using a key management server (KMS).
- AWS.EC2.Windows.CloudWatch.json—Controls which performance counters to send to CloudWatch and which logs to send to CloudWatch Logs. For more information about how to change the settings in this file, see [Sending Logs, Events, and Performance Counters to Amazon CloudWatch \(p. 526\)](#).
- BundleConfig.xml—Controls how EC2Config prepares an instance store-backed instance for AMI creation.
- Config.xml—Controls the primary settings.
- DriveLetterConfig.xml—Controls drive letter mappings.
- EventLogConfig.xml—Controls the event log information that's displayed on the console while the instance is booting.
- WallpaperSettings.xml—Controls the information that's displayed on the desktop background.

ActivationSettings.xml

This file contains settings that control product activation. When Windows boots, the EC2Config service checks whether Windows is already activated. If Windows is not already activated, it attempts to activate Windows by searching for the specified KMS server.

- SetAutodiscover—Indicates whether to detect a KMS automatically.
- TargetKMSServer—Stores the private IP address of a KMS. The KMS must be in the same region as your instance.
- DiscoverFromZone—Discovers the KMS server from the specified DNS zone.
- ReadFromUserData—Gets the KMS server from UserData.
- LegacySearchZones—Discovers the KMS server from the specified DNS zone.
- DoActivate—Attempts activation using the specified settings in the section. This value can be true or false.
- LogResultToConsole—Displays the result to the console.

BundleConfig.xml

This file contains settings that control how EC2Config prepares an instance for AMI creation.

- AutoSysprep—Indicates whether to use Sysprep automatically. Change the value to Yes to use Sysprep.
- SetRDPCertificate—Sets a self-signed certificate to the Remote Desktop server. This enables you to securely RDP into the instances. Change the value to Yes if the new instances should have the certificate.

This setting is not used with Windows Server 2008 or Windows Server 2012 instances because they can generate their own certificates.

- **SetPasswordAfterSysprep**—Sets a random password on a newly launched instance, encrypts it with the user launch key, and outputs the encrypted password to the console. Change the value of this setting to **No** if the new instances should not be set to a random encrypted password.

Config.xml

Plug-ins

- **Ec2SetPassword**—Generates a random encrypted password each time you launch an instance. This feature is disabled by default after the first launch so that reboots of this instance don't change a password set by the user. Change this setting to **Enabled** to continue to generate passwords each time you launch an instance.

This setting is important if you are planning to create an AMI from your instance.

- **Ec2SetComputerName**—Sets the host name of the instance to a unique name based on the IP address of the instance and reboots the instance. To set your own host name, or prevent your existing host name from being modified, you must disable this setting.
- **Ec2InitializeDrives**—Initializes and formats all volumes during startup. This feature is enabled by default.
- **Ec2EventLog**—Displays event log entries in the console. By default, the three most recent error entries from the system event log are displayed. To specify the event log entries to display, edit the `EventLogConfig.xml` file located in the `EC2ConfigService\Settings` directory. For information about the settings in this file, see [Eventlog Key](#) in the MSDN Library.
- **Ec2ConfigureRDP**—Sets up a self-signed certificate on the instance, so users can securely access the instance using Remote Desktop. This feature is disabled on Windows Server 2008 and Windows Server 2012 instances because they can generate their own certificates.
- **Ec2OutputRDPCert**—Displays the Remote Desktop certificate information to the console so that the user can verify it against the thumbprint.
- **Ec2SetDriveLetter**—Sets the drive letters of the mounted volumes based on user-defined settings. By default, when an Amazon EBS volume is attached to an instance, it can be mounted using the drive letter on the instance. To specify your drive letter mappings, edit the `DriveLetterConfig.xml` file located in the `EC2ConfigService\Settings` directory.
- **Ec2WindowsActivate**—The plug-in handles Windows activation. It checks to see if Windows is activated. If not, it updates the KMS client settings, and then activates Windows.

To modify the KMS settings, edit the `ActivationSettings.xml` file located in the `EC2ConfigService\Settings` directory.

- **Ec2DynamicBootVolumeSize**—Extends Disk 0/Volume 0 to include any unpartitioned space.
- **Ec2HandleUserData**—Creates and executes scripts created by the user on the first launch of an instance after Sysprep is run. Commands wrapped in script tags are saved to a batch file, and commands wrapped in PowerShell tags are saved to a .ps1 file.
- **Ec2ElasticGpuSetup**—Installs the Elastic GPU software package if the instance is associated with an elastic GPU.
- **Ec2FeatureLogging**—Sends Windows feature installation and corresponding service status to the console. Supported only for the Microsoft Hyper-V feature and corresponding vmms service.

Global Settings

- **ManageShutdown**—Ensures that instances launched from instance store-backed AMIs do not terminate while running Sysprep.

- **SetDnsSuffixList**—Sets the DNS suffix of the network adapter for Amazon EC2. This allows DNS resolution of servers running in Amazon EC2 without providing the fully qualified domain name.
- **WaitForMetaDataAvailable**—Ensures that the EC2Config service will wait for metadata to be accessible and the network available before continuing with the boot. This check ensures that EC2Config can obtain information from metadata for activation and other plug-ins.
- **ShouldAddRoutes**—Adds a custom route to the primary network adapter to enable the following IP addresses when multiple NICs are attached: 169.254.169.250, 169.254.169.251, and 169.254.169.254. These addresses are used by Windows Activation and when you access instance metadata.
- **RemoveCredentialsfromSysprepOnStartup**—Removes the administrator password from **Sysprep.xml** the next time the service starts. To ensure that this password persists, edit this setting.

DriveLetterConfig.xml

This file contains settings that control drive letter mappings. By default, a volume can be mapped to any available drive letter. You can mount a volume to a particular drive letter as follows.

```
<?xml version="1.0" standalone="yes"?>
<DriveLetterMapping>
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
  . .
  <Mapping>
    <VolumeName></VolumeName>
    <DriveLetter></DriveLetter>
  </Mapping>
</DriveLetterMapping>
```

- **VolumeName**—The volume label. For example, **My Volume**. To specify a mapping for an instance storage volume, use the label **Temporary Storage X**, where X is a number from 0 to 25.
- **DriveLetter**—The drive letter. For example, **M:**. The mapping fails if the drive letter is already in use.

EventLogConfig.xml

This file contains settings that control the event log information that's displayed on the console while the instance is booting. By default, we display the three most recent error entries from the System event log.

- **Category**—The event log key to monitor.
- **ErrorType**—The event type (for example, **Error**, **Warning**, **Information**.)
- **NumEntries**—The number of events stored for this category.
- **LastMessageTime**—To prevent the same message from being pushed repeatedly, the service updates this value every time it pushes a message.
- **AppName**—The event source or application that logged the event.

WallpaperSettings.xml

This file contains settings that control the information that's displayed on the desktop background. The following information is displayed by default.

- **Hostname**—Displays the computer name.
- **Instance ID**—Displays the ID of the instance.
- **Public IP Address**—Displays the public IP address of the instance.

- **Private IP Address**—Displays the private IP address of the instance.
- **Availability Zone**—Displays the Availability Zone in which the instance is running.
- **Instance Size**—Displays the type of instance.
- **Architecture**—Displays the setting of the `PROCESSOR_ARCHITECTURE` environment variable.

You can remove any of the information that's displayed by default by deleting its entry. You can add additional instance metadata to display as follows.

```
<WallpaperInformation>
  <name>display_name</name>
  <source>metadata</source>
  <identifier>meta-data/path</identifier>
</WallpaperInformation>
```

You can add additional System environment variables to display as follows.

```
<WallpaperInformation>
  <name>display_name</name>
  <source>EnvironmentVariable</source>
  <identifier>variable-name</identifier>
</WallpaperInformation>
```

InitializeDrivesSettings.xml

This file contains settings that control how EC2Config initializes drives.

By default, EC2Config initialize drives that were not brought online with the operating system. You can customize the plugin as follows.

```
<InitializeDrivesSettings>
  <SettingsGroup>setting</SettingsGroup>
</InitializeDrivesSettings>
```

Use a settings group to specify how you want to initialize drives:

FormatWithTRIM

Enables the TRIM command when formatting drives. After a drive has been formatted and initialized, the system restores TRIM configuration.

Starting with EC2Config version 3.18, the TRIM command is disabled during the disk format operation by default. This improves formatting times. Use this setting to enable TRIM during the disk format operation for EC2Config version 3.18 and later.

FormatWithoutTRIM

Disables the TRIM command when formatting drives and improves formatting times in Windows. After a drive has been formatted and initialized, the system restores TRIM configuration.

DisableInitializeDrives

Disables formatting for new drives. Use this setting to initialize drives manually.

Configure Proxy Settings for the EC2Config Service

You can configure the EC2Config service to communicate through a proxy using one of the following methods: the AWS SDK for .NET, the `system.net` element, or Microsoft Group Policy and Internet Explorer. Using the AWS SDK for .NET is the preferred method because you can specify a user name and password.

Methods

- [Configure Proxy Settings Using the AWS SDK for .NET \(Preferred\) \(p. 394\)](#)
- [Configure Proxy Settings Using the system.net Element \(p. 394\)](#)
- [Configure Proxy Settings Using Microsoft Group Policy and Microsoft Internet Explorer \(p. 395\)](#)

Configure Proxy Settings Using the AWS SDK for .NET (Preferred)

You can configure proxy settings for the EC2Config service by specifying the `proxy` element in the `Ec2Config.exe.config` file. For more information, see [Configuration Files Reference for AWS SDK for .NET](#).

To specify the proxy element in `Ec2Config.exe.config`

1. Edit the `Ec2Config.exe.config` file on an instance where you want the EC2Config service to communicate through a proxy. By default, the file is located in the following directory:
`%ProgramFiles%\Amazon\Ec2ConfigService`.
2. Add the following `aws` element to the `configSections`. Do not add this to any existing `sectionGroups`.

For EC2Config versions 3.17 or earlier

```
<configSections>
    <section name="aws" type="Amazon.AWSSection, AWSSDK"/>
</configSections>
```

For EC2Config versions 3.18 or later

```
<configSections>
    <section name="aws" type="Amazon.AWSSection, AWSSDK.Core"/>
</configSections>
```

3. Add the following `aws` element to the `Ec2Config.exe.config` file.

```
<aws>
    <proxy
        host="string value"
        port="string value"
        username="string value"
        password="string value" />
</aws>
```

4. Save your changes.

Configure Proxy Settings Using the `system.net` Element

You can specify proxy settings in a `system.net` element in the `Ec2Config.exe.config` file. For more information, see [defaultProxy Element \(Network Settings\)](#) on MSDN.

To specify the `system.net` element in `Ec2Config.exe.config`

1. Edit the `Ec2Config.exe.config` file on an instance where you want the EC2Config service to communicate through a proxy. By default, the file is located in the following directory:
`%ProgramFiles%\Amazon\Ec2ConfigService`.
2. Add a `defaultProxy` entry to `system.net`. For more information, see [defaultProxy Element \(Network Settings\)](#) on MSDN.

For example, the following configuration routes all traffic to use the proxy that is currently configured for Internet Explorer, with the exception of the metadata and licensing traffic, which will bypass the proxy.

```
<defaultProxy>
    <proxy usesystemdefault="true" />
    <bypasslist>
        <add address="169.254.169.250" />
        <add address="169.254.169.251" />
        <add address="169.254.169.254" />
    </bypasslist>
</defaultProxy>
```

3. Save your changes.

Configure Proxy Settings Using Microsoft Group Policy and Microsoft Internet Explorer

The EC2Config service runs under the Local System user account. You can specify instance-wide proxy settings for this account in Internet Explorer after you change Group Policy settings on the instance.

To configure proxy settings using Group Policy and Internet Explorer

1. On an instance where you want the EC2Config service to communicate through a proxy, open a Command prompt as an Administrator, type `gpedit.msc`, and press Enter.
2. In the Local Group Policy Editor, under **Local Computer Policy**, choose **Computer Configuration**, **Administrative Templates**, **Windows Components**, **Internet Explorer**.
3. In the right-pane, choose **Make proxy settings per-machine (rather than per-user)** and then choose **Edit policy setting**.
4. Choose **Enabled**, and then choose **Apply**.
5. Open Internet Explorer, and then choose the **Tools** button.
6. Choose **Internet Option**, and then choose the **Connections** tab.
7. Choose **LAN settings**.
8. Under **Proxy server**, choose the **Use a proxy server for your LAN** option.
9. Specify address and port information and then choose **OK**.

EC2Config Version History

Windows AMIs prior to Windows Server 2016 include an optional service called the EC2Config service (`EC2Config.exe`). EC2Config starts when the instance boots and performs tasks during startup and each time you stop or start the instance. For information about the EC2Config versions included in the Windows AMIs, see [Managed AWS Windows AMIs \(p. 77\)](#).

You can receive notifications when new versions of the EC2Config service are released. For more information, see [Subscribing to EC2Config Service Notifications \(p. 407\)](#).

The following table describes the released versions of EC2Config. For information about the updates for SSM Agent, see [Amazon SSM Agent Release Notes](#).

Version	Details
4.9.3067	<ul style="list-style-type: none">Improvements made to instance hibernationNew version of SSM Agent 2.3.235.0

Version	Details
4.9.3034	<ul style="list-style-type: none"> Added route 169.254.169.253/32 for DNS server New version of SSM Agent 2.3.193.0
4.9.2986	<ul style="list-style-type: none"> Added signing for all EC2Config related binaries New version of SSM Agent 2.3.136.0
4.9.2953	New version of SSM Agent (2.3.117.0)
4.9.2926	New version of SSM Agent (2.3.68.0)
4.9.2905	<ul style="list-style-type: none"> New version of SSM Agent (2.3.50.0) Added route 169.254.169.123/32 to AMZN time service Added route 169.254.169.249/32 to GRID license service Fixed an issue causing EBS NVMe volumes to be marked as ephemeral
4.9.2854	New version of SSM Agent (2.3.13.0)
4.9.2831	New version of SSM Agent (2.2.916.0)
4.9.2818	New version of SSM Agent (2.2.902.0)
4.9.2756	New version of SSM Agent (2.2.800.0)
4.9.2688	New version of SSM Agent (2.2.607.0)
4.9.2660	New version of SSM Agent (2.2.546.0)
4.9.2644	New version of SSM Agent (2.2.493.0)
4.9.2586	New version of SSM Agent (2.2.392.0)
4.9.2565	<ul style="list-style-type: none"> New version of SSM Agent (2.2.355.0) Fixed an issue on M5 and C5 instances (unable to find PV drivers) Add console logging for instance type, newest PV drivers, and NVMe drivers
4.9.2549	New version of SSM Agent (2.2.325.0)
4.9.2461	New version of SSM Agent (2.2.257.0)
4.9.2439	New version of SSM Agent (2.2.191.0)
4.9.2400	New version of SSM Agent (2.2.160.0)
4.9.2327	<ul style="list-style-type: none"> New version of SSM Agent (2.2.120.0) Added COM port discovery on Amazon EC2 bare metal instances Added Hyper-V status logging on Amazon EC2 bare metal instances
4.9.2294	New version of SSM Agent (2.2.103.0)

Version	Details
4.9.2262	New version of SSM Agent (2.2.93.0)
4.9.2246	New version of SSM Agent (2.2.82.0)
4.9.2218	New version of SSM Agent (2.2.64.0)
4.9.2212	New version of SSM Agent (2.2.58.0)
4.9.2203	New version of SSM Agent (2.2.45.0)
4.9.2188	New version of SSM Agent (2.2.30.0)
4.9.2180	<ul style="list-style-type: none"> • New version of SSM Agent (2.2.24.0) • Added the Elastic GPU plugin for GPU instances
4.9.2143	New version of SSM Agent (2.2.16.0)
4.9.2140	New version of SSM Agent (2.1.10.0)
4.9.2130	New version of SSM Agent (2.1.4.0)
4.9.2106	New version of SSM Agent (2.0.952.0)
4.9.2061	New version of SSM Agent (2.0.922.0)
4.9.2047	New version of SSM Agent (2.0.913.0)
4.9.2031	New version of SSM Agent (2.0.902.0)
4.9.2016	<ul style="list-style-type: none"> • New version of SSM Agent (2.0.879.0) • Fixed the CloudWatch Logs directory path for Windows Server 2003
4.9.1981	<ul style="list-style-type: none"> • New version of SSM Agent (2.0.847.0) • Fixed the issue with <code>important.txt</code> being generated in EBS volumes.
4.9.1964	New version of SSM Agent (2.0.842.0)
4.9.1951	<ul style="list-style-type: none"> • New version of SSM Agent (2.0.834.0) • Fixed the issue with drive letter not being mapped from Z: for ephemeral drives.
4.9.1925	<ul style="list-style-type: none"> • New version of SSM Agent (2.0.822.0) • [Bug] This version is not a valid update target from SSM Agent v4.9.1775.
4.9.1900	New version of SSM Agent (2.0.805.0)
4.9.1876	<ul style="list-style-type: none"> • New version of SSM Agent (2.0.796.0) • Fixed an issue with output/error redirection for admin userdata execution.

Version	Details
4.9.1863	<ul style="list-style-type: none"> • New version of SSM Agent (2.0.790.0) • Fixed problems with attaching multiple EBS volumes to an Amazon EC2 instance. • Improved CloudWatch to take a configuration path, keeping the backwards compatibility.
4.9.1791	New version of SSM Agent (2.0.767.0)
4.9.1775	New version of SSM Agent (2.0.761.0)
4.9.1752	New version of SSM Agent (2.0.755.0)
4.9.1711	New version of SSM Agent (2.0.730.0)
4.8.1676	New version of SSM Agent (2.0.716.0)
4.7.1631	New version of SSM Agent (2.0.682.0)
4.6.1579	<ul style="list-style-type: none"> • New version of SSM Agent (2.0.672.0) • Fixed agent update issue with v4.3, v4.4, and v4.5
4.5.1534	New version of SSM Agent (2.0.645.1)
4.4.1503	New version of SSM Agent (2.0.633.0)
4.3.1472	New version of SSM Agent (2.0.617.1)
4.2.1442	New version of SSM Agent (2.0.599.0)
4.1.1378	New version of SSM Agent (2.0.558.0)
4.0.1343	<ul style="list-style-type: none"> • Run Command, SSM Config, the CloudWatch agent, and domain join support have been moved into another agent called SSM Agent. SSM Agent will be installed as part of the EC2Config upgrade. For more information, see EC2Config and AWS Systems Manager (p. 386). • If you have a proxy set up in EC2Config, you will need to update your proxy settings for SSM Agent before upgrading. If you do not update the proxy settings, you will not be able to use Run Command to manage your instances. To avoid this, see the following information before updating to the newer version: Installing SSM Agent on Windows. • If you previously enabled CloudWatch integration on your instances by using a local configuration file (AWS.EC2.Windows.CloudWatch.json), you will need to configure the file to work with SSM Agent. For more information, see Use SSM Agent to Configure CloudWatch (p. 538).

Version	Details
3.19.1153	<ul style="list-style-type: none"> • Re-enabled activation plugin for instances with old KMS configuration. • Change default TRIM behavior to be disabled during disk format operation and added FormatWithTRIM for overriding InitializeDisks plugin with userdata.
3.18.1118	<ul style="list-style-type: none"> • Fix to reliably add routes to the primary network adapter. • Updates to improve support for AWS services.
3.17.1032	<ul style="list-style-type: none"> • Fixes duplicate system logs appearing when filters set to same category. • Fixes to prevent from hanging during disk initialization.
3.16.930	Added support to log "Window is Ready to use" event to Windows Event Log on start.
3.15.880	Fix to allow uploading run command output to S3 bucket names with '.' character.
3.14.786	<p>Added support to override InitializeDisks plugin settings. For example: To speed up SSD disk initialize, you can temporarily disable TRIM by specifying this in userdata:</p> <pre><InitializeDrivesSettings><SettingsGroup>FormatWithoutTRIM</SettingsGroup></InitializeDrivesSettings></pre>
3.13.727	SSM RunCommand - Fixes to process commands reliably after windows reboot.
3.12.649	<ul style="list-style-type: none"> • Fix to gracefully handle reboot when running commands/scripts. • Fix to reliably cancel running commands. • Add support for (optionally) uploading MSI logs to S3 when installing applications via Run Command.
3.11.521	<ul style="list-style-type: none"> • Fixes to enable RDP thumbprint generation for Windows Server 2003. • Fixes to include timezone and UTC offset in the EC2Config log lines. • SSM support to run commands in parallel. • Roll back previous change to bring partitioned disks online.
3.10.442	<ul style="list-style-type: none"> • Fix SSM (Simple Systems Manager) configuration failures when installing MSI applications. • Fix to reliably bring storage disks online. • Updates to improve support for AWS services.

Version	Details
3.9.359	<ul style="list-style-type: none"> Fix in post Sysprep script to leave the configuration of windows update in a default state. Fix the password generation plugin to improve the reliability in getting GPO password policy settings. Restrict EC2Config/SSM log folder permissions to the local Administrators group. Updates to improve support for AWS services.
3.8.294	<ul style="list-style-type: none"> Fixed an issue with CloudWatch that prevented logs from getting uploaded when not on primary drive. Improved the disk initialization process by adding retry logic. Added improved error handling when the SetPassword plugin occasionally failed during AMI creation. Updates to improve support for AWS services.
3.7.308	<ul style="list-style-type: none"> Improvements to the ec2config-cli utility for config testing and troubleshooting within instance. Avoid adding static routes for KMS and metadata service on an OpenVPN adapter. Fixed an issue where user-data execution was not honoring the "persist" tag. Improved error handling when logging to the EC2 console is not available. Updates to improve support for AWS services.
3.6.269	<ul style="list-style-type: none"> Windows activation reliability fix to first use link local address 169.254.0.250/251 for activating windows via KMS Improved proxy handling for SSM, Windows Activation and Domain Join scenarios Fixed an issue where duplicate lines of user accounts were added to the Sysprep answer file
3.5.228	<ul style="list-style-type: none"> Addressed a scenario where the CloudWatch plugin may consume excessive CPU and memory reading Windows Event Logs Added a link to the CloudWatch configuration documentation in the EC2Config Settings UI
3.4.212	<ul style="list-style-type: none"> Fixes to EC2Config when used in combination with VM-Import. Fixed service naming issue in the WiX installer.

Version	Details
3.3.174	<ul style="list-style-type: none"> • Improved exception handling for ssm and domain join failures. • Change to support SSM schema versioning. • Fixed formatting ephemeral disks on Win2K3. • Change to support configuring disk size greater than 2TB. • Reduced virtual memory usage by setting GC mode to default. • Support for downloading artifacts from UNC path in aws:psModule and aws:application plugin. • Improved logging for Windows activation plugin.
3.2.97	<ul style="list-style-type: none"> • Performance improvements by delay loading SSM assemblies. • Improved exception handling for malformed sysprep2008.xml. • Command line support for SSM "Apply" configuration. • Change to support domain join when there is a pending computer rename. • Support for optional parameters in the aws:applications plugin. • Support for command array in aws:psModule plugin.
3.0.54	<ul style="list-style-type: none"> • Enable support for Amazon EC2 Systems Manager (SSM). • Automatically domain join EC2 Windows instances to an AWS directory via SSM. • Configure and upload CloudWatch logs/metrics via SSM. • Install PowerShell modules via SSM. • Install MSI applications via SSM.
2.4.233	<ul style="list-style-type: none"> • Added scheduled task to recover EC2Config from service startup failures. • Improvements to the Console log error messages. • Updates to improve support for AWS services.

Version	Details
2.3.313	<ul style="list-style-type: none"> Fixed an issue with large memory consumption in some cases when the CloudWatch Logs feature is enabled. Fixed an upgrade bug so that ec2config versions lower than 2.1.19 can now upgrade to latest. Updated COM port opening exception to be more friendly and useful in logs. Ec2configServiceSettings UI disabled resizing and fixed the attribution and version display placement in UI.
2.2.12	<ul style="list-style-type: none"> Handled NullPointerException while querying a registry key for determining Windows Sysprep state which returned null occasionally. Freed up unmanaged resources in finally block.
2.2.11	Fixed a issue in CloudWatch plugin for handling empty log lines.
2.2.10	<ul style="list-style-type: none"> Removed configuring CloudWatch Logs settings through UI. Enable users to define CloudWatch Logs settings in %ProgramFiles% \Amazon\Ec2ConfigService\Settings \AWS.EC2.Windows.CloudWatch.json file to allow future enhancements.
2.2.9	Fixed unhandled exception and added logging.
2.2.8	<ul style="list-style-type: none"> Fixes Windows OS version check in EC2Config Installer to support Windows Server 2003 SP1 and later. Fixes null value handling when reading registry keys related to updating Sysprep config files.
2.2.7	<ul style="list-style-type: none"> Added support for EC2Config to run during Sysprep execution for Windows 2008 and greater. Improved exception handling and logging for better diagnostics
2.2.6	<ul style="list-style-type: none"> Reduced the load on the instance and on CloudWatch Logs when uploading log events. Addressed an upgrade issue where the CloudWatch Logs plug-in did not always stay enabled

Version	Details
2.2.5	<ul style="list-style-type: none"> Added support to upload logs to CloudWatch Log Service. Fixed a race condition issue in Ec2OutputRDPCert plug-in Changed EC2Config Service recovery option to Restart from TakeNoAction Added more exception information when EC2Config Crashes
2.2.4	<ul style="list-style-type: none"> Fixed a typo in PostSysprep.cmd Fixed the bug which EC2Config does not pin itself onto start menu for OS2012+
2.2.3	<ul style="list-style-type: none"> Added option to install EC2Config without service starting immediately upon install. To use, run 'Ec2Install.exe start=false' from the command prompt Added parameter in wallpaper plugin to control adding/removing wallpaper. To use, run 'Ec2WallpaperInfo.exe set' or 'Ec2WallpaperInfo.exe revert' from the command prompt Added checking for RealTimelsUniversal key, output incorrect settings of the RealTimelsUniversal registry key to the Console Removed EC2Config dependency on Windows temp folder Removed UserData execution dependency on .Net 3.5
2.2.2	<ul style="list-style-type: none"> Added check to service stop behavior to check that resources are being released Fixed issue with long execution times when joined to domain
2.2.1	<ul style="list-style-type: none"> Updated Installer to allow upgrades from older versions Fixed Ec2WallpaperInfo bug in .Net4.5 only environment Fixed intermittent driver detection bug Added silent install option. Execute Ec2Install.exe with the '-q' option. eg: 'Ec2Install.exe -q'
2.2.0	<ul style="list-style-type: none"> Added support for .Net4 and .Net4.5 only environments Updated Installer

Version	Details
2.1.19	<ul style="list-style-type: none"> Added ephemeral disk labeling support when using Intel network driver (eg. C3 instance Type). For more information, see Enhanced Networking on Windows (p. 684). Added AMI Origin Version and AMI Origin Name support to the console output Made changes to the Console Output for consistent formatting/parsing Updated Help File
2.1.18	<ul style="list-style-type: none"> Added EC2Config WMI Object for Completion notification (-Namespace root\Amazon -Class EC2_ConfigService) Improved Performance of Startup WMI query with large Event Logs; could cause prolonged high CPU during initial execution
2.1.17	<ul style="list-style-type: none"> Fixed UserData execution issue with Standard Output and Standard Error buffer filling Fixed incorrect RDP thumbprint sometimes appearing in Console Output for >= w2k8 OS Console Output now contains 'RDPCERTIFICATE-SubjectName:' for Windows 2008+, which contains the machine name value Added D:\ to Drive Letter Mapping dropdown Moved Help button to top right and changed look/feel Added Feedback survey link to top right
2.1.16	<ul style="list-style-type: none"> General Tab includes link to EC2Config download page for new Versions Desktop Wallpaper overlay now stored in Users Local Appdata folder instead of My Documents to support MyDoc redirection MSSQLServer name sync'd with system in Post-Sysprep script (2008+) Reordered Application Folder (moved files to Plugin directory and removed duplicate files) Changed System Log Output (Console): <ul style="list-style-type: none"> *Moved to a date, name, value format for easier parsing (Please start migrating dependencies to new format) *Added 'Ec2SetPassword' plugin status *Added Sysprep Start and End times Fixed issue of Ephemeral Disks not being labeled as 'Temporary Storage' for non-english Operating Systems Fixed EC2Config Uninstall failure after running Sysprep

Version	Details
2.1.15	<ul style="list-style-type: none"> Optimized requests to the Metadata service Metadata now bypass Proxy Settings Ephemeral Disks labeled as 'Temporary Storage' and Important.txt placed on volume when found (Citrix PV drivers only). For more information, see Upgrading PV Drivers on Your Windows Instances (p. 414). Ephemeral Disks assigned drive letters from Z to A (Citrix PV drivers only) - assignment can be overwritten using Drive Letter Mapping plugin with Volume labels 'Temporary Storage X' where x is a number 0-25) UserData now executes immediately following 'Windows is Ready'
2.1.14	Desktop wallpaper fixes
2.1.13	<ul style="list-style-type: none"> Desktop wallpaper will display hostname by default Removed dependency on Windows Time service Route added in cases where multiple IPs are assigned to a single interface
2.1.11	<ul style="list-style-type: none"> Changes made to Ec2Activation Plugin -Verifies Activation status every 30 days -If Grace Period has 90 days remaining (out of 180), reattempts activation
2.1.10	<ul style="list-style-type: none"> Desktop wallpaper overlay no longer persists with Sysprep or Shutdown without Sysprep Userdata option to execute on every service start with <persist>true</persist> Changed location and name of /DisableWinUpdate.cmd to /Scripts/PostSysprep.cmd Administrator password set to not expire by default in /Scripts/PostSysprep.cmd Uninstall will remove EC2Config PostSysprep script from c:\windows\setup\script\CommandComplete.cmd Add Route supports custom interface metrics
2.1.9	UserData Execution no longer limited to 3851 Characters

Version	Details
2.1.7	<ul style="list-style-type: none"> • OS Version and language identifier written to console • EC2Config version written to console • PV driver version written to console • Detection of Bug Check and output to the console on next boot when found • Option added to config.xml to persist Sysprep credentials • Add Route Retry logic in cases of ENI being unavailable at start • User Data execution PID written to console • Minimum generated password length retrieved from GPO • Set service start to retry 3 attempts • Added S3_DownloadFile.ps1 and S3_Upload file.ps1 examples to /Scripts folder
2.1.6	<ul style="list-style-type: none"> • Version information added to General tab • Renamed the Bundle tab to Image • Simplified the process of specifying passwords and moved the password-related UI from the General tab to the Image tab • Renamed the Disk Settings tab to Storage • Added a Support tab with common tools for troubleshooting • Windows Server 2003 <i>sysprep.ini</i> set to extend OS partition by default • Added the private IP address to the wallpaper • Private IP address displayed on wallpaper • Added retry logic for Console output • Fixed Com port exception for metadata accessibility -- caused EC2Config to terminate before console output is displayed • Checks for activation status on every boot -- activates as necessary • Fixed issue of relative paths -- caused when manually executing wallpaper shortcut from startup folder; pointing to Administrator/logs • Fixed default background color for Windows Server 2003 user (other than Administrator)

Version	Details
2.1.2	<ul style="list-style-type: none"> • Console timestamps in UTC (Zulu) • Removed appearance of hyperlink on Sysprep tab • Addition of feature to dynamically expand Root Volume on first boot for Windows 2008+ • When Set-Password is enabled, now automatically enables EC2Config to set the password • EC2Config checks activation status prior to running Sysprep (presents warning if not activated) • Windows Server 2003 <code>Sysprep.xml</code> now defaults to UTC timezone instead of Pacific • Randomized Activation Servers • Renamed Drive Mapping tab to Disk Settings • Moved Initialize Drives UI items from General to the Disk Settings tab • Help button now points to HTML help file • Updated HTML help file with changes • Updated 'Note' text for Drive Letter Mappings • Added <code>InstallUpdates.ps1</code> to <code>/Scripts</code> folder for automating Patches and cleanup prior to Sysprep
2.1.0	<ul style="list-style-type: none"> • Desktop wallpaper displays instance information by default upon first logon (not disconnect/reconnect) • PowerShell can be executed from the userdata by surrounding the code with <code><powershell></powershell></code>

Subscribing to EC2Config Service Notifications

Amazon SNS can notify you when new versions of the EC2Config service are released. Use the following procedure to subscribe to these notifications.

To subscribe to EC2Config 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 this region because the SNS notifications that you are subscribing to were created in this region.
3. In the navigation pane, choose **Subscriptions**.
4. Choose **Create subscription**.
5. In the **Create subscription** dialog box, do the following:
 - a. For **Topic ARN**, use the following Amazon Resource Name (ARN):

`arn:aws:sns:us-east-1:801119661308:ec2-windows-ec2config`

- b. For **Protocol**, choose **Email**.

- c. For **Endpoint**, type an email address that you can use to receive the notifications.
- d. Choose **Create subscription**.
6. You'll receive an email asking you to confirm your subscription. Open the email and follow the directions to complete your subscription.

Whenever a new version of the EC2Config service is released, we send notifications to subscribers. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from EC2Config notifications

1. Open the Amazon SNS console.
2. In the navigation pane, choose **Subscriptions**.
3. Select the subscription and then choose **Actions, Delete subscriptions**. When prompted for confirmation, choose **Delete**.

Troubleshooting Issues with the EC2Config Service

The following information can help you troubleshoot issues with the EC2Config service.

Update EC2Config on an Unreachable Instance

Use the following procedure to update the EC2Config service on a Windows Server instance that is inaccessible using Remote Desktop.

To update EC2Config on an Amazon EBS-backed Windows instance that you can't connect to

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. Open the context (right-click) menu for the instance, choose **InstanceState**, and then choose **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

4. Choose **Launch Instance** and create a temporary t2.micro instance in the same Availability Zone as the affected instance. Use a different AMI than the one that you used to launch the affected instance.

Important

If you do not create the instance in the same Availability Zone as the affected instance you will not be able to attach the root volume of the affected instance to the new instance.

5. In the EC2 console, choose **Volumes**.
6. Locate the root volume of the affected instance. [Detach \(p. 771\)](#) the volume and [attach \(p. 752\)](#) it to the temporary instance that you created earlier. Attach it with the default device name (xvdf).
7. Use Remote Desktop to connect to the temporary instance, and then use the Disk Management utility to [make the volume available for use \(p. 753\)](#).
8. [Download](#) the latest version of the EC2Config service. Extract the files from the .zip file to the Temp directory on the drive you attached.
9. On the temporary instance, open the Run dialog box, type **regedit**, and press Enter.
10. Choose **HKEY_LOCAL_MACHINE**. From the **File** menu, choose **Load Hive**. Choose the drive and then navigate to and open the following file: `Windows\System32\config\SOFTWARE`. When prompted, specify a key name.

11. Select the key you just loaded and navigate to `Microsoft\Windows\CurrentVersion`. Choose the `RunOnce` key. If this key doesn't exist, choose `CurrentVersion` from the context (right-click) menu, choose **New** and then choose **Key**. Name the key `RunOnce`.
12. From the context (right-click) menu choose the `RunOnce` key, choose **New** and then choose **String Value**. Enter `Ec2Install` as the name and `C:\Temp\Ec2Install.exe /quiet` as the data.
13. Choose the `HKEY_LOCAL_MACHINE\specified key name\Microsoft\Windows NT\CurrentVersion\Winlogon` key. From the context (right-click) menu choose **New**, and then choose **String Value**. Enter `AutoAdminLogon` as the name and `1` as the value data.
14. Choose the `HKEY_LOCAL_MACHINE\specified key name\Microsoft\Windows NT\CurrentVersion\Winlogon>` key. From the context (right-click) menu choose **New**, and then choose **String Value**. Enter `DefaultUserName` as the name and `Administrator` as the value data.
15. Choose the `HKEY_LOCAL_MACHINE\specified key name\Microsoft\Windows NT\CurrentVersion\Winlogon` key. From the context (right-click) menu choose **New**, and then choose **String Value**. Type `DefaultPassword` as the name and enter a password in the value data.
16. In the Registry Editor navigation pane, choose the temporary key that you created when you first opened Registry Editor.
17. From the **File** menu, choose **Unload Hive**.
18. In Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.
19. In the Amazon EC2 console, detach the affected volume from the temporary instance and reattach it to your instance with the device name `/dev/sda1`. You must specify this device name to designate the volume as a root volume.
20. [Stop and Start Your Instance \(p. 362\)](#) the instance.
21. After the instance starts, check the system log and verify that you see the message Windows is ready to use.
22. Open Registry Editor and choose `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon`. Delete the String Value keys you created earlier: `AutoAdminLogon`, `DefaultUserName`, and `DefaultPassword`.
23. Delete or stop the temporary instance you created in this procedure.

Paravirtual Drivers for Windows Instances

Windows AMIs contain a set of drivers to permit access to virtualized hardware. These drivers are used by Amazon EC2 to map instance store and Amazon EBS volumes to their devices. The following table shows key differences between the different drivers.

	RedHat PV	Citrix PV	AWS PV
Instance type	Not supported for all instance types. If you specify an unsupported instance type, the instance is impaired.	Supported for all instance types.	Supported for all instance types.
Attached volumes	Supports up to 16 attached volumes.	Supports more than 16 attached volumes.	Supports more than 16 attached volumes.
Network	The driver has known issues where the network connection resets under high loads; for example, fast FTP file transfers.		The driver automatically configures jumbo frames on

	RedHat PV	Citrix PV	AWS PV
			the network adapter when on a compatible instance type. When the instance is in a cluster placement group (p. 692) , this offers better network performance between instances in the cluster placement group.

The following list shows which PV drivers you should run on each version of Windows Server on Amazon EC2.

- Windows Server 2016: AWS PV
- Windows Server 2012 and 2012 R2: AWS PV
- Windows Server 2008 R2: AWS PV
- Windows Server 2008: Citrix PV 5.9

Contents

- [AWS PV Drivers \(p. 410\)](#)
- [Citrix PV Drivers \(p. 412\)](#)
- [RedHat PV Drivers \(p. 413\)](#)
- [Subscribing to Notifications \(p. 413\)](#)
- [Upgrading PV Drivers on Your Windows Instances \(p. 414\)](#)
- [Troubleshooting PV Drivers \(p. 419\)](#)

AWS PV Drivers

The AWS PV drivers are stored in the %ProgramFiles%\Amazon\Xentools directory. This directory also contains public symbols and a command line tool, `xenstore_client.exe`, that enables you to access entries in XenStore. For example, the following PowerShell command returns the current time from the Hypervisor:

```
PS C:\> [DateTime]::FromFileTimeUTC((gwmi -n root\wmi -cl
AWSXenStoreBase).XenTime).ToString("hh:mm:ss")
11:17:00
```

The AWS PV driver components are listed in the Windows registry under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services`. These driver components are as follows: `XENBUS`, `xeniface`, `xennet`, `xenvbd`, and `xenvif`.

AWS PV also has a driver component named LiteAgent, which runs as a Windows service. It handles tasks such as shutdown and restart events from the API. You can access and manage services by running Services.msc from the command line. This component runs on all instance types including C5 and M5. Updating to AWS PV 8.2 also updates the LiteAgent and adds multiple bug fixes.

Installing the Latest AWS PV Drivers

Amazon Windows AMIs contain a set of drivers to permit access to virtualized hardware. These drivers are used by Amazon EC2 to map instance store and Amazon EBS volumes to their devices. We recommend that you install the latest drivers to improve stability and performance of your EC2 Windows instances.

Installation Options

- You can use AWS Systems Manager to automatically update the PV drivers. For more information, see [Automatically Update PV Drivers on EC2 Windows Instances](#) in the *AWS Systems Manager User Guide*.
- You can download the setup package and run the install program manually. For information about downloading and installing the AWS PV drivers, see [Upgrade Windows Server Instances \(AWS PV Upgrade\)](#) (p. 414).

AWS PV Driver Version History

The following table shows the changes to AWS PV drivers for each driver release.

Driver version	Details	Release date
8.2.5	Additional security enhancements. PowerShell installer now available in package.	12 December 2018
8.2.4	Reliability improvements.	2 October 2018
8.2.3	Bug fixes and performance improvements.	29 May 2018
8.2.1	Network and storage performance improvements plus multiple robustness fixes. To verify that this version has been installed, refer to the following Windows registry value: <code>HKLM\Software\Amazon\PVDriver\Version 8.2.1</code> .	8 March 2018
7.4.6	Stability fixes to make AWS PV drivers more resilient.	26 April 2017
7.4.3	Added support for Windows Server 2016. Stability fixes for all supported Windows OS versions. *AWS PV driver version 7.4.3's signature expires on March 29, 2019. We recommend updating to the latest AWS PV driver.	18 Nov 2016
7.4.2	Stability fixes for support of X1 instance type.	2 Aug 2016
7.4.1	<ul style="list-style-type: none">• Performance improvement in AWS PV Storage driver.• Stability fixes in AWS PV Storage driver: Fixed an issue where the instances were hitting a system crash with bugcheck code 0x0000DEAD.• Stability fixes in AWS PV Network driver.	12 July 2016

Driver version	Details	Release date
	<ul style="list-style-type: none"> Added support for Windows Server 2008R2. 	
7.3.2	<ul style="list-style-type: none"> Improved logging and diagnostics. Stability fix in AWS PV Storage driver. In some cases disks may not surface in Windows after reattaching the disk to the instance. Added support for Windows Server 2012. 	24 June 2015
7.3.1	TRIM update: Fix related to TRIM requests. This fix stabilizes instances and improves instance performance when managing large numbers of TRIM requests.	
7.3.0	TRIM support: The AWS PV driver now sends TRIM requests to the hypervisor. Ephemeral disks will properly process TRIM requests given the underlying storage supports TRIM (SSD). Note that EBS-based storage does not support TRIM as of March 2015.	
7.2.5	<ul style="list-style-type: none"> Stability fix in AWS PV Storage drivers: In some cases the AWS PV driver could dereference invalid memory and cause a system failure. Stability fix while generating a crash dump: In some cases the AWS PV driver could get stuck in a race condition when writing a crash dump. Before this release, the issue could only be resolved by forcing the driver to stop and restart which lost the memory dump. 	
7.2.4	Device ID persistence: This driver fix masks the platform PCI device ID and forces the system to always surface the same device ID, even if the instance is moved. More generally, the fix affects how the hypervisor surfaces virtual devices. The fix also includes modifications to the co-installer for the AWS PV drivers so the system persists mapped virtual devices.	
7.2.2	<ul style="list-style-type: none"> Load the AWS PV drivers in Directory Services Restore Mode (DSRM) mode: Directory Services Restore Mode is a safe mode boot option for Windows Server domain controllers. Persist device ID when virtual network adapter device is reattached: This fix forces the system to check the MAC address mapping and persist the device ID. This fix ensures that adapters retain their static settings if the adapters are reattached. 	
7.2.1	<ul style="list-style-type: none"> Run in safe mode: Fixed an issue where the driver would not load in safe mode. Previously the AWS PV Drivers would only instantiate in normal running systems. Add disks to Microsoft Windows Storage Pools: Previously we synthesized page 83 queries. The fix disabled page 83 support. Note this does not affect storage pools that are used in a cluster environment because PV disks are not valid cluster disks. 	
7.2.0	Base: The AWS PV base version.	

Citrix PV Drivers

The Citrix PV drivers are stored in the %ProgramFiles%\Citrix\XenTools (32-bit instances) or %ProgramFiles(x86)%\Citrix\XenTools (64-bit instances) directory.

The Citrix PV driver components are listed in the Windows registry under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services`. These driver components are as follows: `xenevtchn`, `xeniface`, `xennet`, `Xennet6`, `xensvc`, `xenvbd`, and `xenvif`.

Citrix also has a driver component named `XenGuestAgent`, which runs as a Windows service. It handles tasks such as shutdown and restart events from the API. You can access and manage services by running `Services.msc` from the command line.

If you are encountering networking errors while performing certain workloads, you may need to disable the TCP offloading feature for the Citrix PV driver. For more information, see [TCP Offloading \(p. 423\)](#).

RedHat PV Drivers

RedHat drivers are supported for legacy instances, but are not recommended on newer instances with more than 12GB of RAM due to driver limitations. Instances with more than 12GB of RAM running RedHat drivers can fail to boot and become inaccessible. We recommend upgrading RedHat drivers to Citrix PV drivers, and then upgrade Citrix PV drivers to AWS PV drivers.

The source files for the RedHat drivers are in the `%ProgramFiles%\RedHat` (32-bit instances) or `%ProgramFiles(x86)%\RedHat` (64-bit instances) directory. The two drivers are `rhelnet`, the RedHat Paravirtualized network driver, and `rhelscsi`, the RedHat SCSI miniport driver.

Subscribing to Notifications

Amazon SNS can notify you when new versions of EC2 Windows Drivers are released. Use the following procedure to subscribe to these notifications.

To subscribe to EC2 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 this region because the SNS notifications that you are subscribing to are in this region.
3. In the navigation pane, choose **Subscriptions**.
4. Choose **Create subscription**.
5. In the **Create subscription** dialog box, do the following:
 - a. For **TopicARN**, copy the following Amazon Resource Name (ARN):
`arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers`
 - b. For **Protocol**, choose **Email**.
 - c. For **Endpoint**, type an email address that you can use to receive the notifications.
 - d. Choose **Create subscription**.
6. You'll receive a confirmation email. Open the email and follow the directions to complete your subscription.

Whenever new EC2 Windows drivers are released, we send notifications to subscribers. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from Amazon EC2 Windows driver notification

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v2/home>.
2. In the navigation pane, choose **Subscriptions**.
3. Select the checkbox for the subscription and then choose **Actions, Delete subscriptions**. When prompted for confirmation, choose **Delete**.

Upgrading PV Drivers on Your Windows Instances

To verify which driver your Windows instance uses, open **Network Connections** in Control Panel and view the **Local Area Connection**. Check whether the driver is one of the following:

- AWS PV Network Device
- Citrix PV Ethernet Adapter
- RedHat PV NIC Driver

Alternatively, you can check the output from the `pnputil -e` command.

Contents

- [Upgrade Windows Server Instances \(AWS PV Upgrade\) \(p. 414\)](#)
- [Upgrade a Domain Controller \(AWS PV Upgrade\) \(p. 415\)](#)
- [Upgrade Windows Server 2008 and 2008 R2 Instances \(Redhat to Citrix PV Upgrade\) \(p. 416\)](#)
- [Upgrade Your Citrix Xen Guest Agent Service \(p. 418\)](#)

Upgrade Windows Server Instances (AWS PV Upgrade)

Use the following procedure to perform an in-place upgrade of AWS PV drivers, or to upgrade from Citrix PV drivers to AWS PV drivers on Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, or Windows Server 2016. This upgrade is not available for RedHat drivers, or for other versions of Windows Server.

Important

If your instance is a domain controller, see [Upgrade a Domain Controller \(AWS PV Upgrade\) \(p. 415\)](#). The upgrade process for domain controller instances is different than standard editions of Windows.

To upgrade AWS PV drivers

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Choose the instance that requires the driver upgrade, open the context (right-click) menu, choose **Instance State**, and then choose **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

4. After the instance is stopped, create a backup. Open the context (right-click) menu for the instance, choose **Image**, and then choose **Create Image**.
5. From the context (right-click) menu for the instance, choose **Instance State**, and then choose **Start**.
6. Connect to the instance using Remote Desktop and prepare the instance for upgrade. We recommend that you take all non-system disks offline and note any drive letter mappings to the secondary disks in Disk Management before you perform this upgrade. Note that this step is not required if you are performing an in-place update of AWS PV drivers. We also recommend setting non-essential services to **Manual** start-up in the Services console.
7. [Download](#) the latest driver package to the instance.
8. Extract the contents of the folder and then run `AWSVPDriverSetup.msi`.

After running the MSI, the instance automatically reboots and then upgrades the driver. The instance will not be available for up to 15 minutes. After the upgrade is complete and the instance passes

both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop and verify that the new driver was installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see [AWS PV Driver Version History \(p. 411\)](#) Open Disk Management to review any offline secondary volumes and bring them online corresponding to the drive letters noted in Step 6.

If you previously disabled [TCP Offloading \(p. 423\)](#) using Netsh for Citrix PV drivers we recommend that you re-enable this feature after upgrading to AWS PV drivers. TCP Offloading issues with Citrix drivers are not present in the AWS PV drivers. As a result, TCP Offloading provides better performance with AWS PV drivers.

If you previously applied a static IP address or DNS configuration to the network interface, you must reapply the static IP address or DNS configuration after upgrading AWS PV drivers.

Upgrade a Domain Controller (AWS PV Upgrade)

Use the following procedure on a domain controller to perform either an in-place upgrade of AWS PV drivers, or to upgrade from Citrix PV drivers to AWS PV drivers.

To upgrade a domain controller

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Choose the instance that requires the driver upgrade, open the context (right-click) menu, choose **Instance State**, and then choose **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

4. After the instance is stopped, create a backup. Open the context (right-click) menu for the instance, choose **Image**, and then choose **Create Image**.
5. From the context (right-click) menu for the instance, choose **Instance State**, and then choose **Start**.
6. Run the following command to configure Windows to boot into Directory Services Restore Mode (DSRM):

```
bcdedit /set {default} safeboot dsrepair
```

Warning

Before running this command, confirm that you know the DSRM password. You'll need this information so that you can log in to your instance after the upgrade is complete and the instance automatically reboots.

The system must boot into DSRM because the upgrade utility removes Citrix PV storage drivers so it can install AWS PV drivers. Therefore we recommend noting any drive letter and folder mappings to the secondary disks in Disk Management. When Citrix PV storage drivers are not present, secondary drives will not be detected. Domain controllers that use an NTDS folder on secondary drives will not boot because the secondary disk will not be detected.

Warning

After you run this command do not manually reboot the system. The system will be unreachable because Citrix PV drivers do not support DSRM.

7. Run the following command to add **DisableDCCheck** to the registry:

```
reg add HKLM\SOFTWARE\Wow6432Node\Amazon\AWSPVDriverSetup /v DisableDCCheck /t REG_SZ /d true
```

8. [Download](#) the latest driver package to the instance.
9. Extract the contents of the folder and then run `AWS PV Driver Setup.msi`.

After running the MSI, the instance automatically reboots and then upgrades the driver. The instance will not be available for up to 15 minutes.

10. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop. Open Disk Management to review any offline secondary volumes and bring them online corresponding to the drive letters and folder mappings noted in Step 6.

Important

You must connect to the instance by specifying the user name in the following format `hostname\administrator`. For example, `Win2k12TestBox\administrator`.

11. Run the following command to remove the DSRM boot configuration:

```
bcdeedit /deletevalue safeboot
```

12. Reboot the instance.
13. To complete the upgrade process, verify that the new driver was installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see [AWS PV Driver Version History \(p. 411\)](#).
14. Run the following command to delete `DisableDCCheck` from the registry:

```
reg delete HKLM\SOFTWARE\Wow6432Node\Amazon\AWS PV Driver Setup /v DisableDCCheck
```

Note

If you previously disabled [TCP Offloading \(p. 423\)](#) using Netsh for Citrix PV drivers we recommend that you re-enable this feature after upgrading to AWS PV Drivers. TCP Offloading issues with Citrix drivers are not present in the AWS PV drivers. As a result, TCP Offloading provides better performance with AWS PV drivers.

Upgrade Windows Server 2008 and 2008 R2 Instances (Redhat to Citrix PV Upgrade)

Before you start upgrading your RedHat drivers to Citrix PV drivers, make sure you do the following:

- Install the latest version of the EC2Config service. For more information, see [Installing the Latest Version of EC2Config \(p. 384\)](#).
- Verify that you have Windows PowerShell 2.0 installed. To verify the version that you have installed, run the following command in a PowerShell window:

```
PS C:\> $PSVersionTable.PSVersion
```

If you need to install version 2.0, see [Windows Management Framework \(Windows PowerShell 2.0, WinRM 2.0, and BITS 4.0\)](#) from Microsoft Support.

- Back up your important information on the instance, or create an AMI from the instance. For more information about creating an AMI, see [Creating a Custom Windows AMI \(p. 65\)](#). If you create an AMI, make sure that you do the following:
 - Write down your password.
 - Do not run the Sysprep tool manually or using the EC2Config service.
 - Set your Ethernet adapter to obtain an IP address automatically using DHCP. For more information, see [Configure TCP/IP Settings](#) in the Microsoft TechNet Library.

To upgrade Redhat drivers

1. Connect to your instance and log in as the local administrator. For more information about connecting to your instance, see [Connecting to Your Windows Instance \(p. 359\)](#).
2. In your instance, [download](#) the Citrix PV upgrade package.
3. Extract the contents of the upgrade package to a location of your choice.
4. Double-click the **Upgrade.bat** file. If you get a security warning, choose **Run**.
5. In the **Upgrade Drivers** dialog box, review the information and choose **Yes** if you are ready to start the upgrade.
6. In the **Red Hat Paravirtualized Xen Drivers for Windows uninstaller** dialog box, choose **Yes** to remove the RedHat software. Your instance will be rebooted.

Note

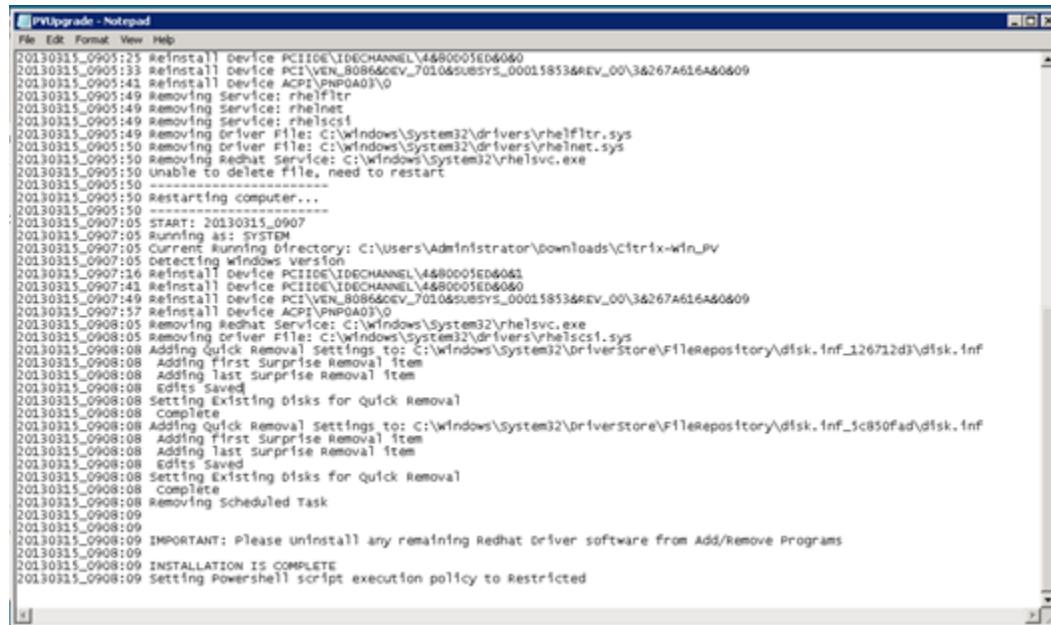
If you do not see the uninstaller dialog box, choose **Red Hat Paravirtualize** in the Windows taskbar.



7. Check that the instance has rebooted and is ready to be used.
 - a. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
 - b. On the **Instances** page, right-click your instance and select **Get System Log**.
 - c. The upgrade operations should have restarted the server 3 or 4 times. You can see this in the log file by the number of times Windows is Ready to use is displayed.

```
Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
Ec2Config service v2.1.9.0
RedHat PV NIC Driver v1.3.10.0
2013/03/15 17:11:01Z: Waiting for meta-data accessibility...
2013/03/15 17:11:02Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D6BF064F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
<Username>Administrator</Username>
<Password>L79ThJPF8LyIIL38IZhtcOFBrjet3vnT2csTiU/GWMrCH7kQtBnznAnXrKd1sirXlx19BwVMsd9b38jFJqv01IUpgNNJRZoCdc7IbUn4</Password>
2013/03/15 17:11:02Z: Product activation was successful.
2013/03/15 17:11:32Z: Message: Windows is Ready to use
Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
Ec2Config service v2.1.9.0
2013/03/15 21:04:24Z: There was an exception writing driver information to console: System.Exception: 
at Ec2Config.Service1.Go()
2013/03/15 21:04:35Z: Waiting for meta-data accessibility...
2013/03/15 21:04:40Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D6BF064F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
2013/03/15 21:05:08Z: Product activation was successful.
2013/03/15 21:05:09Z: Message: Windows is Ready to use
Microsoft Windows NT 6.0.6002 Service Pack 2 (en-US)
Ec2Config service v2.1.9.0
Citrix PV Ethernet Adapter v5.9.960.49119
2013/03/15 21:07:20Z: Waiting for meta-data accessibility...
2013/03/15 21:07:21Z: Meta-data is now available.
<RDPCERTIFICATE>
<THUMBPRINT>D6BF064F21359516C781CA7DF2821C5EFC35648A</THUMBPRINT>
</RDPCERTIFICATE>
2013/03/15 21:07:27Z: Message: Windows is Ready to use
```

8. Connect to your instance and log in as the local administrator.
9. Close the **Red Hat Paravirtualized Xen Drivers for Windows uninstaller** dialog box.
10. Confirm that the installation is complete. Navigate to the **Citrix-WIN_PV** folder that you extracted earlier, open the **PVUpgrade.log** file, and then check for the text **INSTALLATION IS COMPLETE**.



The screenshot shows a Notepad window with the title "PVUpgrade - Notepad". The content of the log file is as follows:

```
20130315_0905:25 Reinstall Device PCIIDE\JDECHANNEL\4&80005ED04060
20130315_0905:33 Reinstall Device PCI\VEN_8086&DEV_7010&SUBSYS_00015853&REV_00\3&267A616A&0&0
20130315_0905:41 Reinstall Device ACPI\PNP0403\0
20130315_0905:49 Removing Service: rhelfltr
20130315_0905:49 Removing Service: rhelnet
20130315_0905:49 Removing Service: rhelscsf
20130315_0905:49 Removing Driver File: C:\Windows\System32\drivers\rhelfltr.sys
20130315_0905:50 Removing Driver File: C:\Windows\System32\drivers\rhelnet.sys
20130315_0905:50 Removing Redhat Service: C:\Windows\System32\rhelsvc.exe
20130315_0905:50 Unable to delete file, need to restart
20130315_0905:50
20130315_0905:50 Restarting computer...
20130315_0905:50
20130315_0907:08 START: 20130315_0907
20130315_0907:08 Running as: SYSTEM
20130315_0907:08 Current Running Directory: C:\users\Administrator\downloads\citrix-wln_PV
20130315_0907:08 Detecting Windows Version
20130315_0907:16 Reinstall Device PCIIDE\JDECHANNEL\4&80005ED04060
20130315_0907:17 Reinstall Device PCIIDE\JDECHANNEL\4&80005ED04060
20130315_0907:19 Reinstall Device PCI\VEN_8086&DEV_7010&SUBSYS_00015853&REV_00\3&267A616A&0&0
20130315_0907:19 Reinstall Device ACPI\PNP0403\0
20130315_0908:05 Removing Redhat Service: C:\Windows\System32\rhelsvc.exe
20130315_0908:05 Removing Driver File: C:\Windows\System32\drivers\rhelscs1.sys
20130315_0908:08 Adding Quick Removal Settings to: C:\Windows\DriverStore\FilterRepository\disk.inf_126712d3\disk.inf
20130315_0908:08 Adding First Surprise Removal Item
20130315_0908:08 Adding Last Surprise Removal Item
20130315_0908:08 Editing Saved
20130315_0908:08 Setting Existing Disks for Quick Removal
20130315_0908:08 Complete
20130315_0908:08 Adding Quick Removal Settings to: C:\Windows\System32\DriverStore\FilterRepository\disk.inf_5c850fad\disk.inf
20130315_0908:08 Adding First Surprise Removal Item
20130315_0908:08 Adding Last Surprise Removal Item
20130315_0908:08 Editing Saved
20130315_0908:08 Setting Existing Disks for Quick Removal
20130315_0908:08 Complete
20130315_0908:08 Removing Scheduled Task
20130315_0908:09
20130315_0908:09
20130315_0908:09 IMPORTANT: Please Uninstall any remaining Redhat Driver software from Add/Remove Programs
20130315_0908:09
20130315_0908:09 INSTALLATION IS COMPLETE
20130315_0908:09 Setting PowerShell script execution policy to Restricted
```

Upgrade Your Citrix Xen Guest Agent Service

If you are using Citrix PV drivers on Windows Server, you can upgrade the Citrix Xen guest agent service. This Windows service handles tasks such as shutdown and restart events from the API. You can run this upgrade package on any version of Windows Server, as long as the instance is running Citrix PV drivers.

Important

Do not perform these steps on Windows Server 2012 or 2012 R2 instances that are running AWS PV drivers.

Before you start upgrading your drivers, make sure you back up your important information on the instance, or create an AMI from the instance. For more information about creating an AMI, see [Creating a Custom Windows AMI \(p. 65\)](#). If you create an AMI, make sure you do the following:

- Do not enable the Sysprep tool in the EC2Config service.
- Write down your password.
- Set your Ethernet adapter to DHCP.

To upgrade your Citrix Xen guest agent service

1. Connect to your instance and log in as the local administrator. For more information about connecting to your instance, see [Connecting to Your Windows Instance \(p. 359\)](#).
2. On your instance, [download](#) the Citrix upgrade package.
3. Extract the contents of the upgrade package to a location of your choice.
4. Double-click the **Upgrade.bat** file. If you get a security warning, choose **Run**.
5. In the **Upgrade Drivers** dialog box, review the information and choose **Yes** if you are ready to start the upgrade.
6. When the upgrade is complete, the **PVUpgrade.log** file will open and contain the text **UPGRADE IS COMPLETE**.
7. Reboot your instance.

Troubleshooting PV Drivers

This topic describes solutions to common issues that you might encounter with Amazon EC2 PV drivers.

Contents

- [Windows Server 2012 R2 loses network and storage connectivity after an instance reboot \(p. 419\)](#)
- [TCP Offloading \(p. 423\)](#)
- [Time Synchronization \(p. 423\)](#)

Windows Server 2012 R2 loses network and storage connectivity after an instance reboot

Windows Server 2012 R2 Amazon Machine Images (AMIs) made available *before* September 10, 2014 can lose network and storage connectivity after an instance reboot. The error in the AWS Management Console system log states: "Difficulty detecting PV driver details for Console Output." The connectivity loss is caused by the Windows Server 2012 R2 Plug and Play Cleanup feature. This feature scans for and disables inactive system devices every 30 days. The feature incorrectly identifies the EC2 network device as inactive and removes it from the system. When this happens, the instance loses network connectivity after a reboot.

For systems that you suspect could be affected by this issue, you can download and run an in-place driver upgrade. If you are unable to perform the in-place driver upgrade, you can run a helper script. The script determines if your instance is affected. If it is affected, and the Amazon EC2 network device has *not* been removed, the script disables the Plug and Play Cleanup scan. If the Amazon EC2 network device has been removed, the script repairs the device, disables the Plug and Play Cleanup scan, and allows your instance to reboot with network connectivity enabled.

In this section

- [Choose How You Want to Fix This Problem \(p. 419\)](#)
- [Method 1 - Enhanced Networking \(p. 420\)](#)
- [Method 2 - Registry configuration \(p. 420\)](#)
- [Run the Remediation Script \(p. 422\)](#)

Choose How You Want to Fix This Problem

There are two methods for restoring network and storage connectivity to an instance affected by this issue. Choose one of the following methods:

Method	Prerequisites	Procedure Overview
Method 1 - Enhanced networking	Enhanced networking is only available in a virtual private cloud (VPC) which requires a C3 instance type. If the server does not currently use the C3 instance type, then you must temporarily change it.	You change the server instance type to a C3 instance. Enhanced networking then enables you to connect to the affected instance and fix the problem. After you fix the problem, you change the instance back to the original instance type. This method is typically faster than Method 2 and less likely to result in user error. You will incur additional charges as long as the C3 instance is running.

Method	Prerequisites	Procedure Overview
Method 2 - Registry configuration	Ability to create or access a second server. Ability to change Registry settings.	You detach the root volume from the affected instance, attach it to a different instance, connect, and make changes in the Registry. You will incur additional charges as long as the additional server is running. This method is slower than Method 1, but this method has worked in situations where Method 1 failed to resolve the problem.

Method 1 - Enhanced Networking

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. Open the context (right-click) menu for the instance, choose **InstanceState**, and then choose **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

4. After the instance is stopped create a backup. Open the context (right-click) menu for the instance, choose **Image**, and then choose **Create Image**.
5. [Change](#) the instance type to any C3 instance type.
6. [Start](#) the instance.
7. Connect to the instance using Remote Desktop and then [download](#) the AWS PV Drivers Upgrade package to the instance.
8. Extract the contents of the folder and run `AWSVPDriverSetup.msi`.

After running the MSI, the instance automatically reboots and then upgrades the drivers. The instance will not be available for up to 15 minutes.

9. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop and verify that the new drivers were installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see [AWS PV Driver Version History \(p. 411\)](#).
10. Stop the instance and change the instance back to its original instance type.
11. Start the instance and resume normal use.

Method 2 - Registry configuration

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. Open the context (right-click) menu for the instance, choose **InstanceState**, and then choose **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

4. Choose **Launch Instance** and create a temporary Windows Server 2008 or Windows Server 2012 instance in the same Availability Zone as the affected instance. Do not create a Windows Server 2012 R2 instance.

Important

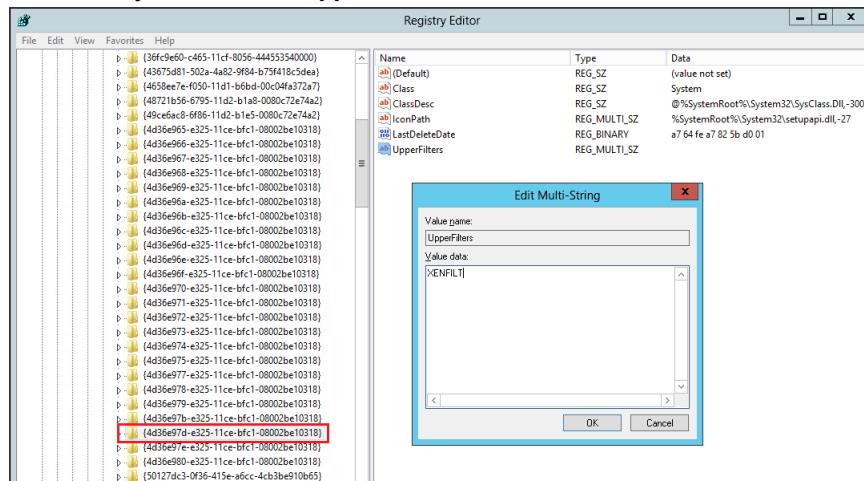
If you do not create the instance in the same Availability Zone as the affected instance you will not be able to attach the root volume of the affected instance to the new instance.

5. In the navigation pane, choose **Volumes**.
6. Locate the root volume of the affected instance. [Detach \(p. 771\)](#) the volume and [attach \(p. 752\)](#) it to the temporary instance you created earlier. Attach it with the default device name (xvdf).
7. Use Remote Desktop to connect to the temporary instance, and then use the Disk Management utility to [make the volume available for use \(p. 753\)](#).
8. On the temporary instance, open the **Run** dialog box, type **regedit**, and press Enter.
9. In the Registry Editor navigation pane, choose **HKEY_LOCAL_MACHINE**, and then from the **File** menu choose **Load Hive**.
10. In the **Load Hive** dialog box, navigate to *Affected Volume\Windows\System32\config\System* and type a temporary name in the **Key Name** dialog box. For example, enter **OldSys**.
11. In the navigation pane of the Registry Editor, locate the following keys:

HKEY_LOCAL_MACHINE\your_temporary_key_name\ControlSet001\Control\Class\4d36e97d-e325-11ce-bfc1-08002be10318

HKEY_LOCAL_MACHINE\your_temporary_key_name\ControlSet001\Control\Class\4d36e96a-e325-11ce-bfc1-08002be10318

12. For each key, double-click **UpperFilters**, enter a value of XENFILT, and then choose **OK**.



13. Locate the following key:

HKEY_LOCAL_MACHINE\your_temporary_key_name\ControlSet001\Services\XENBUS\Parameters

14. Create a new string (REG_SZ) with the name ActiveDevice and the following value:

PCI\VEN_5853&DEV_0001&SUBSYS_00015853&REV_01

15. Locate the following key:

HKEY_LOCAL_MACHINE\your_temporary_key_name\ControlSet001\Services\XENBUS

16. Change the **Count** from 0 to 1.

17. Locate and delete the following keys:

HKEY_LOCAL_MACHINE\your_temporary_key_name\ControlSet001\Services\xenvbd
\StartOverride

HKEY_LOCAL_MACHINE\your_temporary_key_name\ControlSet001\Services\xenfilt
\StartOverride

18. In the Registry Editor navigation pane, choose the temporary key that you created when you first opened the Registry Editor.
19. From the **File** menu, choose **Unload Hive**.
20. In the Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.
21. In the Amazon EC2 console, detach the affected volume from the temporary instance and reattach it to your Windows Server 2012 R2 instance with the device name /dev/sda1. You must specify this device name to designate the volume as a root volume.
22. **Start** the instance.
23. Connect to the instance using Remote Desktop and then [download](#) the AWS PV Drivers Upgrade package to the instance.
24. Extract the contents of the folder and run `AWSPVDriverSetup.msi`.

After running the MSI, the instance automatically reboots and then upgrades the drivers. The instance will not be available for up to 15 minutes.

25. After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop and verify that the new drivers were installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see [AWS PV Driver Version History \(p. 411\)](#).
26. Delete or stop the temporary instance you created in this procedure.

Run the Remediation Script

If you are unable to perform an in-place driver upgrade or migrate to a newer instance you can run the remediation script to fix the problems caused by the Plug and Play Cleanup task.

To run the remediation script

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Choose the instance for which you want to run the remediation script. Open the context (right-click) menu for the instance, choose **Instance State**, and then choose **Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

4. After the instance is stopped create a backup. Open the context (right-click) menu for the instance, choose **Image**, and then choose **Create Image**.
5. Open the context (right-click) menu for the instance, choose **Instance State**, and then choose **Start**.
6. Connect to the instance by using Remote Desktop and then [download](#) the `RemediateDriverIssue.zip` folder to the instance.
7. Extract the contents of the folder.
8. Run the remediation script according to the instructions in the `Readme.txt` file. The file is located in the folder where you extracted `RemediateDriverIssue.zip`.

TCP Offloading

By default, TCP offloading is enabled for the Citrix PV drivers in Windows AMIs. If you encounter transport-level errors or packet transmission errors (as visible on the Windows Performance Monitor)—for example, when you're running certain SQL workloads—you may need to disable this feature.

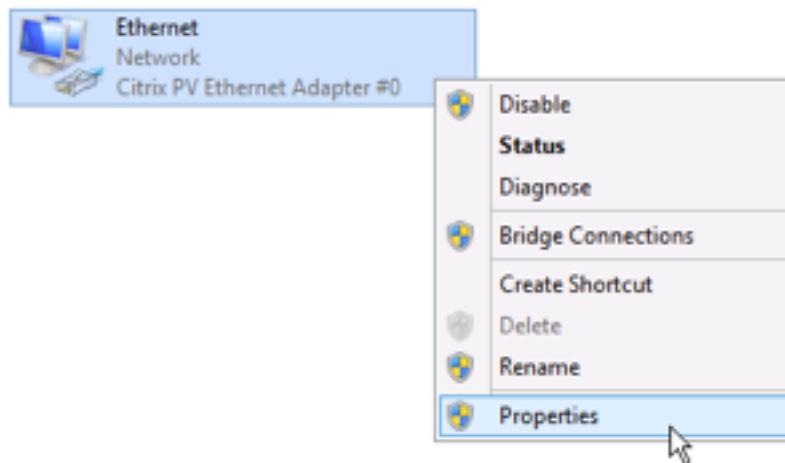
Important

You do not need to perform this procedure on instances running AWS PV or Intel network drivers.

Disabling TCP offloading may reduce the network performance of your instance.

To disable TCP offloading for Windows Server 2012 and 2008

1. Connect to your instance and log in as the local administrator.
2. If you're using Windows Server 2012, press **Ctrl+Esc** to access the **Start** screen, and then choose **Control Panel**. If you're using Windows Server 2008, choose **Start** and select **Control Panel**.
3. Choose **Network and Internet**, then **Network and Sharing Center**.
4. Choose **Change adapter settings**.
5. Right-click **Citrix PV Ethernet Adapter #0** and select **Properties**.



6. In the **Local Area Connection Properties** dialog box, choose **Configure** to open the **Citrix PV Ethernet Adapter #0 Properties** dialog box.
7. On the **Advanced** tab, disable each of the properties, except for **Correct TCP/UDP Checksum Value**. To disable a property, select it from **Property** and choose **Disabled** from **Value**.
8. Choose **OK**.
9. Run the following commands from a Command Prompt window.

```
netsh int ip set global taskoffload=disabled
netsh int tcp set global chimney=disabled
netsh int tcp set global rss=disabled
netsh int tcp set global netdma=disabled
```

10. Reboot the instance.

Time Synchronization

Prior to the release of the 2013.02.13 Windows AMI, the Citrix Xen guest agent could set the system time incorrectly. This can cause your DHCP lease to expire. If you have issues connecting to your instance, you might need to update the agent.

To determine whether you have the updated Citrix Xen guest agent, check whether the C:\Program Files\Citrix\XenGuestAgent.exe file is from March 2013. If the date on this file is earlier than that, update the Citrix Xen guest agent service. For more information, see [Upgrade Your Citrix Xen Guest Agent Service \(p. 418\)](#).

AWS NVMe Drivers for Windows Instances

The latest AWS Windows AMIs of the following Windows operating systems contain the AWS NVMe drivers used to interact with EBS and SSD instance store volumes that are exposed as NVMe block devices for better performance:

- Windows Server 2016
- Windows Server 2012 R2
- Windows Server 2012 RTM
- Windows Server 2008 R2

For more information about EBS and NVMe, see [Amazon EBS and NVMe \(p. 819\)](#). For more information about SSD instance store and NVMe, see [SSD Instance Store Volumes \(p. 850\)](#).

Installing or Upgrading AWS NVMe Drivers

The AWS Windows AMIs provided by Amazon include the AWS NVMe driver. If you are not using the latest AWS Windows AMIs, you can install the current AWS NVMe driver.

To download and install the latest AWS NVMe driver

1. Connect to your instance and log in as the local administrator.
2. [Download](#) the latest driver package to the instance.
3. Extract the zip archive.
4. Install the driver by running the `install.ps1` PowerShell script.
5. If the installer does not reboot your instance for you, restart the instance.

AWS NVMe Driver Version History

The following table describes the released versions of the AWS NVMe driver.

Driver version	Details	Release date
1.3.0	Device optimization improvements	31 August 2018
1.2.0	Performance and reliability improvements for AWS NVMe devices on all supported instances, including bare metal instances	13 June 2018
1.0.0	AWS NVMe driver for supported instance types running Windows Server	12 February 2018

Subscribing to Notifications

Amazon SNS can notify you when new versions of EC2 Windows Drivers are released. Use the following procedure to subscribe to these notifications.

To subscribe to EC2 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 this region because the SNS notifications that you are subscribing to are in this region.
3. In the navigation pane, choose **Subscriptions**.
4. Choose **Create subscription**.
5. In the **Create subscription** dialog box, do the following:
 - a. For **TopicARN**, copy the following Amazon Resource Name (ARN):
`arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers`
 - b. For **Protocol**, choose `Email`.
 - c. For **Endpoint**, type an email address that you can use to receive the notifications.
 - d. Choose **Create subscription**.
6. You'll receive a confirmation email. Open the email and follow the directions to complete your subscription.

Whenever new EC2 Windows drivers are released, we send notifications to subscribers. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from Amazon EC2 Windows driver notification

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v2/home>.
2. In the navigation pane, choose **Subscriptions**.
3. Select the checkbox for the subscription and then choose **Actions, Delete subscriptions**. When prompted for confirmation, choose **Delete**.

Optimizing CPU Options

Amazon EC2 instances support Intel Hyper-Threading Technology (HT 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 HT 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 \(p. 426\)](#)
- [CPU Cores and Threads Per CPU Core Per Instance Type \(p. 426\)](#)
- [Specifying CPU Options for Your Instance \(p. 433\)](#)
- [Viewing the CPU Options for Your Instance \(p. 434\)](#)

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 Amazon EC2 console, the AWS CLI, an AWS SDK, or the Amazon EC2 API.
- 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 more information about examples, see [Specifying CPU Options for Your Instance \(p. 433\)](#).
- The number of vCPUs for the instance is the number of CPU cores multiplied by the threads per core. To specify a custom number of vCPUs, you must specify a valid number of CPU cores and threads per core for the instance type. You cannot exceed the default number of vCPUs for the instance. For more information, see [CPU Cores and Threads Per CPU Core Per Instance Type \(p. 426\)](#).
- To disable Intel HT Technology, specify one thread per core.
- When you [change the instance type \(p. 182\)](#) of an existing instance, the CPU options automatically change to the default CPU options for the new instance type.
- The specified CPU options persist after you stop, start, or reboot an instance.

CPU Cores and Threads Per CPU Core Per Instance Type

The following tables list the instance types that support specifying CPU options. For each type, the table shows the default and supported number of CPU cores and threads per core.

Accelerated Computing Instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
p3.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
p3dn.24xlarge	96	48	2	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
g3s.xlarge	4	2	2	1, 2	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
f1.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
f1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

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

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Optimizing CPU Options

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
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
c5d.large	2	1	2	1	1, 2
c5d.xlarge	4	2	2	2	1, 2
c5d.2xlarge	8	4	2	2, 4	1, 2
c5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
c5d.9xlarge	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
c5d.18xlarge	72	36	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36	1, 2
c5n.large	2	1	2	1	1, 2
c5n.xlarge	4	2	2	2	1, 2
c5n.2xlarge	8	4	2	2, 4	1, 2
c5n.4xlarge	16	8	2	2, 4, 6, 8	1, 2
c5n.9xlarge	36	18	2	2, 4, 6, 8, 10, 12, 14, 16, 18	1, 2
c5n.18xlarge	72	36	2	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
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
m5a.large	2	1	2	1	1, 2
m5a.xlarge	4	2	2	2	1, 2
m5a.2xlarge	8	4	2	2, 4	1, 2
m5a.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5a.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5a.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
m5d.large	2	1	2	1	1, 2
m5d.xlarge	4	2	2	2	1, 2
m5d.2xlarge	8	4	2	2, 4	1, 2
m5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
m5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
m5d.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34,	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
				36, 38, 40, 42, 44, 46, 48	

Memory Optimized Instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
r4.large	2	1	2	1	1, 2
r4.xlarge	4	2	2	1, 2	1, 2
r4.2xlarge	8	4	2	1, 2, 3, 4	1, 2
r4.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
r4.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
r4.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
r5.large	2	1	2	1	1, 2
r5.xlarge	4	2	2	2	1, 2
r5.2xlarge	8	4	2	2, 4	1, 2
r5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r5.24xlarge	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
r5a.large	2	1	2	1	1, 2
r5a.xlarge	4	2	2	2	1, 2
r5a.2xlarge	8	4	2	2, 4	1, 2
r5a.4xlarge	16	8	2	2, 4, 6, 8	1, 2

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Optimizing CPU Options

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
r5a.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r5a.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
r5d.large	2	1	2	1	1, 2
r5d.xlarge	4	2	2	2	1, 2
r5d.2xlarge	8	4	2	2, 4	1, 2
r5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
r5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
r5d.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
x1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
x1.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
x1e.xlarge	4	2	2	1, 2	1, 2
x1e.2xlarge	8	4	2	1, 2, 3, 4	1, 2
x1e.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
x1e.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
x1e.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
x1e.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
z1d.large	2	1	2	1	1, 2
z1d.xlarge	4	2	2	2	1, 2
z1d.2xlarge	8	4	2	2, 4	1, 2
z1d.3xlarge	12	6	2	2, 4, 6	1, 2
z1d.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
z1d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

Storage Optimized Instances

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
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

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
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, 12, 13, 14, 15, 16	1, 2
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 \(p. 430\)](#):

- Default CPU cores: 8
- Default threads per core: 2
- Default vCPUs: 16 ($8 * 2$)
- Valid number of CPU cores: 1, 2, 3, 4, 5, 6, 7, 8
- Valid number of threads per core: 1, 2

Disabling Intel HT Technology

To disable Intel HT Technology, specify one thread per core.

To disable Intel HT Technology during instance launch (Console)

1. Follow the [Launching an Instance Using the Launch Instance Wizard \(p. 315\)](#) procedure.
2. On the **Configure Instance Details** page, for **CPU options**, choose **Specify CPU options**.
3. For **Core count**, choose the number of required CPU cores. In this example, to specify the default CPU core count for an `r4.4xlarge` instance, choose 8.
4. To disable Intel HT Technology, for **Threads per core**, choose 1.
5. Continue as prompted by the wizard. When you've finished reviewing your options on the **Review Instance Launch** page, choose **Launch**. For more information, see [Launching an Instance Using the Launch Instance Wizard \(p. 315\)](#).

To disable Intel HT Technology during instance launch (AWS CLI)

- Use the `run-instances` AWS CLI command and specify a value of 1 for `ThreadsPerCore` for the `--cpu-options` parameter. For `CoreCount`, specify the number of CPU cores. In this example, to specify the default CPU core count for an `r4.4xlarge` instance, specify a value of 8.

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options "CoreCount=8,ThreadsPerCore=1" --key-name MyKeyPair
```

Specifying a Custom Number of vCPUs

You can customize the number of CPU cores and threads per core for the instance.

To specify a custom number of vCPUs during instance launch (console)

The following example launches an `r4.4xlarge` instance with six vCPUs.

1. Follow the [Launching an Instance Using the Launch Instance Wizard \(p. 315\)](#) procedure.
2. On the **Configure Instance Details** page, for **CPU options**, choose **Specify CPU options**.
3. To get six vCPUs, specify three CPU cores and two threads per core, as follows:
 - For **Core count**, choose **3**.
 - For **Threads per core**, choose **2**.
4. Continue as prompted by the wizard. When you've finished reviewing your options on the **Review Instance Launch** page, choose **Launch**. For more information, see [Launching an Instance Using the Launch Instance Wizard \(p. 315\)](#).

To specify a custom number of vCPUs during instance launch (AWS CLI)

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 MyKeyPair
```

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 MyKeyPair
```

Viewing the CPU Options for Your Instance

You can view the CPU options for an existing instance in the Amazon EC2 console or by describing the instance using the AWS CLI.

To view the CPU options for an instance (Console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, choose **Instances**, and select the instance.
3. Choose **Description** and view the **Number of vCPUs** field.
4. To view the core count and threads per core, choose the **Number of vCPUs** field value.

To view the CPU options for an instance (AWS CLI)

- 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 Task Manager 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](#).

Setting the Time for a Windows 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. We recommend that you use Coordinated Universal Time (UTC) for your Windows instances. However, you can use a different time zone if you want.

Contents

- [Changing the Time Zone \(p. 436\)](#)
- [Configuring Network Time Protocol \(NTP\) \(p. 436\)](#)
- [Default Network Time Protocol \(NTP\) Settings for Amazon Windows AMIs \(p. 437\)](#)
- [Configuring Time Settings for Windows Server 2008 and later \(p. 438\)](#)
- [Related Topics \(p. 439\)](#)

Changing the Time Zone

Windows instances are set to the UTC time zone by default. You can change the time to correspond to your local time zone or a time zone for another part of your network.

To change the time zone on an instance

1. From your instance, open a Command Prompt window.
2. Identify the time zone to use on the instance. To get a list of time zones, use the following command: `tzutil /l`. This command returns a list of all available time zones, using the following format:

```
display name  
time zone ID
```

3. Locate the time zone ID to assign to the instance.
4. Assign the time zone to the instance by using the following command:

```
tzutil /s "Pacific Standard Time"
```

The new time zone should take effect immediately.

Configuring Network Time Protocol (NTP)

Windows instances use the `time.windows.com` NTP server to configure the system time. We recommend that you configure your instance to use the Amazon Time Sync Service. This service uses a fleet of satellite-connected and atomic reference clocks in each AWS Region to deliver accurate current time readings of the Coordinated Universal Time (UTC) global standard. The Amazon Time Sync Service automatically smooths any leap seconds that are added to UTC. This service is available at the `169.254.169.123` IP address for any instance running in a VPC, and your instance does not require internet access to use this service.

To verify the NTP configuration

1. From your instance, open a Command Prompt window.
2. Get the current NTP configuration by typing the following command:

```
w32tm /query /configuration
```

This command returns the current configuration settings for the Windows instance.

3. (Optional) Get the status of the current configuration by typing the following command:

```
w32tm /query /status
```

This command returns information such as the last time the instance synced with the NTP server and the poll interval.

To change the NTP server to use the Amazon Time Sync Service

1. From the Command Prompt window, run the following command:

```
w32tm /config /manualpeerlist:169.254.169.123 /syncfromflags:manual /update
```

- Verify your new settings by using the following command:

```
w32tm /query /configuration
```

In the output that's returned, verify that `NtpServer` displays the `169.254.169.123` IP address.

You can change the instance to use a different set of NTP servers if you need to. For example, if you have Windows instances that do not have internet access, you can configure them to use an NTP server located within your private network. Your instance's security group must allow outbound UDP traffic on port 123 (NTP).

To change the NTP servers

- From the Command Prompt window, run the following command:

```
w32tm /config /manualpeerlist:comma-delimited list of NTP servers /  
syncfromflags:manual /update
```

Where `comma-delimited list of NTP servers` is the list of NTP servers for the instance to use.

- Verify your new settings by using the following command:

```
w32tm /query /configuration
```

Default Network Time Protocol (NTP) Settings for Amazon Windows AMIs

Amazon Machine Images (AMIs) generally adhere to the out-of-the-box defaults except in cases where changes are required to function on EC2 infrastructure. The following settings have been determined to work well in a virtualized environment, as well as to keep any clock drift to within one second of accuracy:

- Update Interval** – governs how frequently the time service will adjust system time towards accuracy. AWS configures the update interval to occur once every two minutes.
- NTP Server** – starting with the August 2018 release, AMIs will now use the AWS time service by default. This time service is accessible from any EC2 region at the `169.254.169.123` endpoint. Additionally, the `0x9` flag indicates that the time service is acting as a client, and to use “`SpecialPollInterval`” to determine how frequently to check in with the configured time server.
- Type** – “NTP” means that the service will act as a standalone NTP client as opposed to acting as part of a domain.
- Enabled and InputProvider** – the time service is enabled and provides time to the operating system.
- Special Poll Interval** – checks against the configured NTP Server every 900 seconds, or 15 minutes.

Registry Path	Key Name	Data
<code>HKLM:\System\CurrentControlSet\services\w32time\Config</code>	<code>UpdateInterval</code>	120

Registry Path	Key Name	Data
HKLM:\System\CurrentControlSet\services\w32time\Parameters	NtpServer	169.254.169.123,0x9
HKLM:\System\CurrentControlSet\services\w32time\Parameters	Type	NTP
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	Enabled	1
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	InputProvider	1
HKLM:\System\CurrentControlSet\services\w32time\TimeProviders\NtpClient	SpecialPollInterval	900

Configuring Time Settings for Windows Server 2008 and later

When you change the time on a Windows instance, you must ensure that the time persists through system restarts. Otherwise, when the instance restarts, it reverts back to using UTC time. For Windows Server 2008 and later, you can persist your time setting by adding a **RealTimelsUniversal** registry key.

To set the **RealTimelsUniversal** registry key

- From the instance, open a Command Prompt window.
- Use the following command to add the registry key:

```
reg add "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation" /v RealTimeIsUniversal /d 1 /t REG_DWORD /f
```

- If you are using a Windows Server 2008 AMI (*not* Windows Server 2008 R2) that was created before February 22, 2013, you should verify that the Microsoft hotfix [KB2800213](#) is installed. If this hotfix is not installed, we recommend updating to the latest AWS Windows AMI. If you are using an AMI running Windows Server 2008 R2 (*not* Windows Server 2008), you must verify that the Microsoft hotfix [KB2922223](#) is installed. If this hotfix is not installed, we recommend updating to the latest AWS Windows AMI.
- (Optional) Verify that the instance saved the key successfully using the following command:

```
reg query "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation" /s
```

This command returns the subkeys for the **TimeZoneInformation** registry key. You should see the **RealTimelsUniversal** key at the bottom of the list, similar to the following:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation	
Bias	REG_DWORD 0x1e0
DaylightBias	REG_DWORD 0xffffffffc4
DaylightName	REG_SZ @tzres.dll,-211

DaylightStart	REG_BINARY	000003000200020000000000000000000000000000
StandardBias	REG_DWORD	0x0
StandardName	REG_SZ	@tzres.dll,-212
StandardStart	REG_BINARY	00000B0001000200000000000000000000000000
TimeZoneKeyName	REG_SZ	Pacific Standard Time
DynamicDaylightTimeDisabled	REG_DWORD	0x0
ActiveTimeBias	REG_DWORD	0x1a4
RealTimeIsUniversal	REG_DWORD	0x1

Related Topics

For more information about how the Windows operating system coordinates and manages time, including the addition of a leap second, see the following documentation:

- [How the Windows Time Service Works](#) (Microsoft)
- [W32tm](#) (Microsoft)
- [How the Windows Time service treats a leap second](#) (Microsoft)
- [The story around Leap Seconds and Windows: It's likely not Y2K](#) (Microsoft)

Setting Passwords for Windows Instances

When you connect to a Windows instance, you must specify a user account and password that has permission to access the instance. The first time that you connect to an instance, you are prompted to specify the Administrator account and the default password.

With AWS Windows AMIs for Windows Server 2012 R2 and earlier, the [EC2Config service](#) (p. 383) generates the default password. With AWS Windows AMIs for Windows Server 2016, [EC2Launch](#) (p. 374) generates the default password.

Note

With Windows Server 2016, `Password never expires` is disabled for the local administrator. With Windows Server 2012 R2 and earlier, `Password never expires` is enabled for the local administrator.

Changing the Administrator Password After Connecting

When you connect to an instance the first time, we recommend that you change the Administrator password from its default value. Use the following procedure to change the Administrator password for a Windows instance.

Important

Store the new password in a safe place. You won't be able to retrieve the new password using the Amazon EC2 console. The console can only retrieve the default password. If you attempt to connect to the instance using the default password after changing it, you'll get a "Your credentials did not work" error.

To change the local Administrator password

1. Connect to the instance and open a command prompt.
2. Run the following command. If your new password includes special characters, ensure that you enclose the password in double quotes:

```
net user Administrator "new_password"
```

3. Store the new password in a safe place.

Changing a Lost or Expired Password

If you lose your password or it expires, you can generate a new password. For password reset procedures, see [Resetting a Lost or Expired Windows Administrator Password \(p. 974\)](#).

Adding Windows Components Using Installation Media

Windows Server operating systems include many optional components. Including all optional components in each Amazon EC2 Windows Server AMI is not practical. Instead, we provide you with installation media EBS snapshots that have the necessary files to configure or install components on your Windows instance.

To access and install the optional components, you must find the correct EBS snapshot for your version of Windows Server, create a volume from the snapshot, and attach the volume to your instance.

Before You Begin

Use the AWS Management Console or a command line tool to get the instance ID and Availability Zone of your instance. You must create your EBS volume in the same Availability Zone as your instance.

Adding Windows Components Using the Console

Use the following procedure to use the AWS Management Console to add Windows components to your instance.

To add Windows components to your instance using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Snapshots**.
3. From the **Filter** bar, choose **Public Snapshots**.
4. Add the **Owner** filter and choose **Amazon images**.
5. Add the **Description** filter and type **Windows**.
6. Press Enter
7. Select the snapshot that matches your system architecture and language preference. For example, select **Windows 2016 English Installation Media** if your instance is running Windows Server 2016.
8. Choose **Actions, Create Volume**.
9. In the **Create Volume** dialog box, select the Availability Zone that matches your Windows instance, and then choose **Create**.
10. In the **Volume Successfully Created** message, choose the volume that you just created.
11. Choose **Actions, Attach Volume**.
12. In the **Attach Volume** dialog box, type the instance ID and the name of the device for the attachment, and choose **Attach**. If you need help with the device name, see [Device Naming](#).
13. Connect to your instance and make the volume available. For more information, see [Making an Amazon EBS Volume Available for Use on Windows \(p. 753\)](#).

Important

Do not initialize the volume.

14. Open **Control Panel, Programs and Features**. Choose **Turn Windows features on or off**. If you are prompted for installation media, specify the EBS volume with the installation media.

Adding Windows Components Using the Tools for Windows PowerShell

Use the following procedure to use the Tools for Windows PowerShell to add Windows components to your instance.

To add Windows components to your instance using the Tools for Windows PowerShell

1. Use the [Get-EC2Snapshot](#) cmdlet with the `Owner` and `description` filters to get a list of the available installation media snapshots.

```
PS C:\> Get-EC2Snapshot -Owner amazon -Filter @{ Name="description"; Values="Windows*" }
```

2. In the output, note the ID of the snapshot that matches your system architecture and language preference. For example:

```
...
DataEncryptionKeyId :
Description      : Windows 2016 English Installation Media
Encrypted        : False
KmsKeyId         :
OwnerAlias       : amazon
OwnerId          : 123456789012
Progress         : 100%
SnapshotId       : snap-22da283e
StartTime        : 10/25/2016 8:00:47 PM
State            : completed
StateMessage     :
Tags             : {}
VolumeId         : vol-be5eafcb
VolumeSize       : 6
...
```

3. Use the [New-EC2Volume](#) cmdlet to create a volume from the snapshot. Specify the same Availability Zone as your instance.

```
PS C:\> New-EC2Volume -AvailabilityZone us-east-1a -VolumeType gp2 -
SnapshotId snap-22da283e
```

4. In the output, note the volume ID.

```
Attachments      : {}
AvailabilityZone : us-east-1a
CreateTime       : 4/18/2017 10:50:25 AM
Encrypted        : False
Iops             : 100
KmsKeyId         :
Size             : 6
SnapshotId       : snap-22da283e
State            : creating
Tags             : {}
VolumeId         : vol-06aa9e1fbf8b82ed1
VolumeType       : gp2
```

5. Use the [Add-EC2Volume](#) cmdlet to attach the volume to your instance.

```
PS C:\> Add-EC2Volume -InstanceId i-087711ddaf98f9489 -VolumeId vol-06aa9e1fbf8b82ed1 -
Device xvdh
```

6. Connect to your instance and make the volume available. For more information, see [Making an Amazon EBS Volume Available for Use on Windows \(p. 753\)](#).

Important

Do not initialize the volume.

7. Open **Control Panel, Programs and Features**. Choose **Turn Windows features on or off**. If you are prompted for installation media, specify the EBS volume with the installation media.

Adding Windows Components Using the AWS CLI

Use the following procedure to use the AWS CLI to add Windows components to your instance.

To add Windows components to your instance using the AWS CLI

1. Use the [describe-snapshots](#) command with the `owner-ids` parameter and `description` filter to get a list of the available installation media snapshots.

```
aws ec2 describe-snapshots --owner-ids amazon --filters
    Name=description,Values=Windows*
```

2. In the output, note the ID of the snapshot that matches your system architecture and language preference. For example:

```
{
  "Snapshots": [
    ...
    {
      "OwnerAlias": "amazon",
      "Description": "Windows 2016 English Installation Media",
      "Encrypted": false,
      "VolumeId": "vol-be5eafcb",
      "State": "completed",
      "VolumeSize": 6,
      "Progress": "100%",
      "StartTime": "2016-10-25T20:00:47.000Z",
      "SnapshotId": "snap-22da283e",
      "OwnerId": "123456789012"
    },
    ...
  ]
}
```

3. Use the [create-volume](#) command to create a volume from the snapshot. Specify the same Availability Zone as your instance.

```
aws ec2 create-volume --snapshot-id snap-22da283e --volume-type gp2 --availability-
zone us-east-1a
```

4. In the output, note the volume ID.

```
{
  "AvailabilityZone": "us-east-1a",
  "Encrypted": false,
  "VolumeType": "gp2",
  "VolumeId": "vol-0c98b37f30bcfc290",
  "State": "creating",
  "Iops": 100,
  "SnapshotId": "snap-22da283e",
  "CreateTime": "2017-04-18T10:33:10.940Z",
  "Size": 6
```

}

5. Use the [attach-volume](#) command to attach the volume to your instance.

```
aws ec2 attach-volume --volume-id vol-0c98b37f30bcfc290 --instance-id i-01474ef662b89480 --device xvdf
```

6. Connect to your instance and make the volume available. For more information, see [Making an Amazon EBS Volume Available for Use on Windows \(p. 753\)](#).

Important

Do not initialize the volume.

7. Open **Control Panel, Programs and Features**. Choose **Turn Windows features on or off**. If you are prompted for installation media, specify the EBS volume with the installation media.

Configuring a Secondary Private IPv4 Address for Your Windows Instance

You can specify multiple private IPv4 addresses for your instances. After you assign a secondary private IPv4 address to an instance, you must configure the operating system on the instance to recognize the secondary private IPv4 address.

Configuring the operating system on a Windows instance to recognize a secondary private IPv4 address requires the following:

- [Step 1: Configure Static IP Addressing on Your Windows Instance \(p. 443\)](#)
- [Step 2: Configure a Secondary Private IP Address for Your Windows Instance \(p. 445\)](#)
- [Step 3: Configure Applications to Use the Secondary Private IP Address \(p. 446\)](#)

Note

These instructions are based on Windows Server 2008 R2. The implementation of these steps may vary based on the operating system of the Windows instance.

Prerequisites

Before you begin, make sure you meet the following requirements:

- As a best practice, launch your Windows instances using the latest AMIs. If you are using an older Windows AMI, ensure that it has the Microsoft hot fix referenced in <http://support.microsoft.com/kb/2582281>.
- After you launch your instance in your VPC, add a secondary private IP address. For more information, see [Assigning a Secondary Private IPv4 Address \(p. 651\)](#).
- To allow Internet requests to your website after you complete the tasks in these steps, you must configure an Elastic IP address and associate it with the secondary private IP address. For more information, see [Associating an Elastic IP Address with the Secondary Private IPv4 Address \(p. 653\)](#).

Step 1: Configure Static IP Addressing on Your Windows Instance

To enable your Windows instance to use multiple IP addresses, you must configure your instance to use static IP addressing rather than a DHCP server.

Important

When you configure static IP addressing on your instance, the IP address must match exactly what is shown in the console, CLI, or API. If you enter these IP addresses incorrectly, the instance could become unreachable.

To configure static IP addressing on a Windows instance

1. Connect to your instance.
2. Find the IP address, subnet mask, and default gateway addresses for the instance by performing the following steps:
 - At a Command Prompt window, run the following command:

```
ipconfig /all
```

Review the following section in your output, and note the **IPv4 Address**, **Subnet Mask**, **Default Gateway**, and **DNS Servers** values for the network interface.

```
Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix  . :
Description . . . . . :
Physical Address . . . . . :
DHCP Enabled. . . . . :
Autoconfiguration Enabled . . . . . :
IPv4 Address. . . . . : 10.0.0.131
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.0.0.1
DNS Servers . . . . . : 10.1.1.10
                                         10.1.1.20
```

3. Open the **Network and Sharing Center** by running the following command:

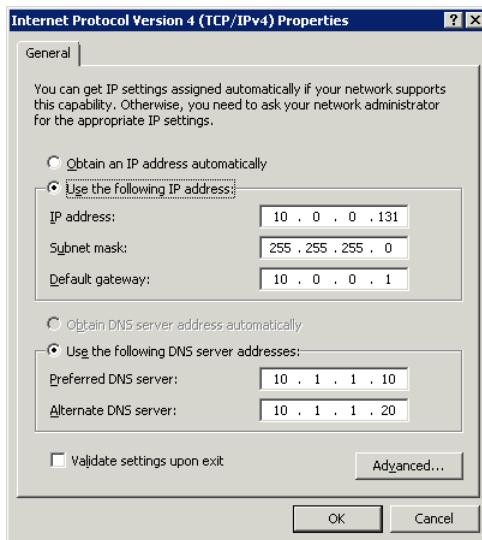
```
%SystemRoot%\system32\control.exe ncpa.cpl
```

4. Open the context (right-click) menu for the network interface (Local Area Connection) and choose **Properties**.
5. Choose **Internet Protocol Version 4 (TCP/IPv4)**, **Properties**.
6. In the **Internet Protocol Version 4 (TCP/IPv4)** **Properties** dialog box, choose **Use the following IP address**, enter the following values, and then choose **OK**.

Field	Value
IP address	The IPv4 address obtained in step 2 above.
Subnet mask	The subnet mask obtained in step 2 above.
Default gateway	The default gateway address obtained in step 2 above.
Preferred DNS server	The DNS server obtained in step 2 above.
Alternate DNS server	The alternate DNS server obtained in step 2 above. If an alternate DNS server was not listed, leave this field blank.

Important

If you set the IP address to any value other than the current IP address, you will lose connectivity to the instance.



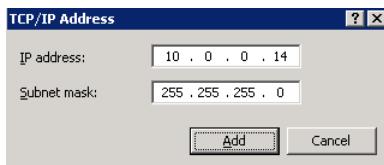
You will lose RDP connectivity to the Windows instance for a few seconds while the instance converts from using DHCP to static addressing. The instance retains the same IP address information as before, but now this information is static and not managed by DHCP.

Step 2: Configure a Secondary Private IP Address for Your Windows Instance

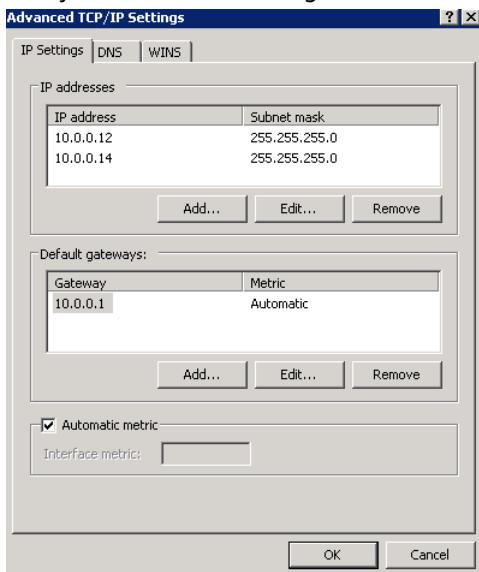
After you have set up static IP addressing on your Windows instance, you are ready to prepare a second private IP address.

To configure a secondary IP address for a Windows instance

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. On the **Description** tab, note the secondary IP address.
4. Connect to your instance.
5. On your Windows instance, choose **Start, Control Panel**.
6. Choose **Network and Internet, Network and Sharing Center**.
7. Select the network interface (Local Area Connection) and choose **Properties**.
8. On the **Local Area Connection Properties** page, choose **Internet Protocol Version 4 (TCP/IPv4), Properties, Advanced**.
9. Choose **Add**.
10. In the **TCP/IP Address** dialog box, type the secondary private IP address for **IP address**. For **Subnet mask**, type the same subnet mask that you entered for the primary private IP address in [Step 1: Configure Static IP Addressing on Your Windows Instance \(p. 443\)](#), and then choose **Add**.



11. Verify the IP address settings and choose **OK**.



12. Choose **OK, Close**.
13. To confirm that the secondary IP address has been added to the operating system, at a command prompt, run the command **ipconfig /all**.

Step 3: Configure Applications to Use the Secondary Private IP Address

You can configure any applications to use the secondary private IP address. For example, if your instance is running a website on IIS, you can configure IIS to use the secondary private IP address.

To configure IIS to use the secondary private IP address

1. Connect to your instance.
2. Open Internet Information Services (IIS) Manager.
3. In the **Connections** pane, expand **Sites**.
4. Open the context (right-click) menu for your website and choose **Edit Bindings**.
5. In the **Site Bindings** dialog box, for **Type**, choose **http, Edit**.
6. In the **Edit Site Binding** dialog box, for **IP address**, select the secondary private IP address. (By default, each website accepts HTTP requests from all IP addresses.)



7. Choose **OK**, **Close**.

Configure a Secondary Elastic Network Interface

You can attach a second elastic network interface to the instance.

To configure a second network interface

1. Configure the static IP addressing for the primary elastic network interface as per the procedures above in [Step 1: Configure Static IP Addressing on Your Windows Instance \(p. 443\)](#).
2. Configure the static IP addressing for the secondary elastic network interface as per the same procedures.

Running Commands on Your Windows Instance at Launch

When you launch a Windows instance in Amazon EC2, you can pass user data to the instance. Instance user data is treated as opaque data; it is up to the instance to interpret it. For example, you can specify data to be used by automated configuration tasks or specify scripts that are run after the instance starts. User data is processed by [EC2Config \(p. 383\)](#) on Windows Server 2012 R2 and earlier and by [EC2Launch \(p. 374\)](#) on Windows Server 2016.

Contents

- [User Data Scripts \(p. 447\)](#)
- [User Data Execution \(p. 449\)](#)
- [User Data and the Console \(p. 450\)](#)
- [User Data and the Tools for Windows PowerShell \(p. 451\)](#)

User Data Scripts

For EC2Config or EC2Launch to execute scripts, you must enclose the script within a special tag when you add it to user data. The tag you use depends on whether the commands run in a Command Prompt window (batch commands) or using Windows PowerShell.

If you specify both a batch script and a Windows PowerShell script, the batch script runs first and the Windows PowerShell script runs next, regardless of the order in which they appear in the instance user data.

Syntax for Batch Scripts

Specify a batch script using the `<script>` tag. Separate the commands using line breaks. For example:

```
<script>
echo Current date and time >> %SystemRoot%\Temp\test.log
echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
```

By default, the user data scripts are executed one time when you launch the instance. To execute the user data scripts every time you reboot or start the instance, add `<persist>true</persist>` to the user data.

```
<script>
echo Current date and time >> %SystemRoot%\Temp\test.log
echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
</script>
<persist>true</persist>
```

Syntax for Windows PowerShell Scripts

The AWS Windows AMIs include the [AWS Tools for Windows PowerShell](#), so you can specify these cmdlets in user data. If you associate an IAM role with your instance, you don't need to specify credentials to the cmdlets, as applications that run on the instance use the role's credentials to access AWS resources (for example, Amazon S3 buckets).

Specify a Windows PowerShell script using the `powershell` tag. Separate the commands using line breaks. For example:

```
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
```

By default, the user data scripts are executed one time when you launch the instance. To execute the user data scripts every time you reboot or start the instance, add `<persist>true</persist>` to the user data.

```
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

Base64 Encoding

If you're using the Amazon EC2 API or a tool that does not perform base64 encoding of the user data, you must encode the user data yourself. If not, an error is logged about being unable to find `script` or `powershell` tags to execute. The following is an example that encodes using Windows PowerShell.

```
$UserData =
[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Script))
```

The following is an example that decodes using PowerShell.

```
$Script =
[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($UserData))
```

For more information about base64 encoding, see <http://tools.ietf.org/html/rfc4648>.

User Data Execution

By default, all AWS Windows AMIs have user data execution enabled for the initial launch. You can specify that user data scripts are executed the next time the instance reboots or restarts. Alternatively, you can specify that user data scripts are executed every time the instance reboots or restarts.

User data scripts are executed from the local administrator account when a random password is generated. Otherwise, user data scripts are executed from the System account.

Instance Launch

Any scripts in the instance user data are executed during the initial launch of the instance. If the persist tag is found, user data execution is enabled for subsequent reboots or starts. The log files for EC2Launch and EC2Config contain the output from the standard output and standard error streams.

With EC2Launch, the log file is C:\ProgramData\Amazon\EC2-Windows\Launch\Log\userdataExecution.log. The following information is logged as user data is executed.

- Userdata execution begins — The start of user data execution
- <persist> tag was provided: true — If the persist tag is found
- Running userdata on every boot — If the persist tag is found
- <powershell> tag was provided.. running powershell content — If the powershell tag is found
- <script> tag was provided.. running script content — If the script tag is found
- Message: The output from user scripts — If user data scripts are executed, their output is logged

With EC2Config, the log file is C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2Config.log. The following information is logged as user data is executed.

- Ec2HandleUserData: Message: Start running user scripts — The start of user data execution
- Ec2HandleUserData: Message: Re-enabled userdata execution — If the persist tag is found
- Ec2HandleUserData: Message: Could not find <persist> and </persist> — If the persist tag is not found
- Ec2HandleUserData: Message: The output from user scripts — If user data scripts are executed, their output is logged

Subsequent Reboots or Starts

When you update instance user data, user data scripts are not executed automatically when you reboot or start the instance. However, you can enable user data execution so that user data scripts are executed one time when you reboot or start the instance or every time you reboot or start the instance.

If you choose the **Shutdown with Sysprep** option, user data scripts are executed when the instance is rebooted or restarted, even if you did not enable user data execution for subsequent reboots or starts.

To enable user data execution on Windows Server 2016 (EC2Launch)

1. Connect to your Windows instance.
2. Open a PowerShell command window and run the following command:

```
InitializeInstance.ps1 -Schedule
```

3. Disconnect from your Windows instance. To execute updated scripts next time the instance is started, stop the instance and update the user data. For more information, see [View and Update the Instance User Data \(p. 450\)](#).

To enable user data execution on Windows Server 2012 R2 and earlier (EC2Config)

1. Connect to your Windows instance.
2. Open C:\Program Files\Amazon\Ec2ConfigService\Ec2ConfigServiceSetting.exe.
3. For **User Data**, select **Enable UserData execution for next service start**.
4. Disconnect from your Windows instance. To execute updated scripts next time the instance is started, stop the instance and update the user data. For more information, see [View and Update the Instance User Data \(p. 450\)](#).

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

When you launch an instance, you specify the script in **Advanced Details, User data** on the **Step 3: Configure Instance Details** page of the Launch Instance wizard. The example in the following image creates a file in the Windows temporary folder, using the current date and time in the file name. When you include <persist>true</persist>, the script is executed every time you reboot or start the instance. When you select **As text**, the Amazon EC2 console performs the base64 encoding for you.

▼ Advanced Details

User data

As text As file Input is already base64 encoded

```
<powershell>
$file = $env:SystemRoot+"\Temp\"+(Get-Date).ToString("MM-dd-yyyy")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

View and Update the Instance User Data

You can view the instance user data for any instance, and you can update the instance user data for a stopped instance.

To update the user data for an instance using the console

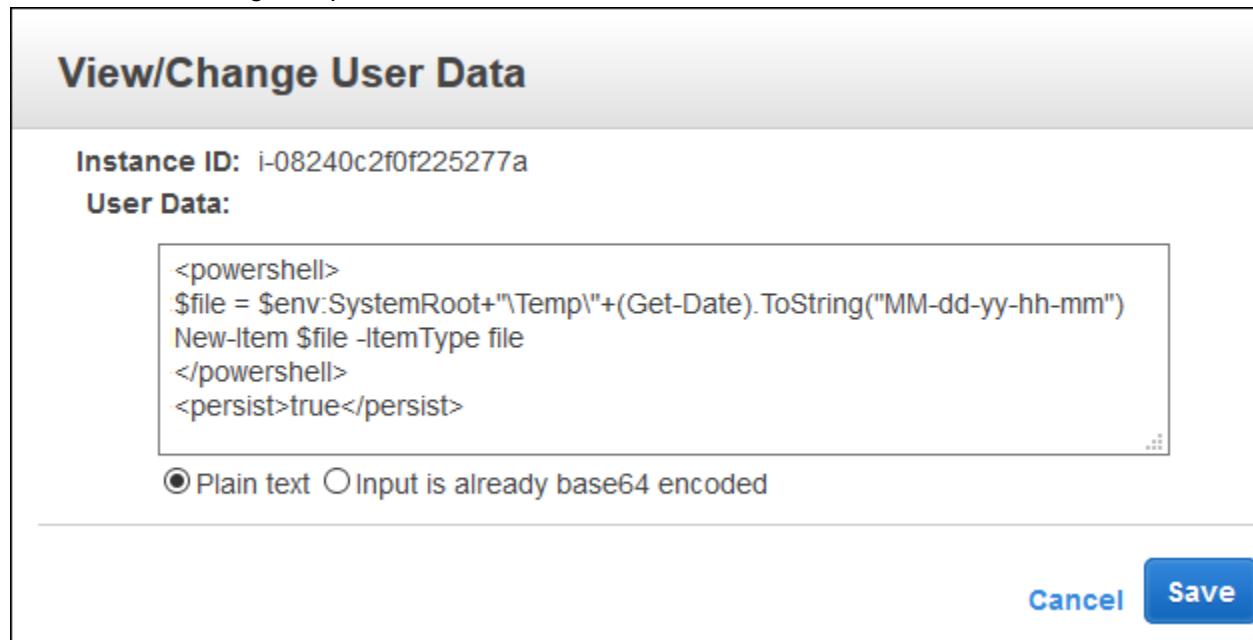
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**. Select the instance.
3. Select the instance and choose **Actions, Instance State, Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

4. When prompted for confirmation, choose **Yes, Stop**. It can take a few minutes for the instance to stop.
5. With the instance still selected, choose **Actions, Instance Settings, View/Change User Data**. You can't change the user data if the instance is running, but you can view it.

- In the **View/Change User Data** dialog box, update the user data, and then choose **Save**. To execute user data scripts every time you reboot or start the instance, add `<persist>true</persist>`, as shown in the following example:



- Restart the instance. If you enabled user data execution for subsequent reboots or starts, the updated user data scripts are executed as part of the instance start process.

User Data and the Tools for Windows PowerShell

You can use the Tools for Windows PowerShell 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 \(p. 456\)](#). For information about user data and the AWS CLI, see [User Data and the AWS CLI](#) in the *Amazon EC2 User Guide for Linux Instances*.

Example: Specify Instance User Data at Launch

Create a text file with the instance user data. To execute user data scripts every time you reboot or start the instance, add `<persist>true</persist>`, as shown in the following example:

```
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

To specify instance user data when you launch your instance, use the `New-EC2Instance` command. This command does not perform base64 encoding of the user data for you. Use the following commands to encode the user data in a text file named `script.txt`.

```
PS C:\> $Script = Get-Content -Raw script.txt
PS C:\> $UserData =
[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Script))
```

Use the `-UserData` parameter to pass the user data to the `New-EC2Instance` command.

```
PS C:\> New-EC2Instance -ImageId ami-abcd1234 -MinCount 1 -MaxCount 1 -  
InstanceType m3.medium \  
-KeyName my-key-pair -SubnetId subnet-12345678 -SecurityGroupIds sg-1a2b3c4d \  
-UserData $UserData
```

Example: Update Instance User Data for a Stopped Instance

You can modify the user data of a stopped instance using the [Edit-EC2InstanceAttribute](#) command.

Create a text file with the new script. Use the following commands to encode the user data in the text file named new-script.txt.

```
PS C:\> $NewScript = Get-Content -Raw new-script.txt  
PS C:\> $NewUserData =  
[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($NewScript))
```

Use the `-UserData` and `-Value` parameters to specify the user data.

```
PS C:\> Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -Attribute userData -  
Value $NewUserData
```

Example: View Instance User Data

To retrieve the user data for an instance, use the [Get-EC2InstanceAttribute](#) command.

```
PS C:\> (Get-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -Attribute  
userData).UserData
```

The following is example output. Note that the user data is encoded.

```
PHBvd2Vyc2h1bGw  
+DQpSZW5hbWUtQ29tcHV0ZXIgLU5ld05hbWUgdXNlcj1kYXRhLXRlc3QNCjwvcG93ZXJzaGVsbD4=
```

Use the following commands to store the encoded user data in a variable and then decode it.

```
PS C:\> $UserData_encoded = (Get-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -  
Attribute userData).UserData  
PS C:  
\> [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($UserData_encoded))
```

The following is example output.

```
<powershell>  
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")  
New-Item $file -ItemType file  
</powershell>  
<persist>true</persist>
```

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 \(p. 456\)](#).

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 \(p. 461\)](#).

Contents

- [Retrieving Instance Metadata \(p. 453\)](#)
- [Working with Instance User Data \(p. 455\)](#)
- [Retrieving Dynamic Data \(p. 456\)](#)
- [Instance Metadata Categories \(p. 456\)](#)
- [Instance Identity Documents \(p. 462\)](#)

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 PowerShell cmdlets to retrieve the URI. For example, if you are running version 3.0 or later of PowerShell, use the following cmdlet:

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/
```

If you don't want to use PowerShell, you can install a third-party tool such as GNU Wget or cURL.

Important

If you do install a third-party tool on a Windows instance, ensure that you read the accompanying documentation carefully, as the method of calling the HTTP and the output format might be different from what is documented here.

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.

```
PS C:\> Invoke-RestMethod -uri 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. For more information, see [Instance Metadata Categories \(p. 456\)](#).

```
PS C:\> Invoke-RestMethod -uri 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.

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/ami-id
```

```
ami-12345678
```

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/reservation-id  
r-fea54097
```

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/local-hostname  
ip-10-251-50-12.ec2.internal
```

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-hostname  
ec2-203-0-113-25.compute-1.amazonaws.com
```

This example gets the subnet ID for an instance.

```
PS C:\> Invoke-RestMethod -uri 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 automatically when you start the instance. However, you can configure settings so that updated user data scripts are executed one time when you start the instance or every time you reboot or start the instance.

Specify Instance User Data at Launch

You can specify user data when you launch an instance. You can specify that the user data is executed one time at launch, or every time you reboot or start the instance. For more information, see [Running Commands on Your Windows Instance at Launch \(p. 447\)](#).

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 URI:

```
http://169.254.169.254/latest/user-data
```

A request for user data returns the data as it is (content type `application/octet-stream`).

This example returns user data that was provided as comma-separated text:

```
PS C:\> Invoke-RestMethod -uri 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:

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/user-data
<powershell>
$file = $env:SystemRoot + "\Temp\" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

To retrieve user data for an instance from your own computer, see [User Data and the Tools for Windows PowerShell \(p. 451\)](#)

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:

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/dynamic/instance-identity/
document
rsa2048
pkcs7
signature
```

For more information about dynamic data and examples of how to retrieve it, see [Instance Identity Documents \(p. 462\)](#).

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
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 non-NVMe instance store volumes, if any are present. The <i>N</i> indicates the index of each 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
events/maintenance	Reserved for future use.	2018-08-17
hostname	The private IPv4 DNS hostname of the instance. In cases where multiple	1.0

Data	Description	Version Introduced
	network interfaces are present, this refers to the eth0 device (the device for which the device number is 0).	
iam/info	If there is an IAM role associated with the instance, contains information about the last time the instance profile was updated, including the instance's LastUpdated date, InstanceProfileArn, and InstanceProfileId. Otherwise, not present.	2012-01-12
iam/security-credentials/ 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. 634)). Otherwise, not present.	2012-01-12
identity-credentials/ec2/ info	[Reserved for internal use only] Information about the credentials that AWS uses to identify an instance to the rest of the Amazon EC2 infrastructure.	2018-05-23
identity-credentials/ec2/ security-credentials/ec2- instance	[Reserved for internal use only] The credentials that AWS uses to identify an instance to the rest of the Amazon EC2 infrastructure.	2018-05-23
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. 110) .	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

Data	Description	Version Introduced
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
network/interfaces/macs/mac/device-number	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 DeviceIndex and device-index fields that are used by the Amazon EC2 API and the EC2 commands for the AWS CLI.	2011-01-01
network/interfaces/macs/mac/interface-id	The ID of the network interface.	2011-01-01
network/interfaces/macs/mac/ipv4-associations/public-ip	The private IPv4 addresses that are associated with each public IP address and assigned to that interface.	2011-01-01
network/interfaces/macs/mac/ipv6s	The IPv6 addresses associated with the interface. Returned only for instances launched into a VPC.	2016-06-30
network/interfaces/macs/mac/local-hostname	The interface's local hostname.	2011-01-01
network/interfaces/macs/mac/local-ipv4s	The private IPv4 addresses associated with the interface.	2011-01-01
network/interfaces/macs/mac/mac	The instance's MAC address.	2011-01-01
network/interfaces/macs/mac/owner-id	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

Data	Description	Version Introduced
network/interfaces/macs/mac/public-hostname	The interface's public DNS (IPv4). This category is only returned if the enableDnsHostnames attribute is set to true. For more information, see Using DNS with Your VPC .	2011-01-01
network/interfaces/macs/mac/public-ipv4s	The Elastic IP addresses associated with the interface. There may be multiple IPv4 addresses on an instance.	2011-01-01
network/interfaces/macs/mac/security-groups	Security groups to which the network interface belongs.	2011-01-01
network/interfaces/macs/mac/security-group-ids	The IDs of the security groups to which the network interface belongs.	2011-01-01
network/interfaces/macs/mac/subnet-id	The ID of the subnet in which the interface resides.	2011-01-01
network/interfaces/macs/mac/subnet-ipv4-cidr-block	The IPv4 CIDR block of the subnet in which the interface resides.	2011-01-01
network/interfaces/macs/mac/subnet-ipv6-cidr-blocks	The IPv6 CIDR block of the subnet in which the interface resides.	2016-06-30
network/interfaces/macs/mac/vpc-id	The ID of the VPC in which the interface resides.	2011-01-01
network/interfaces/macs/mac/vpc-ipv4-cidr-block	The primary IPv4 CIDR block of the VPC.	2011-01-01
network/interfaces/macs/mac/vpc-ipv4-cidr-blocks	The IPv4 CIDR blocks for the VPC.	2016-06-30
network/interfaces/macs/mac/vpc-ipv6-cidr-blocks	The IPv6 CIDR block of the VPC in which the interface resides.	2016-06-30
placement/availability-zone	The Availability Zone in which the instance launched.	2008-02-01
product-codes	Marketplace product codes associated with the instance, if any.	2007-03-01
public-hostname	The instance's public DNS. This category is only returned if the enableDnsHostnames attribute is set to true. For more information, see Using DNS with Your VPC in the Amazon VPC User Guide .	2007-01-19
public-ipv4	The public IPv4 address. If an Elastic IP address is associated with the instance, the value returned is the Elastic IP address.	2007-01-19
public-keys/0/openssh-key	Public key. Only available if supplied at instance launch time.	1.0

Data	Description	Version Introduced
<code>ramdisk-id</code>	The ID of the RAM disk specified at launch time, if applicable.	2007-10-10
<code>reservation-id</code>	The ID of the reservation.	1.0
<code>security-groups</code>	<p>The names of the security groups applied to the instance.</p> <p>After launch, you can change the security groups of the instances. Such changes are reflected here and in <code>network/interfaces/mac/<i>mac</i>/security-groups</code>.</p>	1.0
<code>services/domain</code>	The domain for AWS resources for the region.	2014-02-25
<code>services/partition</code>	<p>The partition that the resource is in. For standard AWS regions, the partition is aws. If you have resources in other partitions, the partition is aws-<i>partitionname</i>. For example, the partition for resources in the China (Beijing) region is aws-cn.</p>	2015-10-20
<code>spot/instance-action</code>	<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. For more information, see instance-action (p. 279).</p>	2016-11-15
<code>spot/termination-time</code>	<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, 2015-01-05T18:02:00Z) only if the Spot Instance has been marked for termination by Amazon EC2. The termination-time item is not set to a time if you terminated the Spot Instance yourself. For more information, see termination-time (p. 280).</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. 462) .	2009-04-04
instance-identity/pkcs7	Used to verify the document's authenticity and content against the signature. See Instance Identity Documents (p. 462) .	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. 462) .	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.

The instance identity document is generated when the instance is launched, and exposed to the instance through [instance metadata \(p. 452\)](#). It validates the attributes of the instances, such as the 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:

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/dynamic/instance-identity/document
```

The following is example output:

```
privateIp      : 10.0.2.174
availabilityZone : us-west-2a
devpayProductCodes :
version       : 2010-08-31
instanceId    : i-1234567890abcdef0
billingProducts : {bp-6ba54002}
instanceType   : m3.medium
architecture   : x86_64
accountId     : 123456789012
kernelId      :
ramdiskId     :
imageId       : ami-1562d075
pendingTime    : 2017-03-13T17:13:27Z
region        : us-west-2
```

To retrieve the instance identity signature, use the following command from your running instance:

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/dynamic/instance-identity/  
signature
```

The following is example output:

```
dExamplesjNQhhJan7pORLpLSr71JEF4V2DhKGlyoYVB0UYrY9njyBCmhEayaGrhtS/AWY+LPx  
1VSQURF5n0gwPNCuO6ICT0fNrm5IH7w9ydyaxexamplejW8XvWPxbuRkcN0TAA1p4RtCAqm4ms  
x2oALjWSCBExample=
```

To retrieve the PKCS7 signature, use the following command from your running instance:

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/dynamic/instance-identity/  
pkcs7
```

The following is example output:

```
MIICiTCCAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC  
VVVmxCzAJBgNVBAgTA1dBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6  
b24xFDASBgNVBAsTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXN0Q2lsYWMxHzAd  
BgkqhkiG9w0BCQEWEg5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN  
MTIwNDI0MjA0NTIxWjCzBiDELMAkGA1UEBhMCVVVmxCzAJBgNVBAgTA1dBMRAwDgYD  
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BTSBDb25z  
b2x1MRIwEAYDVQQDEw1UZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEg5vb25lQGFt  
YXpvbi5jb20wgZBwDOYJKoZIhvcNAQEBBQADgY0AMIGJAOGBAMaK0dn+a4GmWIWJ  
21uUSfwfEvySWtC2XADZ4nB+BLYgvIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T  
xDHudUZg3qX4waLG5M43q7Wgc/Mb0QITxOUSQv7c7ugFFDzQGBBzswY6786m86gPE  
Ibb3OhjZnzcvQAArRhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4  
nUhVVxYUnteD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb  
FFBjvSfpJ1J00zbhNYSS5f6GuEDmFJ10ZxBHjJnyp3780D8uTs7fLvjk79LjSTb  
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE
```

Upgrading an Amazon EC2 Windows Instance to a Newer Version of Windows Server

There are two methods to upgrade an earlier version of Windows Server running on an instance: in-place upgrade and migration (also called side-by-side upgrade). An in-place upgrade upgrades the operating system files while your personal settings and files are intact. A migration involves capturing settings, configurations, and data and porting these to a newer operating system on a fresh Amazon EC2 instance.

Microsoft has traditionally recommended migrating to a newer version of Windows Server instead of upgrading. Migrating can result in fewer upgrade errors or issues, but can take longer than an in-place upgrade because of the need to provision a new instance, plan for and port applications, and adjust configurations settings on the new instance. An in-place upgrade can be faster, but software incompatibilities can produce errors.

Contents

- [Performing a Server Migration \(p. 464\)](#)
- [Performing an In-Place Upgrade \(p. 464\)](#)
- [Migrating to Latest Generation Instance Types \(p. 468\)](#)
- [Troubleshooting an Upgrade \(p. 473\)](#)

Performing a Server Migration

Migrating involves capturing settings, configurations, and data and porting these to a newer operating system on separate hardware. After validation, the migrated system can be promoted to production. You can migrate instances by launching a new instance from an AMI of the new operating system. You can streamline the process further by using [AWS CloudFormation](#) and [Amazon EC2 Systems Manager](#) to automatically apply settings and configurations to the new system with little manual work.

To migrate your server

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **AMIs, Owned by me**, and **Public images**.
3. For **Search**, add the following filters and press Enter. The **Search** values are case-sensitive.
 - **Owner:** Amazon images
 - **AMI Name:** Windows_Server-2008, Windows_Server-2012, or Windows_Server-2016
4. Launch a new instance from an AMI.
5. Log on to the new instance and install all updates.
6. Perform an application installation and configuration changes.
7. Test the server.
8. When validated, promote the server to production.

Performing an In-Place Upgrade

Before you perform an in-place upgrade, you must determine which network drivers the instance is running. PV network drivers enable you to access your instance using Remote Desktop. Starting with Windows Server 2008 R2, instances use either AWS PV, Intel Network Adapter, or the Enhanced Networking drivers. Instances with Windows Server 2003 and Windows Server 2008 use *Citrix PV* drivers. For more information, see [Paravirtual Drivers for Windows Instances \(p. 409\)](#).

Before You Begin an In-Place Upgrade

Complete the following tasks and note the following important details before you begin your in-place upgrade.

- Read the Microsoft documentation to understand the upgrade requirements, known issues, and restrictions. Also review the official instructions for upgrading.
 - [Upgrading to Windows Server 2008](#)
 - [Upgrading to Windows Server 2008 R2](#)
 - [Upgrade Options for Windows Server 2012](#)
 - [Upgrade Options for Windows Server 2012 R2](#)
 - [Upgrade and conversion options for Windows Server 2016](#)
- We do not recommend performing an operating system upgrade on a T1 or T2 instance type. These types of instances might not have enough resources to manage the upgrade process. To upgrade one of these instances, you must resize the instance to another instance type, perform the upgrade, and then resize it back to a T1 or T2 instance type. For more information, see [Changing the Instance Type \(p. 182\)](#).
- Verify that the root volume on your Windows instance has enough free disk space. The Windows Setup process might not warn you of insufficient disk space. For information about how much disk space is required to upgrade a specific operating system, see the Microsoft documentation. If the volume does

not have enough space, it can be expanded. For more information, see [Modifying the Size, IOPS, or Type of an EBS Volume on Windows \(p. 773\)](#).

- Determine your upgrade path. You must upgrade the operating system to the same architecture. For example, you must upgrade a 32-bit system to a 32-bit system. Windows Server 2008 R2 and later are 64-bit only.
- Disable antivirus and anti-spyware software and firewalls. These types of software can conflict with the upgrade process. Re-enable antivirus and anti-spyware software and firewalls after the upgrade completes.
- The Upgrade Helper Service only supports instances running Citrix PV drivers. If the instance is running Red Hat drivers, you must manually [upgrade those drivers \(p. 414\)](#) first.

Upgrade an Instance In-Place with AWS PV, Intel Network Adapter, or the Enhanced Networking Drivers

Use the following procedure to upgrade a Windows Server instance using the AWS PV, Intel Network Adapter, or the Enhanced Networking network drivers.

To perform the in-place upgrade

1. Create an AMI of the system you plan to upgrade for either backup or testing purposes. You can then perform the upgrade on the copy to simulate a test environment. If the upgrade completes, you can switch traffic to this instance with little downtime. If the upgrade fails, you can revert to the backup. For more information, see [Creating a Custom Windows AMI \(p. 65\)](#).
2. Ensure that your Windows Server instance is using the latest network drivers. See [Upgrading PV Drivers on Your Windows Instances \(p. 414\)](#) for information on upgrading your AWS PV driver.
3. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
4. In the navigation pane, choose **Instances**. Locate the instance. Make a note of the instance ID and Availability Zone for the instance. You need this information later in this procedure.
5. If you are upgrading from Windows Server 2012 or 2012 R2 to Windows Server 2016, do the following on your instance before proceeding:
 - a. Uninstall the EC2Config service. For more information, see [Stopping, Restarting, Deleting, or Uninstalling EC2Config \(p. 385\)](#).
 - b. Install the EC2Launch service. For more information, see [Installing the Latest Version of EC2Launch \(p. 375\)](#).
 - c. Install the Amazon SSM Agent. For more information, see [Installing SSM Agent](#).
6. Create a new volume from a Windows Server installation media snapshot.
 - a. In the navigation pane, choose **Snapshots, Public Snapshots**.
 - b. Add the **Owner** filter and choose **Amazon images**.
 - c. Add the **Description** filter and type **Windows**. Press Enter.
 - d. Select the snapshot that matches the system architecture and language preference you are upgrading to. For example, select **Windows 2016 English Installation Media** to upgrade to Windows Server 2016.
 - e. Choose **Actions, Create Volume**.
 - f. In the **Create Volume** dialog box, choose the Availability Zone that matches your Windows instance, and choose **Create**.
7. In the **Volume Successfully Created** message, choose the volume that you just created.
8. Choose **Actions, Attach Volume**.
9. In the **Attach Volume** dialog box, type the instance ID and choose **Attach**.

10. Begin the upgrade by using Windows PowerShell to open the installation media volume you attached to the instance.

- a. If you are upgrading to Windows Server 2016, run the following:

```
./setup.exe /auto upgrade
```

If you are upgrading to an earlier version of Windows Server, run the following:

```
Sources/setup.exe
```

- b. For **Select the operating system you want to install**, select the full installation SKU for your Windows Server instance, and choose **Next**.
- c. For **Which type of installation do you want?**, choose **Upgrade**.
- d. Complete the wizard.

Windows Server Setup copies and processes files. After several minutes, your Remote Desktop session closes. The time it takes to upgrade depends on the number of applications and server roles running on your Windows Server instance. The upgrade process could take as little as 40 minutes or several hours. The instance fails status check 1 of 2 during the upgrade process. When the upgrade completes, both status checks pass. You can check the system log for console output or use Amazon CloudWatch metrics for disk and CPU activity to determine whether the upgrade is progressing.

Note

If upgrading to Windows Server 2016, after the upgrade is complete you can change the desktop background manually to remove the previous operating system name if desired.

If the instance has not passed both status checks after several hours, see [Troubleshooting an Upgrade \(p. 473\)](#).

Upgrade an Instance In-Place with Citrix PV Drivers

Citrix PV drivers are used in Windows Server 2003 and 2008. There is a known issue during the upgrade process where Windows Setup removes portions of the Citrix PV drivers that enable you to connect to the instance by using Remote Desktop. To avoid this problem, the following procedure describes how to use the Upgrade Helper Service during your in-place upgrade.

Using the Upgrade Helper Service

You must run the Upgrade Helper Service before you start the upgrade. After you run it, the utility creates a Windows service that executes during the post-upgrade steps to correct the driver state. The executable is written in C# and can run on .NET Framework versions 2.0 through 4.0.

When you run Upgrade Helper Service on the system *before* the upgrade, it performs the following tasks:

- Creates a new Windows service named `UpgradeHelperService`.
- Verifies that the Citrix PV drivers are installed.
- Checks for unsigned boot critical drivers and presents a warning if any are found. Unsigned boot critical drivers could cause system failure after the upgrade if the drivers are not compatible with the newer Windows Server version.

When you run Upgrade Helper Service on the system *after* the upgrade, it performs the following tasks:

- Enables the `RealTimeIsUniversal` registry key for the correct time synchronization.
- Restores the missing PV driver by executing the following command:

```
pnputil -i -a "C:\Program Files (x86)\Citrix\XenTools\*.inf"
```

- Installs the missing device by executing the following command:

```
C:\Temp\EC2DriverUtils.exe install "C:\Program Files (x86)\Citrix\XenTools\xevtchn.inf"  
ROOT\XENEVTCHN
```

- Automatically removes UpgradeHelperService when complete.

Performing the Upgrade on Instances Running Citrix PV Drivers

To complete the upgrade, you must attach the installation media volume to your EC2 instance and use `UpgradeHelperService.exe`.

To upgrade a Windows Server instance running Citrix PV drivers

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and locate the instance. Make a note of the instance ID and Availability Zone for the instance. You need this information later in this procedure.
3. Create a new volume from a Windows Server installation media snapshot.
 - a. In the navigation pane, choose **Snapshots, Public Snapshots**.
 - b. Add the **Owner** filter and choose **Amazon images**.
 - c. Add the **Description** filter and type **Windows**. Press Enter.
 - d. Select the snapshot that matches the system architecture of your instance. For example, **Windows 2008 64-bit Installation Media**.
 - e. Choose **Actions, Create Volume**.
 - f. In the **Create Volume** dialog box, select the Availability Zone that matches your Windows instance, and choose **Create**.
4. In the **Volume Successfully Created** dialog box, choose the volume that you just created.
5. Choose **Actions, Attach Volume**.
6. In the **Attach Volume** dialog box, type the instance ID and choose **Attach**.
7. On your Windows instance, on the C:\ drive, create a folder named temp.

Important

This folder must be available in the same location after the upgrade. Creating the folder in a Windows system folder or a user profile folder, such as the desktop, can cause the upgrade to fail.

8. Download [OSUpgrade.zip](#) and extract the files into the C:\temp folder.
9. Run C:\temp\UpgradeHelperService.exe and review the C:\temp\Log.txt file for any warnings.
10. Use [Knowledge Base article 950376](#) from Microsoft to uninstall PowerShell from a Windows 2003 instance.
11. Begin the upgrade by using Windows Explorer to open the installation media volume that you attached to the instance.
12. Run the Sources\Setup.exe file.
13. For **Select the operating system you want to install**, select the full installation SKU for your Windows Server instance, and then choose **Next**.
14. For **Which type of installation do you want?**, choose **Upgrade**.
15. Complete the wizard.

Windows Server Setup copies and processes files. After several minutes, your Remote Desktop session closes. The time it takes to upgrade depends on the number of applications and server roles running on your Windows Server instance. The upgrade process could take as little as 40 minutes or several hours. The instance fails status check 1 of 2 during the upgrade process. When the upgrade completes, both status checks pass. You can check the system log for console output or use Amazon CloudWatch metrics for disk and CPU activity to determine whether the upgrade is progressing.

Post Upgrade Tasks

1. Log in to the instance to initiate an upgrade for the .NET Framework and reboot the system when prompted.
2. Install the latest version of the EC2Config service. For more information, see [Installing the Latest Version of EC2Config \(p. 384\)](#).
3. Install Microsoft hotfix [KB2800213](#).
4. Install Microsoft hotfix [KB2922223](#).
5. If you upgraded to Windows Server 2012 R2, we recommend that you upgrade the PV drivers to AWS PV drivers. For more information, see [Windows Server 2012 R2](#).
6. Re-enable antivirus and anti-spyware software and firewalls.

Migrating to Latest Generation Instance Types

The AWS Windows AMIs are configured with the default settings used by the Microsoft installation media with some customizations, including drivers and configurations, that support the latest generation instance types. However, if you are launching instances from custom Windows AMIs or from Windows AMIs provided by Amazon created before August 2018, we recommend that you follow the steps in this topic when migrating to the latest generation of EC2 instances, Nitro instances, including bare metal instances.

For more information, see [Amazon EC2 Update — Additional Instance Types, Nitro System, and CPU Options](#).

Contents

- [Part 1: Installing and Upgrading AWS PV Drivers \(p. 469\)](#)
- [Part 2: Installing and upgrading ENA \(p. 469\)](#)
- [Part 3: Upgrading AWS NVMe drivers \(p. 470\)](#)
- [Part 4: Updating EC2Config and EC2Launch \(p. 470\)](#)
- [Part 5: Installing the Serial Port Driver for Bare Metal Instances \(p. 471\)](#)
- [Part 6: Updating Power Management Settings \(p. 471\)](#)
- [\(Alternative\) Upgrading the AWS PV, ENA, and NVMe Drivers Using AWS Systems Manager \(p. 472\)](#)

Note

Alternatively, you can use the `AWSSupport-UpgradeWindowsAWSDrivers` automation document to automate the procedures described in Part 1, Part 2, and Part 3. If you choose to use the automated procedure, see [\(Alternative\) Upgrading the AWS PV, ENA, and NVMe Drivers Using AWS Systems Manager \(p. 472\)](#), and then continue with Part 4 and Part 5.

Before you begin

This procedure assumes that you are currently running on a previous generation Xen-based instance type, such as an M4 or C4, and you are migrating to a latest generation instance type, such as an M5 or C5.

Note

When migrating to the latest generation instances, the static IP or custom DNS network settings on the existing ENI may be lost as the instance will default to a new Enhanced Networking Adapter device.

Before following the steps in this procedure, we recommend that you create a backup of the instance. From the [EC2 console](#), choose the instance that requires the migration, open the context (right-click) menu, and choose **Instance State, Stop**.

Warning

When you stop an instance, the data on any instance store volumes is erased. To preserve data on instance store volumes, ensure that you back up the data to persistent storage.

Open the context (right-click) menu for the instance in the [EC2 console](#), choose **Image**, and then choose **Create Image**.

Note

Parts 4 and 5 of these instructions can be completed after you migrate or change the instance type to the latest generation, such as M5 or C5. However, we recommend that you complete them before you migrate if you are migrating specifically to an EC2 Bare Metal instance type.

Part 1: Installing and Upgrading AWS PV Drivers

Though AWS PV drivers are not used in the Nitro system, you should still upgrade them if you are on previous versions of either Citrix PV or AWS PV. The latest AWS PV drivers resolve bugs in previous versions of the drivers that may appear while you are on a Nitro system, or if you need to migrate back to a Xen-based instance. As a best practice, we recommend always updating to the latest drivers for Windows instances on AWS.

Use the following procedure to perform an in-place upgrade of AWS PV drivers, or to upgrade from Citrix PV drivers to AWS PV drivers on Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, or Windows Server 2016. For more information, see [Upgrading PV Drivers on Your Windows Instances \(p. 414\)](#).

To perform an upgrade of or to AWS PV drivers

1. Connect to the instance using Remote Desktop and prepare the instance for upgrade. Take all non-system disks offline before you perform the upgrade. If you are performing an in-place update of AWS PV drivers, this step is not required. Set non-essential services to **Manual** start-up in the Services console.
2. [Download](#) the latest driver package to the instance.
3. Extract the contents of the folder and run `AWSPVDriverSetup.msi`.

After running the MSI, the instance automatically reboots and upgrades the driver. The instance may not be available for up to 15 minutes.

After the upgrade is complete and the instance passes both health checks in the Amazon EC2 console, connect to the instance using Remote Desktop and verify that the new driver was installed. In Device Manager, under **Storage Controllers**, locate **AWS PV Storage Host Adapter**. Verify that the driver version is the same as the latest version listed in the Driver Version History table. For more information, see [AWS PV Driver Version History \(p. 411\)](#).

Part 2: Installing and upgrading ENA

Upgrade to the latest Elastic Network Adapter driver to ensure that all network features are supported. If you launched your instance and it does not have enhanced networking already enabled, you must

download and install the required network adapter driver on your instance. Then, set the `enaSupport` instance attribute to **activate enhanced networking**. You can only enable this attribute on supported instance types and only if the ENA driver is installed. For more information, see [Enabling Enhanced Networking with the Elastic Network Adapter \(ENA\) on Windows Instances \(p. 685\)](#).

1. [Download](#) the latest driver to the instance.
2. Extract the zip archive.
3. Install the driver by running the `install.ps1` PowerShell script from the extracted folder.

Note

To avoid installation errors, run this script as an administrator.

4. Check if your AMI has `enaSupport` enabled. If not, continue by following the documentation at [Enabling Enhanced Networking with the Elastic Network Adapter \(ENA\) on Windows Instances \(p. 685\)](#).

Part 3: Upgrading AWS NVMe drivers

AWS NVMe drivers are used to interact with Amazon EBS and SSD instance store volumes that are exposed as NVMe block devices in the Nitro system for better performance.

Important

The following instructions are modified specifically for when you install or upgrade AWS NVMe on a previous generation instance with the intention to migrate the instance to the latest generation instance type.

1. [Download](#) the latest driver package to the instance.
2. Extract the zip archive.
3. Install the driver by running `install.ps1 -NoReboot`.
4. Open a PowerShell session and run the following command:

```
start rundll32.exe sppnp.dll,Sysprep_Generalize_Pnp -wait
```

This command only runs sysprep on the driver devices. It does not run the full sysprep preparation.

5. For Windows Server 2008 R2 and Windows Server 2012 RTM, shut down the instance, change the instance type to a latest generation instance and start it, then proceed to Part 4. If you start the instance again on a previous generation instance type before migrating to a latest generation instance type, it will not boot. For other supported Windows AMIs, you can change the instance type any time after the device sysprep.

Part 4: Updating EC2Config and EC2Launch

For Windows instances, the latest EC2Config and EC2Launch utilities provide additional functionality and information when running on the Nitro system, including on EC2 Bare Metal. By default, the EC2Config service is included in AMIs prior to Windows Server 2016. EC2Launch replaces EC2Config on Windows Server 2016 AMIs.

When the EC2Config and EC2Launch services are updated, new Windows AMIs from AWS include the latest version of the service. However, you must update your own Windows AMIs and instances with the latest version of EC2Config and EC2Launch.

To install or update EC2Config

1. Download and unzip the [EC2Config Installer](#).

2. Run `EC2Install.exe`. For a complete list of options, run `EC2Install` with the `/?` option. By default, setup displays prompts. To run the command with no prompts, use the `/quiet` option.

For more information, see [Installing the Latest Version of EC2Config \(p. 384\)](#).

To install or update EC2Launch

1. If you have already installed and configured EC2Launch on an instance, make a backup of the EC2Launch configuration file. The installation process does not preserve changes in this file. By default, the file is located in the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` directory.
2. Download [EC2-Windows-Launch.zip](#) to a directory on the instance.
3. Download [install.ps1](#) to the same directory where you downloaded `EC2-Windows-Launch.zip`.
4. Run `install.ps1`.

Note

To avoid installation errors, run the this script as an administrator.

5. If you made a backup of the EC2Launch configuration file, copy it to the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config` directory.

For more information, see [Configuring a Windows Instance Using EC2Launch \(p. 374\)](#).

Part 5: Installing the Serial Port Driver for Bare Metal Instances

The `i3.metal` instance type uses a PCI-based serial device rather than an I/O port-based serial device. The latest Windows AMIs automatically use the PCI-based serial device and have the serial port driver installed. If you are not using an instance launched from an Amazon-provided Windows AMI dated 2018.04.11 or later, you must install the Serial Port Driver to enable the serial device for EC2 features such as Password Generation and Console Output. The latest EC2Config and EC2Launch utilities also support `i3.metal` and provide additional functionality. Follow the steps in Part 4, if you have not yet done so.

To install the serial port driver

1. [Download](#) the serial driver package to the instance.
2. Extract the contents of the folder, open the context (right-click) menu for `aws_ser.INF`, and choose **install**.
3. Choose **Okay**.

Part 6: Updating Power Management Settings

The following update to power management settings will set displays to never turn off, which allows for graceful OS shutdowns on the Nitro system. All Windows AMIs provided by Amazon as of 2018.11.28 already have this default configuration.

1. Open a command prompt or PowerShell session.
2. Run the following commands:

```
powercfg /setacvalueindex 381b4222-f694-41f0-9685-ff5bb260df2e 7516b95f-f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbc2b7e 0
powercfg /setacvalueindex 8c5e7fda-e8bf-4a96-9a85-a6e23a8c635c 7516b95f-f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbc2b7e 0
powercfg /setacvalueindex a1841308-3541-4fab-bc81-f71556f20b4a 7516b95f-f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbc2b7e 0
```

(Alternative) Upgrading the AWS PV, ENA, and NVMe Drivers Using AWS Systems Manager

The **AWSSupport-UpgradeWindowsAWSDrivers** automation document automates the steps described in Part 1, Part 2, and Part 3. This method can also repair an instance where the driver upgrades have failed.

The **AWSSupport-UpgradeWindowsAWSDrivers** automation document upgrades or repairs storage and network AWS drivers on the specified EC2 instance. The document attempts to install the latest versions of AWS drivers online by calling the AWS Systems Manager Agent (SSM Agent). If the SSM Agent is not contactable, the document can perform an offline installation of the AWS drivers if explicitly requested.

To automatically upgrade the AWS PV ENA, and NVMe drivers using AWS Systems Manager

1. Open the Systems Manager console at <https://console.aws.amazon.com/systems-manager>.
2. Choose **Automation**, **Execute Automation**.
3. Choose the **AWSSupport-UpgradeWindowsAWSDrivers** automation document and then configure the following options in the **Input Parameters** section:

Instance ID

Enter the unique ID of the instance to upgrade.

AllowOffline

(Optional) Choose one of the following options:

- **True** — Choose this option to perform an offline installation. The instance is stopped and restarted during the upgrade process.

Warning

When you stop an instance, the data on any instance store volumes is erased. To preserve data on instance store volumes, ensure that you back up the data to persistent storage.

- **False** — (Default) To perform an online installation, leave this option selected. The instance is restarted during the upgrade process.

Important

Online and offline upgrades create an AMI before attempting the upgrade operations. The AMI persists after the automation completes. Secure your access to the AMI, or delete it if it is no longer needed.

SubnetId

(Optional) Enter one of the following values:

- **SelectedInstanceSubnet** — (Default) The upgrade process launches the *helper* instance into the same subnet as the instance that is to be upgraded. The subnet must allow communication to the SSM endpoints.
- **CreateNewVPC** — The upgrade process launches the *helper* instance into a new VPC. Use this option if you're not sure whether the target instance's subnet allows communication to the SSM endpoints. Your IAM user must have permission to create a VPC.
- A specific subnet ID — Specify the ID of a specific subnet into which to launch the *helper* instance. The subnet must be in the same Availability Zone as the instance that is to be upgraded, and it must allow communication with the SSM endpoints.

4. Choose **Execute automation**.
5. Allow the upgrade to complete. It could take up to 10 minutes to complete an online upgrade, and up to 25 minutes to complete an offline upgrade.

Troubleshooting an Upgrade

AWS provides upgrade support for issues or problems with the Upgrade Helper Service, an AWS utility that helps you perform in-place upgrades involving Citrix PV drivers.

After the upgrade, the instance might temporarily experience higher than average CPU utilization while the .NET Runtime Optimization service optimizes the .NET framework. This is expected behavior.

If the instance has not passed both status checks after several hours, check the following.

- If you upgraded to Windows Server 2008 and both status checks fail after several hours, the upgrade may have failed and be presenting a prompt to **Click OK** to confirm rolling back. Because the console is not accessible at this state, there is no way to click the button. To get around this, perform a reboot via the Amazon EC2 console or API. The reboot takes ten minutes or more to initiate. The instance might become available after 25 minutes.
- Remove applications or server roles from the server and try again.

If the instance does not pass both status checks after removing applications or server roles from the server, do the following.

- Stop the instance and attach the root volume to another instance. For more information, see the description of how to stop and attach the root volume to another instance in "[Waiting for the metadata service](#)" (p. 993).
- Analyze Windows Setup log files and event logs for failures.

For other issues or problems with an operating system upgrade or migration, we recommend reviewing the articles listed in [Before You Begin an In-Place Upgrade](#) (p. 464).

Identify EC2 Windows Instances

Your application might need to determine whether it is running on an EC2 instance.

For information about identifying Linux instances, see [Identify EC2 Linux Instances](#) in the *Amazon EC2 User Guide for Linux Instances*.

Inspecting the Instance Identity Document

For a definitive and cryptographically verified method of identifying an EC2 instance, check the instance identity document, including its signature. These documents are available on every EC2 instance at the local, non-routable address <http://169.254.169.254/latest/dynamic/instance-identity/>. For more information, see [Instance Identity Documents](#) (p. 462).

Inspecting the System UUID

You can get the system UUID and look for the presence of the characters "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, EC2 instances using SMBIOS 2.4 might represent the UUID in little-endian format, therefore the "EC2" characters do not appear at the beginning of the UUID.

Example : Get the UUID using WMI or Windows PowerShell

Use the Windows Management Instrumentation command line (WMIC) as follows:

```
wmic path win32_computersystemproduct get uuid
```

Alternatively, if you're using Windows PowerShell, use the **Get-WmiObject** cmdlet as follows:

```
PS C:\> Get-WmiObject -query "select uuid from Win32_ComputerSystemProduct" | Select UUID
```

In the following example output, the UUID starts with "EC2", which indicates that the system is probably an EC2 instance.

```
EC2AE145-D1DC-13B2-94ED-01234ABCDEF
```

For instances using SMBIOS 2.4, the UUID might be represented in little-endian format; for example:

```
45E12AEC-DCD1-B213-94ED-01234ABCDEF
```

Amazon Elastic Graphics

Amazon Elastic Graphics provides flexible, low-cost, and high performance graphics acceleration for your Windows instances. Elastic Graphics accelerators come in multiple sizes and are a low-cost alternative to using GPU graphics instance types (such as G2 and G3). You have the flexibility to choose an instance type that meets the compute, memory, and storage needs of your application. Then, choose the accelerator for your instance that meets the graphics requirements of your workload.

Elastic Graphics is suited for applications that require a small or intermittent amount of additional graphics acceleration, and that use OpenGL graphics support. If you need access to full, directly attached GPUs and use of DirectX, CUDA, or Open Computing Language (OpenCL) parallel computing frameworks, use an accelerated computing instance type instance instead. For more information, see [Windows Accelerated Computing Instances \(p. 163\)](#).

Contents

- [Elastic Graphics Basics \(p. 475\)](#)
- [Pricing for Elastic Graphics \(p. 477\)](#)
- [Elastic Graphics Limitations \(p. 477\)](#)
- [Working with Elastic Graphics \(p. 477\)](#)
- [Using CloudWatch Metrics to Monitor Elastic Graphics \(p. 482\)](#)
- [Troubleshooting \(p. 484\)](#)

Elastic Graphics Basics

To use Elastic Graphics, launch a Windows instance and specify an accelerator type for the instance during launch. AWS finds available Elastic Graphics capacity and establishes a network connection between your instance and the Elastic Graphics accelerator.

The following instance types support Elastic Graphics accelerators:

- C3 | C4 | C5 | C5d
- D2
- H1
- I3 *
- M3 | M4 | M5 | M5d
- P2 | P3
- R3 | R4 | R5 | R5d
- t2.medium or larger | t3.medium or larger
- X1 | X1e
- z1d

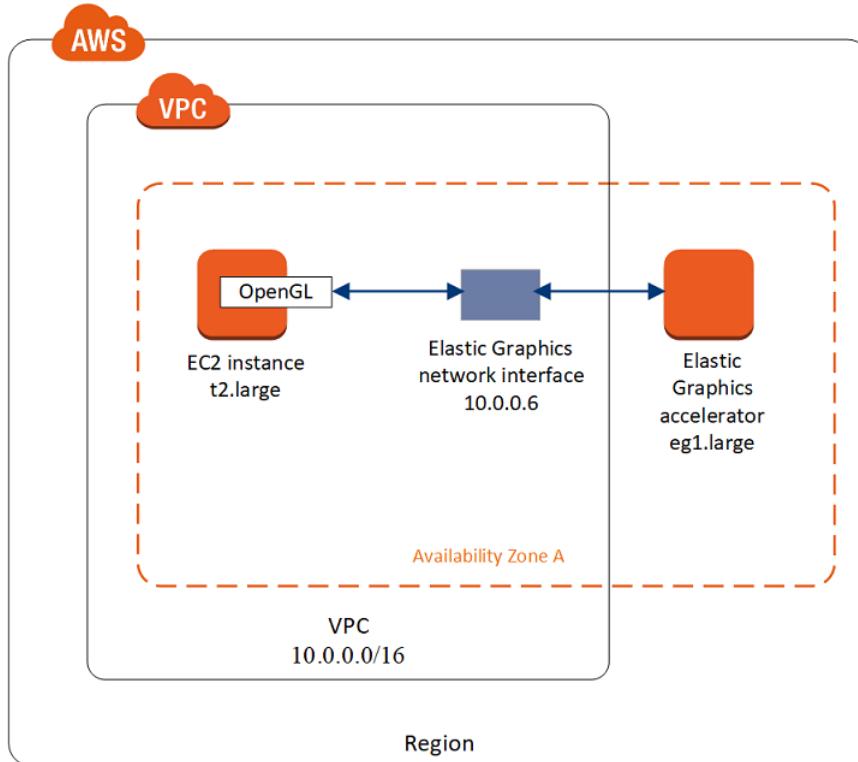
* Bare metal instances are not supported

The following Elastic Graphics accelerators are available. You can attach any Elastic Graphics accelerator to any supported instance type.

Elastic Graphics accelerator	Graphics memory (GB)
eg1.medium	1
eg1.large	2
eg1.xlarge	4
eg1.2xlarge	8

An Elastic Graphics accelerator does not form part of the hardware of your instance. Instead, it is network-attached through a network interface, known as the *Elastic Graphics network interface*. When you launch an instance with graphics acceleration, the Elastic Graphics network interface is created in your VPC for you.

The Elastic Graphics network interface is created in the same subnet and VPC as your instance and is assigned a private IPv4 address from that subnet. The accelerator attached to your Amazon EC2 instance is allocated from a pool of available accelerators in the same Availability Zone as your instance.



Elastic Graphics accelerators support the API standards for OpenGL 4.3 API and earlier, which can be used for batch applications or 3D-graphics acceleration. An Amazon-optimized OpenGL library on your instance detects the attached accelerator. It directs OpenGL API calls from your instance to the accelerator, which then processes the requests and returns the results. Traffic between the instance and the accelerator uses the same bandwidth as the instance's network traffic so we recommend that you have adequate network bandwidth available. Consult your software vendor for any OpenGL compliance and version questions.

By default, the default security group for your VPC is associated with the Elastic Graphics network interface. The Elastic Graphics network traffic uses the TCP protocol and port 2007. Ensure that the security group for your instance allows for this. For more information, see [Configuring Your Security Groups \(p. 477\)](#).

Pricing for Elastic Graphics

You are charged for each second that an Elastic Graphics accelerator is attached to an instance in the running state when the accelerator is in the Ok state. You are not charged for an accelerator attached to an instance that is in the pending, stopping, stopped, shutting-down, or terminated state. You are also not charged when an accelerator is in the Unknown or Impaired state.

Pricing for accelerators is available at On-Demand rates only. You can attach an accelerator to a Reserved, Scheduled, or Spot Instance. The On-Demand price for the accelerator applies in all cases.

For more information, see [Amazon Elastic Graphics Pricing](#).

Elastic Graphics Limitations

Before you start using Elastic Graphics accelerators, be aware of the following limitations:

- You can attach accelerators only to Windows instances. Linux instances are not supported.
- You can attach one accelerator to an instance at a time, and only during instance launch.
- You can't share an accelerator between instances.
- You can't detach an accelerator from an instance or transfer it to another instance. If you no longer require an accelerator, you must terminate your instance. To change the accelerator type, create an AMI from your instance, terminate the instance, and launch a new instance with a different accelerator specification.
- The only supported versions of the OpenGL API are 4.3 and earlier. DirectX, CUDA, and OpenCL are not supported.
- The Elastic Graphics accelerator is not visible or accessible through the device manager of your instance.
- You can't reserve or schedule accelerator capacity.
- You can't attach accelerators to instances in EC2-Classic.

Working with Elastic Graphics

You can launch an instance and associate it with an Elastic Graphics accelerator during launch. You must then manually install the necessary libraries on your instance that enable communication with the accelerator.

Tasks

- [Configuring Your Security Groups \(p. 477\)](#)
- [Launching an Instance with an Elastic Graphics accelerator \(p. 478\)](#)
- [Installing the Required Software for Elastic Graphics \(p. 479\)](#)
- [Verifying Elastic Graphics Functionality on Your Instance \(p. 479\)](#)
- [Viewing Elastic Graphics Information \(p. 481\)](#)
- [Submitting Feedback \(p. 482\)](#)

Configuring Your Security Groups

If you use the Amazon EC2 console to launch your instance with an Elastic Graphics accelerator and create a security group for you, the console adds the inbound and outbound rules that are required to

allow traffic on the Elastic Graphics port. If you are launching your instance using the AWS CLI or an SDK, you must ensure that your security group allows traffic on the Elastic Graphics port.

To create a security group for Elastic Graphics

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**, **Create Security Group**.
3. Provide a name for your security group, such as "Elastic Graphics security group", and a description for the security group. Select the VPC that you will use to launch your instance with an Elastic Graphics accelerator.
4. Create an inbound security group rule as follows:
 - a. On the **Inbound** tab, choose **Add Rule**
 - b. For **Type**, choose **Elastic Graphics**. For **Source**, choose **Custom** and type the ID of the security group.
5. Create an outbound security group rule as follows:
 - a. On the **Outbound** tab, choose **Add Rule**
 - b. For **Type**, choose **All TCP**. For **Destination**, choose **Custom** and type the ID of the security group.
6. Choose **Create**.

For more information, see [Amazon EC2 Security Groups for Windows Instances \(p. 548\)](#).

Launching an Instance with an Elastic Graphics accelerator

You can associate an Elastic Graphics accelerator to an instance during launch. If the launch fails, the following are possible reasons:

- Insufficient Elastic Graphics accelerator capacity
- Exceeded limit on Elastic Graphics accelerators in the Region
- Not enough private IPv4 addresses in your VPC to create a network interface for the accelerator

For more information, see [Elastic Graphics Limitations \(p. 477\)](#).

To associate an Elastic Graphics accelerator during instance launch (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the dashboard, choose **Launch Instance**.
3. Select a Windows AMI and a supported instance type. For more information, see [Elastic Graphics Basics \(p. 475\)](#).
4. On the **Configure Instance Details** page, select a VPC and subnet in which to launch your instance.
5. Choose **Add Graphics Acceleration**, and select an Elastic Graphics accelerator type.
6. (Optional) On the **Add Storage** and **Add Tags** pages, add volumes and tags as needed.
7. On the **Configure Security Group** page, you can let the console create a security group for you with the required inbound and outbound rules, or you can use the security group that you created manually in [Configuring Your Security Groups \(p. 477\)](#). Add additional security groups as needed.
8. Choose **Review and Launch** to review your instance options and then choose **Launch**.

To associate an Elastic Graphics accelerator during instance launch (AWS CLI)

You can use the [run-instances](#) AWS CLI command with the following parameter:

```
--elastic-gpu-specification Type=eg1.medium
```

For the --security-group-ids parameter, you must include a security group that has the required inbound and outbound rules. For more information, see [Configuring Your Security Groups \(p. 477\)](#).

To associate an Elastic Graphics accelerator during instance launch (Tools for Windows PowerShell)

Use the [New-EC2Instance](#) Tools for Windows PowerShell command.

Installing the Required Software for Elastic Graphics

If you launched your instance using a current AWS Windows AMI, the required software is installed automatically during the first boot. If you launched your instance using Windows AMIs that do not automatically install the required software, you must install the required software on the instance manually.

To install the required software for Elastic Graphics (if necessary)

1. Connect to the instance.
2. Download the [Elastic Graphics installer](#) and open it. The installation manager connects to the Elastic Graphics endpoint and downloads the latest version of the required software.
3. Reboot the instance to verify.

Verifying Elastic Graphics Functionality on Your Instance

The Elastic Graphics packages on your instance include tools that you can use to view the status of the accelerator, and to verify that OpenGL commands from your instance to the accelerator are functional.

If your instance was launched with an AMI that does not have the Elastic Graphics packages pre-installed, you can download and install them yourself. For more information, see [Installing the Required Software for Elastic Graphics \(p. 479\)](#).

Contents

- [Using the Elastic Graphics Status Monitor \(p. 479\)](#)
- [Using the Elastic Graphics Command Line Tool \(p. 480\)](#)

Using the Elastic Graphics Status Monitor

You can use the status monitor tool to view information about the status of an attached Elastic Graphics accelerator. By default, this tool is available in the notification area of the taskbar in your Windows instance and shows the status of the graphics accelerator. The following are the possible values.

Healthy

The Elastic Graphics accelerator is enabled and healthy.

Updating

The status of the Elastic Graphics accelerator is currently updating. It might take a few minutes to display the status.

Out of service

The Elastic Graphics accelerator is out of service. To get more information about the error, choose [Read More](#).

Using the Elastic Graphics Command Line Tool

You can use the Elastic Graphics command line tool, `egcli.exe`, to check the status of the accelerator. If there is a problem with the accelerator, the tool returns an error message.

To launch the tool, open a command prompt from within your instance and run the following command:

```
C:\Program Files\Amazon\EC2ElasticGPUs\manager\egcli.exe
```

The tool also supports the following parameters:

`--json, -j`

Indicates whether to show the JSON message. The possible values are `true` and `false`. The default is `true`.

`--imds, -i`

Indicates whether to check the instance metadata for the availability of the accelerator. The possible values are `true` and `false`. The default is `true`.

The following is example output. A status of `OK` indicates that the accelerator is enabled and healthy.

```
EG Infrastructure is available.  
Instance ID egpu-f6d94dfa66df4883b284e96db7397ee6  
Instance Type eg1.large  
EG Version 1.0.0.885 (Manager) / 1.0.0.95 (OpenGL Library) / 1.0.0.69 (OpenGL Redirector)  
EG Status: Healthy  
JSON Message:  
{  
    "version": "2016-11-30",  
    "status": "OK"  
}
```

The following are the possible values for `status`:

`OK`

The Elastic Graphics accelerator is enabled and healthy.

`UPDATING`

The Elastic Graphics driver is being updated.

`NEEDS_REBOOT`

The Elastic Graphics driver has been updated and a reboot of the Amazon EC2 instance is required.

`LOADING_DRIVER`

The Elastic Graphics driver is being loaded.

`CONNECTING_EGPU`

The Elastic Graphics driver is verifying the connectivity with the Elastic Graphics accelerator.

ERROR_UPDATE_RETRY

An error occurred while updating the Elastic Graphics driver, an update will be retried soon.

ERROR_UPDATE

An unrecoverable error occurred while updating the Elastic Graphics driver.

ERROR_LOAD_DRIVER

An error occurred loading the Elastic Graphics driver.

ERROR_EGPU_CONNECTIVITY

The Elastic Graphics accelerator is unreachable.

Viewing Elastic Graphics Information

You can view information about the Elastic Graphics accelerator attached to your instance.

To view information about an Elastic Graphics accelerator (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. On the **Description** tab, find **Elastic Graphics ID**. Choose the ID to view the following information about the Elastic Graphics accelerator:
 - **Attachment State**
 - **Type**
 - **Health status**

To view information about an Elastic Graphics accelerator (AWS CLI)

You can use the `describe-elastic-gpus` AWS CLI command:

```
aws ec2 describe-elastic-gpus
```

You can use the `describe-network-interfaces` AWS CLI command and filter by owner ID to view information about the Elastic Graphics network interface.

```
aws ec2 describe-network-interfaces --filters "Name=attachment.instance-owner-id,Values=amazon-elasticgpus"
```

To view information about an Elastic Graphics accelerator (Tools for Windows PowerShell)

Use the following commands:

- `Get-EC2ElasticGpu`
- `Get-EC2NetworkInterface`

To view information about an Elastic Graphics accelerator using instance metadata

1. Connect to your Windows instance that is using an Elastic Graphics accelerator.
2. Do one of the following:
 - From PowerShell, use the following cmdlet:

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/elastic-gpus/associations/egpu-f6d94dfa66df4883b284e96db7397ee6
```

- From your web browser, paste the following URL into the address field:

```
http://169.254.169.254/latest/meta-data/elastic-gpus/associations/egpu-f6d94dfa66df4883b284e96db7397ee6
```

Submitting Feedback

You can submit feedback about your experience with Elastic Graphics so the team can make further improvements.

To submit feedback using the Elastic Graphics Status Monitor

- In the notification area of the taskbar in your Windows instance, open the Elastic Graphics Status Monitor.
- In the lower left corner, choose **Feedback**.
- Enter your feedback and choose **Submit**.

Using CloudWatch Metrics to Monitor Elastic Graphics

You can monitor your Elastic Graphics accelerator using Amazon CloudWatch, which collects metrics about your accelerator performance. These statistics are recorded for a period of two weeks, so that you can access historical information and gain a better perspective on how your service is performing.

By default, Elastic Graphics accelerators send metric data to CloudWatch in 5-minute periods.

For more information about Amazon CloudWatch, see the [Amazon CloudWatch User Guide](#).

Elastic Graphics Metrics

The AWS/ElasticGPUs namespace includes the following metrics for Elastic Graphics.

Metric	Description
GPUConnectivityCheckFailed	Reports whether connectivity to the Elastic Graphics accelerator is active or has failed. A value of zero (0) indicates that the connection is active. A value of one (1) indicates a connectivity failure. Units: Count
GPUHealthCheckFailed	Reports whether the Elastic Graphics accelerator has passed a status health check in the last minute. A value of zero (0) indicates that the status check passed. A value of one (1) indicates a status check failure. Units: Count

Metric	Description
GPUMemoryUtilization	The GPU memory used. Units: MiB

Elastic Graphics Dimensions

You can filter the metrics data for your Elastic Graphics accelerators using the following dimensions.

Dimension	Description
EGPUId	Filters the data by the Elastic Graphics accelerator.
InstanceId	Filters the data by the instance to which the Elastic Graphics accelerator is attached.

Viewing CloudWatch Metrics for Elastic Graphics

Metrics are grouped first by the service namespace, and then by the supported dimensions. You can use the following procedures to view the metrics for your Elastic Graphics accelerators.

To view Elastic Graphics metrics using the CloudWatch console

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. If necessary, change the Region. From the navigation bar, select the Region where your Elastic Graphics accelerator resides. For more information, see [Regions and Endpoints](#).
3. In the navigation pane, choose **Metrics**.
4. For **All metrics**, select **Elastic Graphics, Elastic Graphics Metrics**.

To view Elastic Graphics metrics (AWS CLI)

Use the following `list-metrics` command:

```
aws cloudwatch list-metrics --namespace "AWS/ElasticGPUs"
```

Creating CloudWatch Alarms to Monitor Elastic Graphics

You can create a CloudWatch alarm that sends an Amazon SNS message when the alarm changes state. An alarm watches a single metric over a time period you specify, and sends a notification to an Amazon SNS topic based on the value of the metric relative to a given threshold over a number of time periods.

For example, you can create an alarm that monitors the health of an Elastic Graphics accelerator and sends a notification when the graphics accelerator fails a status health check for three consecutive 5-minute periods.

To create an alarm for an Elastic Graphics accelerator health status

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.

2. In the navigation pane, choose **Alarms**, **Create Alarm**.
3. Choose **Select metric**, **Elastic Graphics**, **Elastic Graphics Metrics**.
4. Select the **GPUHealthCheckFailed** metric and choose **Select metric**.
5. Configure the alarm as follows:
 - a. For **Alarm details**, type a name and description for your alarm. For **Whenever**, choose **>=** and type 1.
 - b. For **Actions**, select an existing notification list or choose **New list**.
 - c. Choose **Create Alarm**.

Troubleshooting

The following are common errors and troubleshooting steps.

Contents

- [Investigating Application Performance Issues \(p. 484\)](#)
 - [OpenGL Rendering Performance Issues \(p. 484\)](#)
 - [Remote Access Performance Issues \(p. 485\)](#)
- [Resolving Unhealthy Status Issues \(p. 486\)](#)
 - [Stop and Start the Instance \(p. 486\)](#)
 - [Verify the Installed Components \(p. 486\)](#)
 - [Check the Elastic Graphics Logs \(p. 486\)](#)

Investigating Application Performance Issues

Elastic Graphics uses the instance network to send OpenGL commands to a remotely attached graphics card. In addition, a desktop running an OpenGL application with an Elastic Graphics accelerator is usually accessed using a remote access technology. It is important to distinguish between a performance problem related to the OpenGL rendering or the desktop remote access technology.

OpenGL Rendering Performance Issues

The OpenGL rendering performance is determined by the number of OpenGL commands and frames generated on the remote instance.

Rendering performance may vary depending on the following factors:

- Elastic Graphics accelerator performance
- Network performance
- CPU performance
- Rendering model, scenario complexity
- OpenGL application behavior

An easy way to evaluate performance is to display the number of rendered frames on the remote instance. Elastic Graphics accelerators display a maximum of 25 FPS on the remote instance to achieve the best perceived quality while reducing network usage.

To show the number of frames produced

1. Open the following file in a text editor. If the file does not exist, create it.

```
C:\Program Files\Amazon\EC2ElasticGPUs\conf\eg.conf
```

2. Identify the [Application] section, or add it if it is not present, and add the following configuration parameter:

```
[Application]
show_fps=1
```

3. Restart the application and check the FPS again.

If the FPS reaches 15-25 FPS when updating the rendered scene, then the Elastic Graphics accelerator is performing at peak. Any other performance problems you experience are likely related to the remote access to the instance desktop. If that is the case, see the Remote Access Performance Issues section.

If the FPS number is lower than 15, you can try the following:

- Improve Elastic Graphics accelerator performance by selecting a more powerful graphics accelerator type.
- Improve overall network performance by using these tips:
 - Check the amount of incoming and outgoing bandwidth to and from the Elastic Graphics accelerator endpoint. The Elastic Graphics accelerator endpoint can be retrieved with the following PowerShell command:

```
PS C:\> (Invoke-WebRequest http://169.254.169.254/latest/meta-data/elastic-gpus/
associations/[ELASTICGPU_ID]).content
```

- The network traffic from the instance to the Elastic Graphics accelerator endpoint relates to the volume of commands the OpenGL application is producing.
- The network traffic from the Elastic Graphics accelerator endpoint to the instance relates to the number of frames generated by the graphics accelerator.
- If you see the network usage reaching the instances maximum network throughput, try using an instance with a higher network throughput allowance.
- Improve CPU performance:
 - Applications may require a lot of CPU resources in addition to what the Elastic Graphics accelerator requires. If Windows Task Manager is reporting a high usage of CPU resources, try using an instance with more CPU power.

Remote Access Performance Issues

An instance with an attached Elastic Graphics accelerator can be accessed using different remote access technologies. Performance and quality may vary depending on:

- The remote access technology
- Instance performance
- Client performance
- Network latency and bandwidth between the client and the instance

Possible choices for the remote access protocol include:

- Microsoft Remote Desktop Connection
- NICE DCV
- VNC

For more information about optimization, see the specific protocol.

Resolving Unhealthy Status Issues

If the Elastic Graphics accelerator is in an unhealthy state, use the following troubleshooting steps to resolve the issue.

Stop and Start the Instance

If your Elastic Graphics accelerator is in an unhealthy state, stopping the instance and starting it again is the simplest option. For more information, see [Stopping and Starting Your Instances \(p. 364\)](#).

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

Verify the Installed Components

Open the Windows Control Panel and confirm that the following components are installed:

- Amazon Elastic Graphics Manager
- Amazon Elastic Graphics OpenGL Library
- Amazon EC2 Elastic GPUs OpenGL Redirector

If any of these items are missing, you must install them manually. For more information, see [Installing the Required Software for Elastic Graphics \(p. 479\)](#).

Check the Elastic Graphics Logs

Open the Windows Event Viewer, expand the **Application and Services Logs** section, and search for errors in the following event logs:

- EC2ElasticGPUs
- EC2ElasticGPUs GUI

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. 500)	
Network utilization	NetworkIn (p. 500) NetworkOut (p. 500)	
Disk performance	DiskReadOps (p. 500) DiskWriteOps (p. 500)	
Disk Reads/Writes	DiskReadBytes (p. 500) DiskWriteBytes (p. 500)	
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 \(p. 488\)](#)
- [Manual Monitoring Tools \(p. 489\)](#)

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 \(p. 490\)](#).

- **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 \(p. 490\)](#).

- **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 \(p. 498\)](#).

- **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 \(p. 916\)](#).

Manual Monitoring Tools

Another important part of monitoring Amazon EC2 involves manually monitoring those items that the monitoring scripts, status checks, and CloudWatch alarms don't cover. The Amazon EC2 and CloudWatch console dashboards provide an at-a-glance view of the state of your Amazon EC2 environment.

- Amazon EC2 Dashboard shows:
 - Service Health and Scheduled Events by Region
 - Instance state
 - Status checks
 - Alarm status
 - Instance metric details (In the navigation pane choose **Instances**, select an instance, and choose the **Monitoring** tab)
 - Volume metric details (In the navigation pane choose **Volumes**, select a volume, and choose the **Monitoring** tab)
- Amazon CloudWatch Dashboard shows:
 - Current alarms and status
 - Graphs of alarms and resources
 - Service health status

In addition, you can use CloudWatch to do the following:

- Graph Amazon EC2 monitoring data to troubleshoot issues and discover trends
- Search and browse all your AWS resource metrics
- Create and edit alarms to be notified of problems
- See at-a-glance overviews of your alarms and AWS resources

Best Practices for Monitoring

Use the following best practices for monitoring to help you with your Amazon EC2 monitoring tasks.

- Make monitoring a priority to head off small problems before they become big ones.

- Create and implement a monitoring plan that collects monitoring data from all of the parts in your AWS solution so that you can more easily debug a multi-point failure if one occurs. Your monitoring plan should address, at a minimum, the following questions:
 - What are your goals for monitoring?
 - What resources you will monitor?
 - How often you will monitor these resources?
 - What monitoring tools will you use?
 - Who will perform the monitoring tasks?
 - Who should be notified when something goes wrong?
- Automate monitoring tasks as much as possible.
- Check the log files on your EC2 instances.

Monitoring the Status of Your Instances

You can monitor the status of your instances by viewing status checks and scheduled events for your instances. A status check gives you the information that results from automated checks performed by Amazon EC2. These automated checks detect whether specific issues are affecting your instances. The status check information, together with the data provided by Amazon CloudWatch, gives you detailed operational visibility into each of your instances.

You can also see status 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 \(p. 490\)](#)
- [Scheduled Events for Your Instances \(p. 494\)](#)

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 \(p. 493\)](#).

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 \(p. 373\)](#).

Contents

- [Types of Status Checks \(p. 491\)](#)
- [Viewing Status Checks \(p. 491\)](#)

- [Reporting Instance Status \(p. 492\)](#)
- [Creating and Editing Status Check Alarms \(p. 493\)](#)

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. Amazon EC2 checks the health of the instance by sending an address resolution protocol (ARP) request to the ENI. 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
- Status checks that occur during instance reboot or while a Windows instance store-backed instance is being bundled report an instance status check failure until the instance becomes available again.

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 (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. On the **Instances** page, the **Status Checks** column lists the operational status of each instance.
4. To view the status of a specific instance, select the instance, and then choose the **Status Checks** tab.

The screenshot shows the 'Status Checks' tab selected in the navigation bar. The 'System Status Checks' section indicates a 'System reachability check passed'. The 'Instance Status Checks' section shows a failure with the message 'Instance reachability check failed at October 7, 2013 11:52:11 AM UTC+2 (16 minutes ago)'. A 'Create Status Check Alarm' button is also visible.

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.

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, use the following command:

```
aws ec2 describe-instance-status --filters Name=instance-status.status,Values=impaired
```

To get the status of a single instance, use the following command:

```
aws ec2 describe-instance-status --instance-ids i-1234567890abcdef0
```

Alternatively, use the following commands:

- `Get-EC2InstanceState` (AWS Tools for Windows PowerShell)
- `DescribeInstanceState` (Amazon EC2 Query API)

Reporting Instance Status

You can provide feedback if you are having problems with an instance whose status is not shown as impaired, or you want to send AWS additional details about the problems you are experiencing with an impaired instance.

We use reported feedback to identify issues impacting multiple customers, but do not respond to individual account issues. Providing feedback does not change the status check results that you currently see for the instance.

Reporting Status Feedback Using the Console

To report instance status (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose the **Status Checks** tab, and choose **Submit feedback**.
4. Complete the **Report Instance Status** form, and then choose **Submit**.

Reporting Status Feedback Using the Command Line 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. 491).

To create a status check alarm (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose the **Status Checks** tab, and choose **Create Status Check Alarm**.
4. Select **Send a notification to**. Choose an existing SNS topic, or choose **create topic** to create a new one. If creating a new topic, in **With these recipients**, enter your email address and the addresses of any additional recipients, separated by commas.
5. (Optional) Select **Take the action**, and then select the action that you'd like to take.
6. In **Whenever**, select the status check that you want to be notified about.

Note

If you selected **Recover this instance** in the previous step, select **Status Check Failed (System)**.

7. In **For at least**, set the number of periods you want to evaluate and in **consecutive periods**, select the evaluation period duration before triggering the alarm and sending an email.
8. (Optional) In **Name of alarm**, replace the default name with another name for the alarm.
9. Choose **Create Alarm**.

Important

If you added an email address to the list of recipients or created a new topic, Amazon SNS sends a subscription confirmation email message to each new address. Each recipient must confirm the subscription by choosing the link contained in that message. Alert notifications are sent only to confirmed addresses.

If you need to make changes to an instance status alarm, you can edit it.

To edit a status check alarm (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.

3. Select the instance and choose **Actions, CloudWatch Monitoring, Add/Edit Alarms**.
4. In the **Alarm Details** dialog box, choose the name of the alarm.
5. In the **Edit Alarm** dialog box, make the desired changes, and then choose **Save**.

Creating a Status Check Alarm Using the AWS CLI

In the following example, the alarm publishes a notification to an SNS topic, `arn:aws:sns:us-west-2:111122223333:my-sns-topic`, when the instance fails either the instance check or system status check for at least two consecutive periods. The metric is `StatusCheckFailed`.

To create a status check alarm (AWS CLI)

1. Select an existing SNS topic or create a new one. For more information, see [Using the AWS CLI with Amazon SNS](#) in the *AWS Command Line Interface User Guide*.
2. Use the following [list-metrics](#) command to view the available Amazon CloudWatch metrics for Amazon EC2:

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

3. Use the following [put-metric-alarm](#) command to create the alarm:

```
aws cloudwatch put-metric-alarm --alarm-name StatusCheckFailed-Alarm-for-i-1234567890abcdef0 --metric-name StatusCheckFailed --namespace AWS/EC2 --statistic Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 --unit Count --period 300 --evaluation-periods 2 --threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --alarm-actions arn:aws:sns:us-west-2:111122223333:my-sns-topic
```

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 \(p. 495\)](#)
- [Viewing Scheduled Events \(p. 495\)](#)
- [Working with Instances Scheduled to Stop or Retire \(p. 496\)](#)

- [Working with Instances Scheduled for Reboot \(p. 497\)](#)
- [Working with Instances Scheduled for Maintenance \(p. 498\)](#)

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 (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**. Any resources with an associated event are displayed. You can filter by resource type, or by specific event types. You can select the resource to view details.

Resource Name	Resource Type	Resource Id	Event Type
my-instance	instance	i-c3870335	instance-stop

Event: i-c3870335

Availability Zone	us-west-2a
Event type	instance-stop
Event status	Scheduled
Description	The instance is running on degraded hardware
Start time	May 22, 2015 at 5:00:00 PM UTC-7
End time	

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 select an instance. If the instance has an associated instance stop or instance retirement event, it is displayed in the lower pane.

A Retiring: This instance is scheduled for retirement after May 22, 2015 at 5:00:00 PM UTC-7. ⓘ

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": "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)
- [DescribeInstanceStatus](#) (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 \(p. 366\)](#).

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 \(p. 362\)](#).

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 (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Events**.
3. Choose **Instance resources** from the filter list, 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 (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 \(p. 365\)](#).

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 \(p. 362\)](#).

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 \(p. 499\)](#).

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 \(p. 499\)](#)
- [List the Available CloudWatch Metrics for Your Instances \(p. 500\)](#)
- [Get Statistics for Metrics for Your Instances \(p. 509\)](#)
- [Graph Metrics for Your Instances \(p. 516\)](#)
- [Create a CloudWatch Alarm for an Instance \(p. 516\)](#)
- [Create Alarms That Stop, Terminate, Reboot, or Recover an Instance \(p. 517\)](#)

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.

Monitoring Type	Description
Basic	Data is available automatically in 5-minute periods at no charge.
Detailed	<p>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.</p> <p>For information about pricing, see the Amazon CloudWatch product page.</p>

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 (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, CloudWatch Monitoring, Enable Detailed Monitoring**.
4. In the **Enable Detailed Monitoring** dialog box, choose **Yes, Enable**.
5. Choose **Close**.

To enable detailed monitoring when launching an instance (console)

When launching an instance using the AWS Management Console, select the **Monitoring** check box on the **Configure Instance Details** page.

To enable detailed monitoring for an existing instance (AWS CLI)

Use the following `monitor-instances` command to enable detailed monitoring for the specified instances.

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

To enable detailed monitoring when launching an instance (AWS CLI)

Use the `run-instances` command with the `--monitoring` flag to enable detailed monitoring.

```
aws ec2 run-instances --image-id ami-09092360 --monitoring Enabled=true...
```

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 (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, CloudWatch Monitoring, Disable Detailed Monitoring**.
4. In the **Disable Detailed Monitoring** dialog box, choose **Yes, Disable**.
5. Choose **Close**.

To disable detailed monitoring (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 \(p. 509\)](#).

Instance Metrics

The `AWS/EC2` namespace includes the following CPU credit metrics for your [burstable performance instances \(p. 116\)](#).

Metric	Description
<code>CPUCreditUsage</code>	<p>The number of CPU credits spent by the instance for CPU utilization. One CPU credit equals one vCPU running at 100% utilization for one minute or an equivalent combination of vCPUs, utilization, and time (for example, one vCPU running at 50% utilization for two minutes or two vCPUs running at 25% utilization for two minutes).</p> <p>CPU credit metrics are available at a five-minute frequency only. If you specify a period greater than five minutes, use the <code>Sum</code> statistic instead of the <code>Average</code> statistic.</p> <p>Units: Credits (vCPU-minutes)</p>
<code>CPUCreditBalance</code>	The number of earned CPU credits that an instance has accrued since it was launched or started. For T2 Standard, the

Metric	Description
CPUCreditBalance	<p>CPUCreditBalance also includes the number of launch credits that have been accrued.</p> <p>Credits are accrued in the credit balance after they are earned, and removed from the credit balance when they are spent. The credit balance has a maximum limit, determined by the instance size. After the limit is reached, any new credits that are earned are discarded. For T2 Standard, launch credits do not count towards the limit.</p> <p>The credits in the CPUCreditBalance are available for the instance to spend to burst beyond its baseline CPU utilization.</p> <p>When an instance is running, credits in the CPUCreditBalance do not expire. When a T3 instance stops, the CPUCreditBalance value persists for seven days. Thereafter, all accrued credits are lost. When a T2 instance stops, the CPUCreditBalance value does not persist, and all accrued credits are lost.</p> <p>CPU credit metrics are available at a five-minute frequency only.</p> <p>Units: Credits (vCPU-minutes)</p>
CPUSurplusCreditBalance	<p>The number of surplus credits that have been spent by an unlimited instance when its CPUCreditBalance value is zero.</p> <p>The CPUSurplusCreditBalance value is paid down by earned CPU credits. If the number of surplus credits exceeds the maximum number of credits that the instance can earn in a 24-hour period, the spent surplus credits above the maximum incur an additional charge.</p> <p>Units: Credits (vCPU-minutes)</p>
CPUSurplusCreditsCharged	<p>The number of spent surplus credits that are not paid down by earned CPU credits, and which thus incur an additional charge.</p> <p>Spent surplus credits are charged when any of the following occurs:</p> <ul style="list-style-type: none"> The spent surplus credits exceed the maximum number of credits that the instance can earn in a 24-hour period. Spent surplus credits above the maximum are charged at the end of the hour. The instance is stopped or terminated. The instance is switched from unlimited to standard. <p>Units: Credits (vCPU-minutes)</p>

The AWS/EC2 namespace includes the following instance metrics.

Metric	Description
CPUUtilization	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.

Metric	Description
	<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>
DiskWriteOps	<p>Completed write operations to all instance store volumes available to the instance in a specified period of time.</p> <p>To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period.</p> <p>If there are no instance store volumes, either the value is 0 or the metric is not reported.</p> <p>Units: Count</p>
DiskReadBytes	<p>Bytes read from all instance store volumes available to the instance.</p> <p>This metric is used to determine the volume of the data the application reads from the hard disk of the instance. This can be used to determine the speed of the application.</p> <p>The number reported is the number of bytes received during the period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to find Bytes/second. If you have detailed (one-minute) monitoring, divide it by 60.</p> <p>If there are no instance store volumes, either the value is 0 or the metric is not reported.</p> <p>Units: Bytes</p>

Metric	Description
DiskWriteBytes	<p>Bytes written to all instance store volumes available to the instance.</p> <p>This metric is used to determine the volume of the data the application writes onto the hard disk of the instance. This can be used to determine the speed of the application.</p> <p>The number reported is the number of bytes received during the period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to find Bytes/second. If you have detailed (one-minute) monitoring, divide it by 60.</p> <p>If there are no instance store volumes, either the value is 0 or the metric is not reported.</p> <p>Units: Bytes</p>
NetworkIn	<p>The number of bytes received on all network interfaces by the instance. This metric identifies the volume of incoming network traffic to a single instance.</p> <p>The number reported is the number of bytes received during the period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to find Bytes/second. If you have detailed (one-minute) monitoring, divide it by 60.</p> <p>Units: Bytes</p>
NetworkOut	<p>The number of bytes sent out on all network interfaces by the instance. This metric identifies the volume of outgoing network traffic from a single instance.</p> <p>The number reported is the number of bytes sent during the period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to find Bytes/second. If you have detailed (one-minute) monitoring, divide it by 60.</p> <p>Units: Bytes</p>
NetworkPacketsIn	<p>The number of packets received on all network interfaces by the instance. This metric identifies the volume of incoming traffic in terms of the number of packets on a single instance. This metric is available for basic monitoring only.</p> <p>Units: Count</p> <p>Statistics: Minimum, Maximum, Average</p>
NetworkPacketsOut	<p>The number of packets sent out on all network interfaces by the instance. This metric identifies the volume of outgoing traffic in terms of the number of packets on a single instance. This metric is available for basic monitoring only.</p> <p>Units: Count</p> <p>Statistics: Minimum, Maximum, Average</p>

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>
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 the Nitro-based instances that are not bare metal instances. For the list of Nitro-based instance types, see [Nitro-based Instances \(p. 113\)](#).

Note

Metric values for Nitro-based instances will always be integers (whole numbers), whereas values for Xen-based instances support decimals. Therefore, low instance CPU utilization on Nitro-based instances may appear to be rounded down to 0.

Metric	Description
EBSReadOps	<p>Completed read operations from all Amazon EBS volumes attached to the instance in a specified period of time.</p> <p>To calculate the average read I/O operations per second (Read IOPS) for the period, divide the total operations in the period by the number of seconds in that period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to calculate the Read IOPS. If</p>

Metric	Description
	<p>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	<p>Bytes written to all EBS volumes attached to the instance in a specified period of time.</p> <p>The number reported is the number of bytes written during the period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to find Write Bytes/second. If you have detailed (one-minute) monitoring, divide it by 60.</p> <p>Unit: Bytes</p>
EBSIOBalance%	<p>Available only for the smaller instance sizes. Provides information about the percentage of I/O credits remaining in the burst bucket. This metric is available for basic monitoring only.</p> <p>The Sum statistic is not applicable to this metric.</p> <p>Unit: Percent</p>
EBSByteBalance%	<p>Available only for the smaller instance sizes. Provides information about the percentage of throughput credits remaining in the burst bucket. This metric is available for basic monitoring only.</p> <p>The Sum statistic is not applicable to this metric.</p> <p>Unit: Percent</p>

For information about the metrics provided for your EBS volumes, see [Amazon EBS Metrics \(p. 758\)](#).
For information about the metrics provided for your Spot fleets, see [CloudWatch Metrics for Spot Fleet \(p. 264\)](#).

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 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.

Listing Metrics Using the Console

Metrics are grouped first by namespace, and then by the various dimension combinations within each namespace. For example, you can view all metrics provided by Amazon EC2, or metrics grouped by instance ID, instance type, image (AMI) ID, or Auto Scaling group.

To view available metrics by category (console)

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** metric namespace.

The screenshot shows the 'All metrics' tab selected in the top navigation bar. A search bar at the top allows for searching 'any metric, dimension or resource id'. Below it, a summary section shows '722 Metrics' across several services:

Service	Metrics
EBS	117 Metrics
EC2	316 Metrics
EFS	7 Metrics
ELB	210 Metrics
ElasticBeanstalk	8 Metrics
RDS	60 Metrics
S3	4 Metrics

4. Select a metric dimension (for example, **Per-Instance Metrics**).

The screenshot shows the 'Graphed metrics' tab selected for the EC2 service. The URL in the address bar indicates the user is viewing 'EC2' metrics. A search bar is present at the top. Below it, a summary section shows '103 Metrics' across five dimensions:

Dimension	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**.

The screenshot shows the AWS CloudWatch Metrics console. In the main list, there are several rows for an instance named 'my-instance'. One row has an Instance ID of 'i-abbc12a7'. A context menu is open over this row, with the cursor hovering over the 'Metric Name' column. The menu options are: 'Add to search', 'Search for this only', 'Add to graph', 'Graph this metric only', 'Graph all search results', and 'Jump to resource'. The 'Jump to resource' option is highlighted with a dark background.

All metrics	Graphed metrics	Graph options
All > EC2 > Per-Instance Metrics	Search for any metric, dimension or resource id	
Instance Name (192)	InstanceId	Metric Name
my-instance	i-abbc12a7	CPUUtilization
my-instance		DiskReadBytes
my-instance		DiskReadOps
my-instance		DiskWriteBytes
my-instance		DiskWriteOps
my-instance		NetworkIn
my-instance		NetworkOut
my-instance	i-abbc12a7	NetworkPacketsIn
my-instance	i-abbc12a7	NetworkPacketsOut

Listing Metrics Using the AWS CLI

Use the [list-metrics](#) command to list the CloudWatch metrics for your instances.

To list all the available metrics for Amazon EC2 (AWS CLI)

The following example specifies the AWS/EC2 namespace to view all the metrics for Amazon EC2.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

The following is example output:

```
{  
    "Metrics": [  
        {  
            "Namespace": "AWS/EC2",  
            "Dimensions": [  
                {  
                    "Name": "InstanceId",  
                    "Value": "i-1234567890abcdef0"  
                }  
            ],  
            "MetricName": "NetworkOut"  
        },  
        {  
            "Namespace": "AWS/EC2",  
            "Dimensions": [  
                {  
                    "Name": "InstanceId",  
                    "Value": "i-1234567890abcdef0"  
                }  
            ],  
            "MetricName": "CPUUtilization"  
        },  
        {  
            "Namespace": "AWS/EC2",  
            "Dimensions": [  
                {  
                    "Name": "InstanceId",  
                    "Value": "i-1234567890abcdef0"  
                }  
            ]  
        }  
    ]  
}
```

```
        },
        ],
        "MetricName": "NetworkIn"
    },
    ...
}
```

To list all the available metrics for an instance (AWS CLI)

The following example specifies the AWS/EC2 namespace and the `InstanceId` dimension to view the results for the specified instance only.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --dimensions
  Name=InstanceId,Value=i-1234567890abcdef0
```

To list a metric across all instances (AWS CLI)

The following example specifies the AWS/EC2 namespace and a metric name to view the results for the specified metric only.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --metric-name CPUUtilization
```

Get Statistics for Metrics for Your Instances

You can get statistics for the CloudWatch metrics for your instances.

Contents

- [Statistics Overview \(p. 509\)](#)
- [Get Statistics for a Specific Instance \(p. 510\)](#)
- [Aggregate Statistics Across Instances \(p. 512\)](#)
- [Aggregate Statistics by Auto Scaling Group \(p. 514\)](#)
- [Aggregate Statistics by AMI \(p. 515\)](#)

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 <code>Sum / SampleCount</code> during the specified period. By comparing this statistic with the <code>Minimum</code> and <code>Maximum</code> , you can determine the full scope of a metric.

Statistic	Description
	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

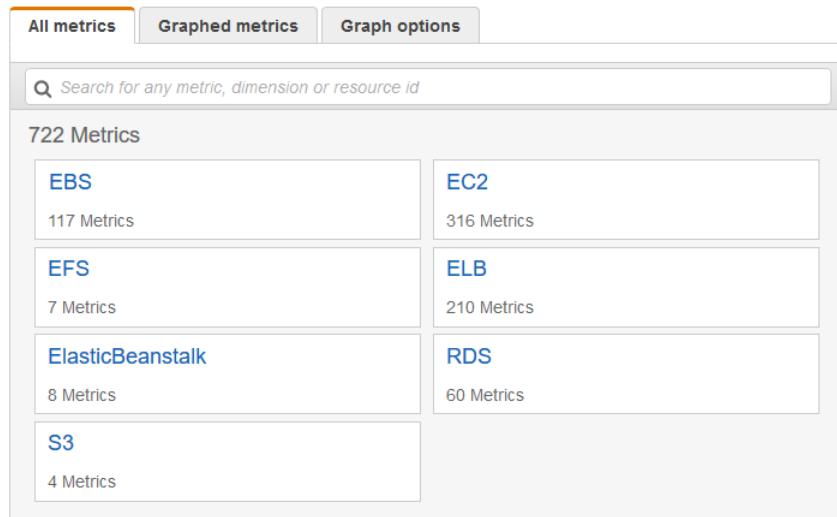
The following examples show you how to use the AWS Management Console or the AWS CLI to determine the maximum CPU utilization of a specific EC2 instance.

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 \(p. 499\)](#).

To display the CPU utilization for a specific instance (console)

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** metric namespace.

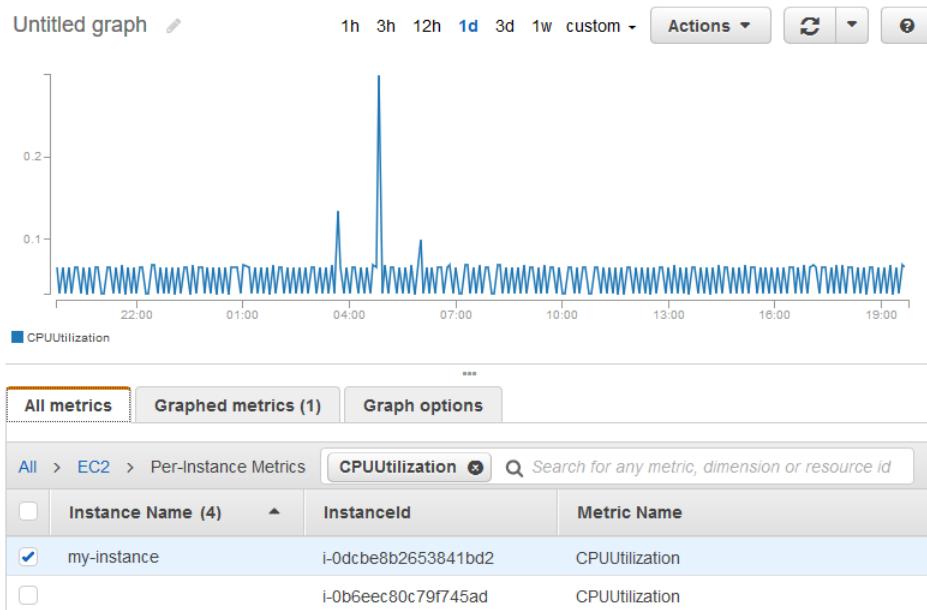


4. Choose the **Per-Instance Metrics** dimension.

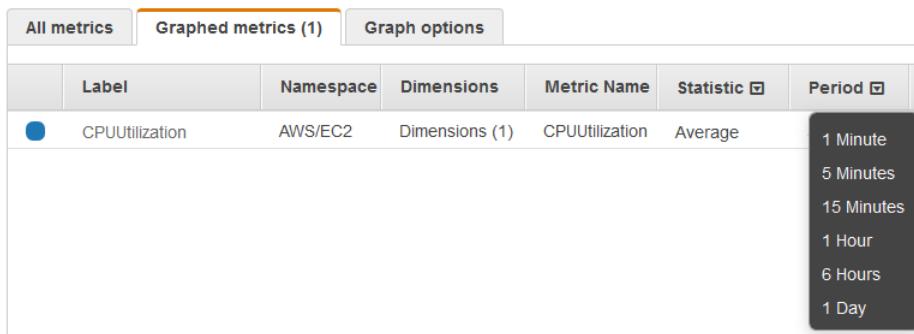
The screenshot shows the AWS CloudWatch Metrics console with the following interface elements:

- Top Navigation:** All metrics (selected), Graphed metrics, Graph options.
- Breadcrumbs:** All > EC2.
- Search Bar:** Search for any metric, dimension or resource id.
- Metrics List:**
 - 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

- In the search field, enter **CPUutilization** and press Enter. Choose the row for the specific instance, which displays a graph for the **CPUUtilization** metric for the instance. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.



- To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.



To get the CPU utilization for a specific instance (AWS CLI)

Use the following [get-metric-statistics](#) command to get the **CPUUtilization** metric for the specified instance, using the specified period and time interval:

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization --period 3600 \
--statistics Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 \
--start-time 2016-10-18T23:18:00 --end-time 2016-10-19T23:18:00
```

The following is example output. Each value represents the maximum CPU utilization percentage for a single EC2 instance.

```
{
  "Datapoints": [
    {
      "Timestamp": "2016-10-19T00:18:00Z",
      "Maximum": 0.33000000000000002,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2016-10-19T03:18:00Z",
      "Maximum": 99.67000000000002,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2016-10-19T07:18:00Z",
      "Maximum": 0.3400000000000002,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2016-10-19T12:18:00Z",
      "Maximum": 0.3400000000000002,
      "Unit": "Percent"
    },
    ...
  ],
  "Label": "CPUUtilization"
}
```

Aggregate Statistics Across Instances

Aggregate statistics are available for the instances that have detailed monitoring enabled. Instances that use basic monitoring are not included in the aggregates. In addition, Amazon CloudWatch does not aggregate data across regions. Therefore, metrics are completely separate between regions. Before you

can get statistics aggregated across instances, you must enable detailed monitoring (at an additional charge), which provides data in 1-minute periods.

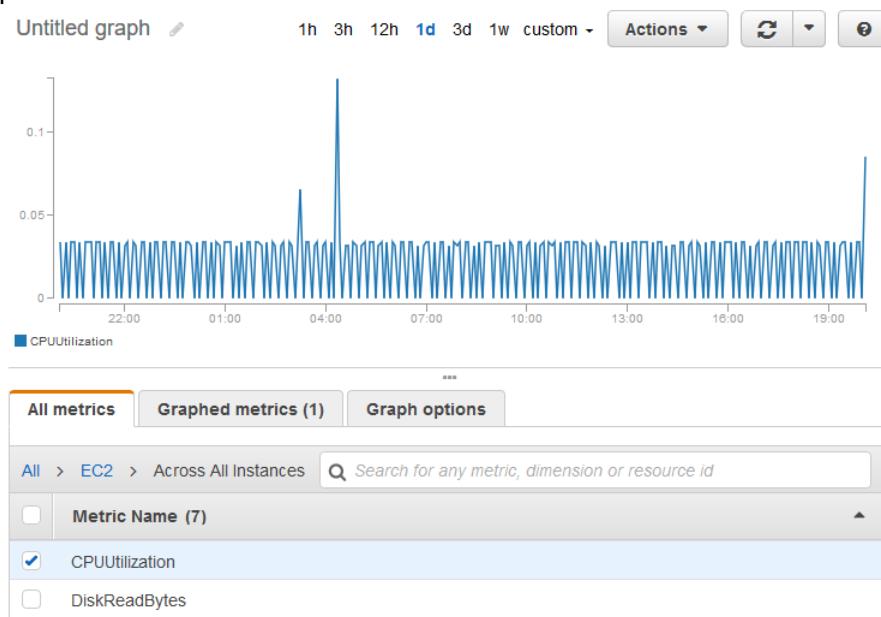
This example shows you how to use detailed monitoring to get the average CPU usage for your EC2 instances. Because no dimension is specified, CloudWatch returns statistics for all dimensions in the AWS / EC2 namespace.

Important

This technique for retrieving all dimensions across an AWS namespace does not work for custom namespaces that you publish to Amazon CloudWatch. With custom namespaces, you must specify the complete set of dimensions that are associated with any given data point to retrieve statistics that include the data point.

To display average CPU utilization across your instances (console)

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** namespace and then choose **Across All Instances**.
4. Choose the row that contains **CPUUtilization**, which displays a graph for the metric for all your EC2 instances. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.



5. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

To get average CPU utilization across your instances (AWS CLI)

Use the `get-metric-statistics` command as follows to get the average of the **CPUUtilization** metric across your instances.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization \
--period 3600 --statistics "Average" "SampleCount" \
--start-time 2016-10-11T23:18:00 --end-time 2016-10-12T23:18:00
```

The following is example output:

```
{  
    "Datapoints": [  
        {  
            "SampleCount": 238.0,  
            "Timestamp": "2016-10-12T07:18:00Z",  
            "Average": 0.038235294117647062,  
            "Unit": "Percent"  
        },  
        {  
            "SampleCount": 240.0,  
            "Timestamp": "2016-10-12T09:18:00Z",  
            "Average": 0.1667083333333332,  
            "Unit": "Percent"  
        },  
        {  
            "SampleCount": 238.0,  
            "Timestamp": "2016-10-11T23:18:00Z",  
            "Average": 0.041596638655462197,  
            "Unit": "Percent"  
        },  
        ...  
    ],  
    "Label": "CPUUtilization"  
}
```

Aggregate Statistics by Auto Scaling Group

You can aggregate statistics for the EC2 instances in an Auto Scaling group. Note that Amazon CloudWatch cannot aggregate data across regions. Metrics are completely separate between regions.

This example shows you how to retrieve the total bytes written to disk for one Auto Scaling group. The total is computed for one-minute periods for a 24-hour interval across all EC2 instances in the specified Auto Scaling group.

To display DiskWriteBytes for the instances in an Auto Scaling group (console)

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** namespace and then choose **By Auto Scaling Group**.
4. Choose the row for the **DiskWriteBytes** metric and the specific Auto Scaling group, which displays a graph for the metric for the instances in the Auto Scaling group. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.
5. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

To display DiskWriteBytes for the instances in an Auto Scaling group (AWS CLI)

Use the `get-metric-statistics` command as follows.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name DiskWriteBytes --  
period 360 \  
--statistics "Sum" "SampleCount" --dimensions Name=AutoScalingGroupName,Value=my-asg --  
start-time 2016-10-16T23:18:00 --end-time 2016-10-18T23:18:00
```

The following is example output:

```
{  
    "Datapoints": [  
        {
```

```
        "SampleCount": 18.0,
        "Timestamp": "2016-10-19T21:36:00Z",
        "Sum": 0.0,
        "Unit": "Bytes"
    },
    {
        "SampleCount": 5.0,
        "Timestamp": "2016-10-19T21:42:00Z",
        "Sum": 0.0,
        "Unit": "Bytes"
    }
],
"Label": "DiskWriteBytes"
}
```

Aggregate Statistics by AMI

You can aggregate statistics for your instances that have detailed monitoring enabled. Instances that use basic monitoring are not included. Note that Amazon CloudWatch cannot aggregate data across regions. Metrics are completely separate between regions.

Before you can get statistics aggregated across instances, you must enable detailed monitoring (at an additional charge), which provides data in 1-minute periods. For more information, see [Enable or Disable Detailed Monitoring for Your Instances \(p. 499\)](#).

This example shows you how to determine average CPU utilization for all instances that use a specific Amazon Machine Image (AMI). The average is over 60-second time intervals for a one-day period.

To display the average CPU utilization by AMI (console)

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2** namespace and then choose **By Image (AMI) Id**.
4. Choose the row for the **CPUUtilization** metric and the specific AMI, which displays a graph for the metric for the specified AMI. To name the graph, choose the pencil icon. To change the time range, select one of the predefined values or choose **custom**.
5. To change the statistic or the period for the metric, choose the **Graphed metrics** tab. Choose the column heading or an individual value, and then choose a different value.

To get the average CPU utilization for an image ID (AWS CLI)

Use the `get-metric-statistics` command as follows.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization --
period 3600 \
--statistics Average --dimensions Name=ImageId,Value=ami-3c47a355 --start-
time 2016-10-10T00:00:00 --end-time 2016-10-11T00:00:00
```

The following is example output. Each value represents an average CPU utilization percentage for the EC2 instances running the specified AMI.

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-10T07:00:00Z",
            "Average": 0.04100000000000009,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T07:01:00Z",
            "Average": 0.04100000000000009,
            "Unit": "Percent"
        }
    ]
}
```

```
        "Timestamp": "2016-10-10T14:00:00Z",
        "Average": 0.079579831932773085,
        "Unit": "Percent"
    },
    {
        "Timestamp": "2016-10-10T06:00:00Z",
        "Average": 0.036000000000000011,
        "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 \(p. 500\)](#).

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.

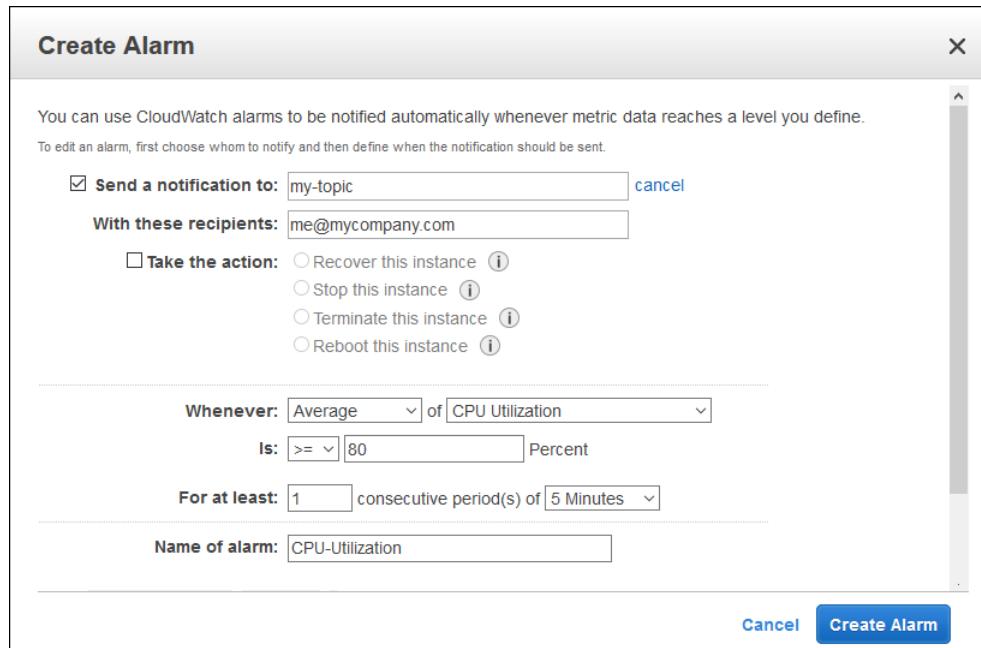
To create an alarm using the CloudWatch console

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**, enter a name for the SNS topic. For **With these recipients**, enter one or more email addresses to receive notification.
 - b. Specify the metric and the criteria for the policy. For example, you can leave the default settings for **Whenever** (Average of CPU Utilization). For **Is**, choose \geq and enter 80 percent. For **For at least**, enter 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 `AWSLambdaRoleForCloudWatchEvents` service-linked role enables AWS to perform alarm actions on your behalf. The first time you create an alarm in the AWS Management Console, the IAM CLI, or the IAM API, CloudWatch creates the service-linked role for you.

There are a number of scenarios in which you might want to automatically stop or terminate your instance. For example, you might have instances dedicated to batch payroll processing jobs or scientific computing tasks that run for a period of time and then complete their work. Rather than letting those instances sit idle (and accrue charges), you can stop or terminate them, which can help you to save money. The main difference between using the stop and the terminate alarm actions is that you can easily restart a stopped instance if you need to run it again later, and you can keep the same instance ID and root volume. However, you cannot restart a terminated instance. Instead, you must launch a new instance.

You can add the stop, terminate, reboot, or recover actions to any alarm that is set on an Amazon EC2 per-instance metric, including basic and detailed monitoring metrics provided by Amazon CloudWatch (in the AWS/EC2 namespace), as well as any custom metrics that include the `InstanceId` dimension, as long as its value refers to a valid running Amazon EC2 instance.

Console Support

You can create alarms using the Amazon EC2 console or the CloudWatch console. The procedures in this documentation use the Amazon EC2 console. For procedures that use the CloudWatch console, see [Create Alarms That Stop, Terminate, Reboot, or Recover an Instance](#) in the *Amazon CloudWatch User Guide*.

Permissions

If you are an AWS Identity and Access Management (IAM) user, you must have the following permissions to create or modify an alarm:

- `iam:CreateServiceLinkedRole`, `iam:GetPolicy`, `iam:GetPolicyVersion`, and `iam:GetRole`
 - For all alarms with Amazon EC2 actions
- `ec2:DescribeInstanceStatus` and `ec2:DescribeInstances` – For all alarms on Amazon EC2 instance status metrics
- `ec2:StopInstances` – For alarms with stop actions
- `ec2:TerminateInstances` – For alarms with terminate actions
- No additional 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 \(p. 518\)](#)
- [Adding Terminate Actions to Amazon CloudWatch Alarms \(p. 519\)](#)
- [Adding Reboot Actions to Amazon CloudWatch Alarms \(p. 520\)](#)
- [Adding Recover Actions to Amazon CloudWatch Alarms \(p. 521\)](#)
- [Using the Amazon CloudWatch Console to View Alarm and Action History \(p. 522\)](#)
- [Amazon CloudWatch Alarm Action Scenarios \(p. 522\)](#)

Adding Stop Actions to Amazon CloudWatch Alarms

You can create an alarm that stops an Amazon EC2 instance when a certain threshold has been met. For example, you may run development or test instances and occasionally forget to shut them off. You can create an alarm that is triggered when the average CPU utilization percentage has been lower than 10 percent for 24 hours, signaling that it is idle and no longer in use. You can adjust the threshold, duration, and period to suit your needs, plus you can add an Amazon Simple Notification Service (Amazon SNS) notification so that you receive an email when the alarm is triggered.

Instances that use an Amazon EBS volume as the root device can be stopped or terminated, whereas instances that use the instance store as the root device can only be terminated.

To create an alarm to stop an idle instance (Amazon EC2 console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.

3. Select the instance. On the **Monitoring** tab, choose **Create Alarm**.

4. In the **Create Alarm** dialog box, do the following:

- a. To receive an email when the alarm is triggered, for **Send a notification to**, choose an existing Amazon SNS topic, or choose **create topic** to create a new one.

To create a new topic, for **Send a notification to**, enter a name for the topic, and then for **With these recipients**, enter the email addresses of the recipients (separated by commas). After you create the alarm, you will receive a subscription confirmation email that you must accept before you can get notifications for this topic.

- b. Choose **Take the action, Stop this instance**.
- c. For **Whenever**, choose the statistic you want to use and then choose the metric. In this example, choose **Average** and **CPU Utilization**.
- d. For **Is**, specify the metric threshold. In this example, enter **10** percent.
- e. For **For at least**, specify the evaluation period for the alarm. In this example, enter **24** consecutive period(s) of **1 Hour**.
- f. To change the name of the alarm, for **Name of alarm**, enter a new name. Alarm names must contain only ASCII characters.

If you don't enter a name for the alarm, Amazon CloudWatch automatically creates one for you.

Note

You can adjust the alarm configuration based on your own requirements before creating the alarm, or you can edit them later. This includes the metric, threshold, duration, action, and notification settings. However, after you create an alarm, you cannot edit its name later.

- g. Choose **Create Alarm**.

Adding Terminate Actions to Amazon CloudWatch Alarms

You can create an alarm that terminates an EC2 instance automatically when a certain threshold has been met (as long as termination protection is not enabled for the instance). For example, you might want to terminate an instance when it has completed its work, and you don't need the instance again. If you might want to use the instance later, you should stop the instance instead of terminating it. For information on enabling and disabling termination protection for an instance, see [Enabling Termination Protection for an Instance](#) in the *Amazon EC2 User Guide for Linux Instances*.

To create an alarm to terminate an idle instance (Amazon EC2 console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance. On the **Monitoring** tab, choose **Create Alarm**.
4. In the **Create Alarm** dialog box, do the following:

- a. To receive an email when the alarm is triggered, for **Send a notification to**, choose an existing Amazon SNS topic, or choose **create topic** to create a new one.

To create a new topic, for **Send a notification to**, enter a name for the topic, and then for **With these recipients**, enter the email addresses of the recipients (separated by commas). After you create the alarm, you will receive a subscription confirmation email that you must accept before you can get notifications for this topic.

- b. Choose **Take the action, Terminate this instance**.
- c. For **Whenever**, choose a statistic and then choose the metric. In this example, choose **Average** and **CPU Utilization**.

- d. For **Is**, specify the metric threshold. In this example, enter **10** percent.
- e. For **For at least**, specify the evaluation period for the alarm. In this example, enter **24** consecutive period(s) of **1 Hour**.
- f. To change the name of the alarm, for **Name of alarm**, enter a new name. Alarm names must contain only ASCII characters.

If you don't enter a name for the alarm, Amazon CloudWatch automatically creates one for you.

Note

You can adjust the alarm configuration based on your own requirements before creating the alarm, or you can edit them later. This includes the metric, threshold, duration, action, and notification settings. However, after you create an alarm, you cannot edit its name later.

- g. Choose **Create Alarm**.

Adding Reboot Actions to Amazon CloudWatch Alarms

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically reboots the instance. The reboot alarm action is recommended for Instance Health Check failures (as opposed to the recover alarm action, which is suited for System Health Check failures). An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance. When you reboot an instance, it remains on the same physical host, so your instance keeps its public DNS name, private IP address, and any data on its instance store volumes.

Rebooting an instance doesn't start a new instance billing hour, unlike stopping and restarting your instance. For more information, see [Reboot Your Instance](#) in the *Amazon EC2 User Guide for Linux Instances*.

Important

To avoid a race condition between the reboot and recover actions, avoid setting the same number of evaluation periods for a reboot alarm and a recover alarm. We recommend that you set reboot alarms to three evaluation periods of one minute each. For more information, see [Evaluating an Alarm](#) in the *Amazon CloudWatch User Guide*.

To create an alarm to reboot an instance (Amazon EC2 console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance. On the **Monitoring** tab, choose **Create Alarm**.
4. In the **Create Alarm** dialog box, do the following:
 - a. To receive an email when the alarm is triggered, for **Send a notification to**, choose an existing Amazon SNS topic, or choose **create topic** to create a new one.

To create a new topic, for **Send a notification to**, enter a name for the topic, and for **With these recipients**, enter the email addresses of the recipients (separated by commas). After you create the alarm, you will receive a subscription confirmation email that you must accept before you can get notifications for this topic.
 - b. Select **Take the action, Reboot this instance**.
 - c. For **Whenever**, choose **Status Check Failed (Instance)**.
 - d. For **For at least**, specify the evaluation period for the alarm. In this example, enter **3** consecutive period(s) of **1 Minute**.
 - e. To change the name of the alarm, for **Name of alarm**, enter a new name. Alarm names must contain only ASCII characters.

If you don't enter a name for the alarm, Amazon CloudWatch automatically creates one for you.

- f. Choose **Create Alarm**.

Adding Recover Actions to Amazon CloudWatch Alarms

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance. If the instance becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair, you can automatically recover the instance. Terminated instances cannot be recovered. A recovered instance is identical to the original instance, including the instance ID, private IP addresses, Elastic IP addresses, and all instance metadata.

When the `StatusCheckFailed_System` alarm is triggered, and the recover action is initiated, you are notified by the Amazon SNS topic that you chose when you created the alarm and associated the recover action. During instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is lost. When the process is complete, information is published to the SNS topic you've configured for the alarm. Anyone who is subscribed to this SNS topic receives an email notification that includes the status of the recovery attempt and any further instructions. You notice an instance reboot on the recovered instance.

The recover action can be used only with `StatusCheckFailed_System`, not with `StatusCheckFailed_Instance`.

The following problems can cause system status checks to fail:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host that impact network reachability

The recover action is supported only on instances with the following characteristics:

- Use one of the following instance types: C3, C4, C5, C5n, M3, M4, M5, M5a, R3, R4, R5, R5a, T2, T3, X1, or X1e
- Use default or dedicated instance tenancy
- Use EBS volumes only (do not configure instance store volumes). For more information, see '['Recover this instance' is disabled](#)'.

If your instance has a public IP address, it retains the public IP address after recovery.

Important

To avoid a race condition between the reboot and recover actions, avoid setting the same number of evaluation periods for a reboot alarm and a recover alarm. We recommend that you set recover alarms to two evaluation periods of one minute each. For more information, see [Evaluating an Alarm](#) in the *Amazon CloudWatch User Guide*.

To create an alarm to recover an instance (Amazon EC2 console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance. On the **Monitoring** tab, choose **Create Alarm**.
4. In the **Create Alarm** dialog box, do the following:
 - a. To receive an email when the alarm is triggered, for **Send a notification to**, choose an existing Amazon SNS topic, or choose **create topic** to create a new one.

To create a new topic, for **Send a notification to**, enter a name for the topic, and for **With these recipients**, enter the email addresses of the recipients (separated by commas). After you create the alarm, you will receive a subscription confirmation email that you must accept before you can get email for this topic.

Note

- Users must subscribe to the specified SNS topic to receive email notifications when the alarm is triggered.
 - The AWS account root user always receives email notifications when automatic instance recovery actions occur, even if an SNS topic is not specified.
 - The AWS account root user always receives email notifications when automatic instance recovery actions occur, even if it is not subscribed to the specified SNS topic.
- b. Select **Take the action, Recover this instance**.
 - c. For **Whenever**, choose **Status Check Failed (System)**.
 - d. For **For at least**, specify the evaluation period for the alarm. In this example, enter **2 consecutive period(s) of 1 Minute**.
 - e. To change the name of the alarm, for **Name of alarm**, enter a new name. Alarm names must contain only ASCII characters.

If you don't enter a name for the alarm, Amazon CloudWatch automatically creates one for you.
 - f. Choose **Create Alarm**.

Using the Amazon CloudWatch Console to View Alarm and Action History

You can view alarm and action history in the Amazon CloudWatch console. Amazon CloudWatch keeps the last two weeks' worth of alarm and action history.

To view the history of triggered alarms and actions (CloudWatch console)

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Alarms**.
3. Select an alarm.
4. The **Details** tab shows the most recent state transition along with the time and metric values.
5. Choose the **History** tab to view the most recent history entries.

Amazon CloudWatch Alarm Action Scenarios

You can use the Amazon EC2 console to create alarm actions that stop or terminate an Amazon EC2 instance when certain conditions are met. In the following screen capture of the console page where you set the alarm actions, we've numbered the settings. We've also numbered the settings in the scenarios that follow, to help you create the appropriate actions.

Create Alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.

To edit an alarm, first choose whom to notify and then define when the notification should be sent.

① Send a notification to: [create topic](#)

② Take the action: Recover this instance [i](#)
 Stop this instance [i](#)
 Terminate this instance [i](#)
 Reboot this instance [i](#)

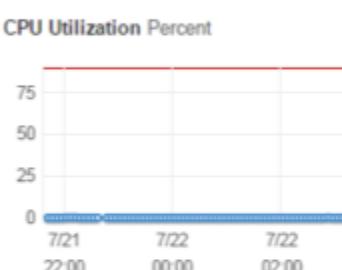
Whenever: ② of ③

Is: ④ ⑤ Percent

For at least: ⑥ consecutive period(s) of ⑦

Name of alarm:

CPU Utilization Percent



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](#).

Sending Logs, Events, and Performance Counters to Amazon CloudWatch

You can configure your Amazon EC2 instances to send Windows Server logs, events, and performance counters to Amazon CloudWatch Logs and Amazon CloudWatch Events. Amazon EC2 offers several methods for configuring your instances to export this data. The method you choose will depend, in part, on the version of Windows Server you are running and the version of the configuration agent running on your instance. It will also depend on whether you want to manually configure your instances to use a local configuration file or remotely configure them using Systems Manager Run Command or Systems Manager State Manager.

Important

The unified CloudWatch Agent has replaced the SSM Agent as the tool for sending log data to Amazon CloudWatch Logs. Support for using the SSM Agent to send log data will be deprecated in the near future. We recommend that you begin using the unified CloudWatch Agent for your log collection processes as soon as possible. For more information, see the following topics:

- [Send Logs to CloudWatch Logs \(CloudWatch Agent\)](#) in the *AWS Systems Manager User Guide*

- [Migrate Windows Server Instance Log Collection to the CloudWatch Agent in the AWS Systems Manager User Guide](#)
- [Collect Metrics from Amazon Elastic Compute Cloud Instances and On-Premises Servers with the CloudWatch Agent in the Amazon CloudWatch User Guide](#)

For more information about CloudWatch Logs, see the [Amazon CloudWatch Logs User Guide](#). For more information about Systems Manager, see the [AWS Systems Manager User Guide](#).

Note

We refer to CloudWatch Logs, CloudWatch Events, and CloudWatch collectively as CloudWatch, unless otherwise noted.

Amazon EC2 instances use an agent to send log data to CloudWatch. With Windows Server 2008 to Windows Server 2012 R2, the agent is either the EC2Config service or SSM Agent. With Windows Server 2016, the agent is SSM Agent. For more information, see [Installing and Configuring SSM Agent](#).

Contents

- [Methods to Send Instance Metrics to CloudWatch \(p. 527\)](#)
- [Preliminary Tasks for Configuring Integration with CloudWatch \(p. 528\)](#)
- [Configure Instances for CloudWatch \(p. 536\)](#)

Methods to Send Instance Metrics to CloudWatch

The following table describes the methods available to integrate with CloudWatch.

Method	Description
Systems Manager Run Command	<p>Complete the prerequisites for Systems Manager. Create a Systems Manager document and remotely send commands to the instances. The agent on the instances (SSM Agent) starts to send data to CloudWatch within a few minutes. For more information, see Use Systems Manager Run Command (p. 537).</p> <p>Pros – Commands are executed remotely.</p> <p>Cons – Requires additional setup.</p>
Systems Manager State Manager	<p>Complete the prerequisites for Systems Manager. Create a Systems Manager document and associate it with one or more instances. The agent on the instances starts to send data according to the schedule defined in the association. For more information, see Use Systems Manager State Manager (p. 537).</p> <p>Pros – Commands are executed remotely on a schedule.</p> <p>Cons – Requires additional setup.</p>
Local configuration file	<p>Create a Systems Manager document and copy it to each instance. Copy any updates to the configuration file to each instance manually as well. The agent on the instance (SSM Agent or EC2Config) starts sending data to CloudWatch within a few minutes. For more information, see Use a Local Configuration File (p. 538).</p> <p>Pros – No additional setup required.</p> <p>Cons – Legacy process. Requires logging in to each instance.</p>

Preliminary Tasks for Configuring Integration with CloudWatch

Complete the following preliminary tasks to configure integration with CloudWatch. These tasks apply to all methods for configuring instances to send logs, events, and performance counters to CloudWatch.

Important

The unified CloudWatch Agent has replaced the SSM Agent as the tool for sending log data to Amazon CloudWatch Logs. Support for using the SSM Agent to send log data will be deprecated in the near future. We recommend that you begin using the unified CloudWatch Agent for your log collection processes as soon as possible. For more information, see the following topics:

- [Send Logs to CloudWatch Logs \(CloudWatch Agent\)](#) in the *AWS Systems Manager User Guide*
- [Migrate Windows Server Instance Log Collection to the CloudWatch Agent](#) in the *AWS Systems Manager User Guide*
- [Collect Metrics from Amazon Elastic Compute Cloud Instances and On-Premises Servers with the CloudWatch Agent](#) in the *Amazon CloudWatch User Guide*

Tasks

- [Download the Sample Configuration File \(p. 528\)](#)
- [Configure the JSON File for CloudWatch \(p. 528\)](#)
- [Create an IAM User and Role for Systems Manager \(p. 535\)](#)
- [Verify Systems Manager Prerequisites \(p. 536\)](#)
- [Verify Internet Access \(p. 536\)](#)
- [Next Step \(p. 536\)](#)

Download the Sample Configuration File

Download the following sample file to your computer: [AWS.EC2.Windows.CloudWatch.json](#).

Configure the JSON File for CloudWatch

You determine which logs, events, and performance counters are sent to CloudWatch by specifying your choices in a configuration file. The process of creating this file and specifying your choices can take 30 minutes or more to complete. After you have completed this task once, you can reuse the configuration file on all of your instances.

Important

Use UTF-8 without BOM encoding for the CloudWatch.json file. If you don't, CloudWatch might not read the configuration file correctly.

Steps

- [Step 1: Configure Settings for CloudWatch \(p. 528\)](#)
- [Step 2: Configure the Data to Send \(p. 530\)](#)
- [Step 3: Configure Flow Control \(p. 535\)](#)

Step 1: Configure Settings for CloudWatch

By specifying credentials, a region, and a metric namespace for CloudWatch, you enable an instance to send performance counter data to CloudWatch. If you don't want to send performance counter data, you can skip this procedure. To send the same performance counter data to different locations, add

additional sections with unique IDs (for example, CloudWatch2 and CloudWatch3) and a different region for each ID.

To configure settings to send performance counter data to CloudWatch

1. In the JSON file, locate the CloudWatch section near the bottom of the file.

```
{  
    "Id": "CloudWatch",  
    "FullName":  
        "AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",  
        "Parameters": {  
            "AccessKey": "",  
            "SecretKey": "",  
            "Region": "us-west-1",  
            "NameSpace": "Windows/Default"  
        }  
},
```

2. (Optional) If you are using a local configuration file with local credentials, type your access key ID for AccessKey and your secret access key for SecretKey.

Important

If you use Systems Manager Run Command or State Manager, do not provide credentials in the configuration file, as there is a chance that they could be exposed in log files, including debug log files. You'll configure credentials using an IAM role.

3. For Region, type the region where you want to send log data (for example us-east-2). Although you can send performance counters to a different region from where you send your log data, we recommend that you set this parameter to the same region where your instance is running.
4. For NameSpace, type the metric namespace where performance counter data will be written.

Next, specify credentials, region, log group name, and a log stream namespace. This enables the instance to send log data to CloudWatch Logs. To send the same log data to different locations, add additional sections with unique IDs (for example, CloudWatchLogs2 and CloudWatchLogs3) and a different region for each ID.

To configure settings to send log data to CloudWatch Logs

1. In the JSON file, locate the CloudWatchLogs section.

```
{  
    "Id": "CloudWatchLogs",  
    "FullName":  
        "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",  
        "Parameters": {  
            "AccessKey": "",  
            "SecretKey": "",  
            "Region": "us-east-1",  
            "LogGroup": "Default-Log-Group",  
            "LogStream": "{instance_id}"  
        }  
},
```

2. (Optional) If you are using a local configuration file with local credentials, type your access key ID for AccessKey and your secret access key for SecretKey.

Important

If you use Systems Manager Run Command or State Manager, do not provide credentials in the configuration file, as there is a chance that they could be exposed in log files, including debug log files. You'll configure credentials using an IAM role.

3. For **Region**, type the region where you want to send log data (for example, `us-east-2`).
4. For **LogGroup**, type the name for your log group. This name will appear on the **Log Groups** screen in the CloudWatch console.
5. For **LogStream**, type the destination log stream. This name will appear on the **Log Groups > Streams** screen in the CloudWatch console.

If you use `{instance_id}`, the default, the log stream name is the instance ID of this instance.

If you specify a log stream name that doesn't already exist, CloudWatch Logs automatically creates it for you. You can define a log stream name using a literal string, the predefined variables `{instance_id}`, `{hostname}`, and `{ip_address}`, or a combination of these.

Step 2: Configure the Data to Send

You can send performance counters, event log data, Event Tracing for Windows (ETW) data, and other log data to Amazon CloudWatch Logs and Amazon CloudWatch Events.

To configure the performance counters to send to CloudWatch

1. Locate the **PerformanceCounter** section.

```
{
    "Id": "PerformanceCounter",
    "FullName":
"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.EC2.W
    "Parameters": {
        "CategoryName": "Memory",
        "CounterName": "Available MBytes",
        "InstanceName": "",
        "MetricName": "AvailableMemory",
        "Unit": "Megabytes",
        "DimensionName": "",
        "DimensionValue": ""
    }
},
```

2. You can select any performance counters that are available in Performance Monitor. You can select different categories to upload to CloudWatch as metrics, such as .NET CLR Data, ASP.NET Applications, HTTP Service, Memory, or Process and Processors.

For each performance counter, copy the **PerformanceCounter** section and change the **Id** parameter to make it unique (for example, `PerformanceCounter2`). Update the other parameters as necessary.

3. For **CategoryName**, type the performance counter category. To find the available categories and counters, do the following:
 - a. Open Performance Monitor. (To find Performance Monitor, right-click the **Start** menu, choose **Search**, and type `perfmon`.)
 - b. In the navigation pane, choose **Monitoring Tools, Performance Monitor**.
 - c. In the results pane, choose the green + (plus) button. The categories and counters are listed in the **Add Counters** dialog box.
4. For **CounterName**, type the name of the performance counter.
5. For **InstanceId**, type values from the **Add Counters** dialog box in Performance Monitor, which can be one of the following:
 - Blank, if the selected object has no instances.
 - A single instance of the selected object.

- `_Total` to use the aggregate of all instances.

Do not use an asterisk (*) to indicate all instances because each performance counter component only supports one metric.

6. For `MetricName`, type the CloudWatch metric that you want performance data to appear under.
7. For `Unit`, type the appropriate unit of measure for the metric. The possible values are as follows:
`Seconds | Microseconds | Milliseconds | Bytes | Kilobytes | Megabytes | Gigabytes | Terabytes | Bits | Kilobits | Megabits | Gigabits | Terabits | Percent | Count | Bytes/Second | Kilobytes/Second | Megabytes/Second | Gigabytes/Second | Terabytes/Second | Bits/Second | Kilobits/Second | Megabits/Second | Gigabits/Second | Terabits/Second | Count/Second | None.`
8. (Optional) To specify a dimension for your metric, type a dimension name and value for `DimensionName` and `DimensionValue`. These parameters provide another view when listing metrics. You can also use the same dimension for multiple metrics so that you can view all metrics belonging to a specific dimension.

Note

If the SSM Agent or the CloudWatch plugin is stopped, performance counter data is not logged in CloudWatch. This behavior is different than custom logs or Windows Event logs. Custom logs and Windows Event logs preserve performance counter data and upload it to CloudWatch after the SSM Agent or the CloudWatch plugin is available.

To send Windows application event log data to CloudWatch Logs

1. In the JSON file, locate the `ApplicationEventLog` section.

```
{  
    "Id": "ApplicationEventLog",  
    "FullName":  
        "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",  
    "Parameters": {  
        "LogName": "Application",  
        "Levels": "1"  
    }  
},
```

2. For `Levels`, specify the type of messages to upload. You can specify one of the following values:

- **1** – Upload only error messages.
- **2** – Upload only warning messages.
- **4** – Upload only information messages.

You can combine values to include more than one type of message. For example, a value of **3** uploads error messages (**1**) and warning messages (**2**). A value of **7** uploads error messages (**1**), warning messages (**2**), and information messages (**4**).

To send security log data to CloudWatch Logs

1. In the JSON file, locate the `SecurityEventLog` section.

```
{  
    "Id": "SecurityEventLog",  
    "FullName":  
        "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",  
    "Parameters": {  
        "LogName": "Security",  
        "Levels": "1"  
    }  
},
```

```
        "LogName": "Security",
        "Levels": "7"
    },
},
```

2. For **Levels**, type **7** to upload all messages.

To send system event log data to CloudWatch Logs

1. In the JSON file, locate the **SystemEventLog** section.

```
{
    "Id": "SystemEventLog",
    "FullName":
    "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. For **Levels**, specify the type of messages to upload. You can specify one of the following values:
 - **1** – Upload only error messages.
 - **2** – Upload only warning messages.
 - **4** – Upload only information messages.

You can combine values to include more than one type of message. For example, a value of **3** uploads error messages (**1**) and warning messages (**2**). A value of **7** uploads error messages (**1**), warning messages (**2**), and information messages (**4**).

To send other types of event log data to CloudWatch Logs

1. In the JSON file, add a new section.

```
{
    "Id": "",
    "FullName":
    "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "",
        "Levels": "7"
    }
},
```

2. For **Id**, type a name for the log to upload (for example, **WindowsBackup**).
3. For **LogName**, type the name of the log to upload. You can find the name of the log as follows.
 - a. Open Event Viewer.
 - b. In the navigation pane, choose **Applications and Services Logs**.
 - c. Navigate to the log, and then choose **Actions, Properties**.
4. For **Levels**, specify the type of messages to upload. You can specify one of the following values:
 - **1** – Upload only error messages.
 - **2** – Upload only warning messages.
 - **4** – Upload only information messages.

You can combine values to include more than one type of message. For example, a value of **3** uploads error messages (**1**) and warning messages (**2**). A value of **7** uploads error messages (**1**), warning messages (**2**), and information messages (**4**).

To send Event Tracing for Windows data to CloudWatch Logs

ETW (Event Tracing for Windows) provides an efficient and detailed logging mechanism that applications can write logs to. Each ETW is controlled by a session manager that can start and stop the logging session. Each session has a provider and one or more consumers.

1. In the JSON file, locate the `ETW` section.

```
{  
    "Id": "ETW",  
    "FullName":  
        "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",  
    "Parameters": {  
        "LogName": "Microsoft-Windows-WinINet/Analytic",  
        "Levels": "7"  
    }  
},
```

2. For `LogName`, type the name of the log to upload.
3. For `Levels`, specify the type of messages to upload. You can specify one of the following values:
 - **1** – Upload only error messages.
 - **2** – Upload only warning messages.
 - **4** – Upload only information messages.

You can combine values to include more than one type of message. For example, a value of **3** uploads error messages (**1**) and warning messages (**2**). A value of **7** uploads error messages (**1**), warning messages (**2**), and information messages (**4**).

To send custom logs (any text-based log file) to CloudWatch Logs

1. In the JSON file, locate the `CustomLogs` section.

```
{  
    "Id": "CustomLogs",  
    "FullName":  
        "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",  
    "Parameters": {  
        "LogDirectoryPath": "C:\\CustomLogs\\\",  
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",  
        "Encoding": "UTF-8",  
        "Filter": "",  
        "CultureName": "en-US",  
        "TimeZoneKind": "Local",  
        "LineCount": "5"  
    }  
},
```

2. For `LogDirectoryPath`, type the path where logs are stored on your instance.
3. For `TimestampFormat`, type the timestamp format you want to use. For a list of supported values, see the [Custom Date and Time Format Strings](#) topic on MSDN.

Important

Your source log file must have the timestamp at the beginning of each log line and there must be a space following the timestamp.

4. For `Encoding`, type the file encoding to use (for example, UTF-8). For a list of supported values, see the [Encoding Class](#) topic on MSDN.

Note

Use the encoding name, not the display name.

5. (Optional) For `Filter`, type the prefix of log names. Leave this parameter blank to monitor all files. For a list of supported values, see the [FileSystemWatcherFilter Property](#) topic on MSDN.
6. (Optional) For `CultureName`, type the locale where the timestamp is logged, the structure of which is based on RFC 5646. If `CultureName` is blank, it defaults to the same locale currently used by your Windows instance.

Note

The `div`, `div-MV`, `hu`, and `hu-HU` values are not supported.

7. (Optional) For `TimeZoneKind`, type `Local` or `UTC`. You can set this to provide time zone information when no time zone information is included in your log's timestamp. If this parameter is left blank and if your timestamp doesn't include time zone information, CloudWatch Logs defaults to the local time zone. This parameter is ignored if your timestamp already contains time zone information.
8. (Optional) For `LineCount`, type the number of lines in the header to identify the log file. For example, IIS log files have virtually identical headers. You could enter `3`, which would read the first three lines of the log file's header to identify it. In IIS log files, the third line is the date and timestamp, which is different between log files. For this reason, we recommend including at least one line of actual log data to uniquely fingerprint the log file.

To send IIS log data to CloudWatch Logs

1. In the JSON file, locate the `IISLog` section.

```
{  
    "Id": "IISLogs",  
    "FullName":  
        "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",  
    "Parameters": {  
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",  
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",  
        "Encoding": "UTF-8",  
        "Filter": "",  
        "CultureName": "en-US",  
        "TimeZoneKind": "UTC",  
        "LineCount": "5"  
    },  
},
```

2. For `LogDirectoryPath`, type the folder where IIS logs are stored for an individual site (for example, `C:\inetpub\logs\LogFiles\W3SVCn`).

Note

Only W3C log format is supported. IIS, NCSA, and Custom formats are not supported.

3. For `TimestampFormat`, type the timestamp format you want to use. For a list of supported values, see the [Custom Date and Time Format Strings](#) topic on MSDN.
4. For `Encoding`, type the file encoding to use (for example, UTF-8). For a list of supported values, see the [Encoding Class](#) topic on MSDN.

Note

Use the encoding name, not the display name.

-
5. (Optional) For `Filter`, type the prefix of log names. Leave this parameter blank to monitor all files. For a list of supported values, see the [FileSystemWatcherFilter Property](#) topic on MSDN.
 6. (Optional) For `CultureName`, type the locale where the timestamp is logged, the structure of which is based on RFC 5646. If `CultureName` is blank, it defaults to the same locale currently used by your Windows instance.

Note

The `div`, `div-MV`, `hu`, and `hu-HU` values are not supported.

7. (Optional) For `TimeZoneKind`, type `Local` or `UTC`. You can set this to provide time zone information when no time zone information is included in your log's timestamp. If this parameter is left blank and if your timestamp doesn't include time zone information, CloudWatch Logs defaults to the local time zone. This parameter is ignored if your timestamp already contains time zone information.
8. (Optional) For `LineCount`, type the number of lines in the header to identify the log file. For example, IIS log files have virtually identical headers. You could enter `3`, which would read the first three lines of the log file's header to identify it. In IIS log files, the third line is the date and time stamp, which is different between log files. For this reason, we recommend including at least one line of actual log data to uniquely fingerprint the log file.

Step 3: Configure Flow Control

Each data type must have a corresponding destination in the `Flows` section. For example, to send a performance counter defined in the `PerformanceCounter` section to the destination defined in the `CloudWatch` section, add `PerformanceCounter`, `CloudWatch` to the `Flows` section. Similarly, to send the custom log, ETW log, and system log to CloudWatch Logs, add `(CustomLogs, ETW, SystemEventLog), CloudWatchLogs`.

Warning

Adding a step that is not valid blocks the flow. For example, if you add a disk metric step, but your instance doesn't have a disk, all steps in the flow are blocked.

Note that you can send the same performance counter or log file to more than one destination. For example, to send the application log to two different destinations that you defined in the `CloudWatchLogs` section, add `ApplicationEventLog, (CloudWatchLogs, CloudWatchLogs2)` to the `Flows` section.

To configure flow control

1. In the `AWS.EC2.Windows.CloudWatch.json` file, locate the `Flows` section.

```
"Flows": {  
    "Flows": [  
        "PerformanceCounter,CloudWatch",  
        "(PerformanceCounter,PerformanceCounter2), CloudWatch2",  
        "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",  
        "CustomLogs, CloudWatchLogs2",  
        "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"  
    ]  
}
```

2. For `Flows`, add each data type that to be uploaded (for example, `ApplicationEventLog`) and its destination (for example, `CloudWatchLogs`).

Create an IAM User and Role for Systems Manager

If you plan to use a local configuration file with local credentials, then you can skip this task.

An IAM role for instance credentials is required when you use Systems Manager Run Command or State Manager and optional when you use a local configuration file. This role enables Systems Manager to perform actions on the instance. You can optionally create a unique IAM user account for configuring and running Systems Manager. For more information, see [Configuring Access to Systems Manager](#) in the *AWS Systems Manager User Guide*. For information about how to attach an IAM role to an existing instance, see [Attaching an IAM Role to an Instance \(p. 638\)](#).

Verify Systems Manager Prerequisites

If you plan to use a local configuration file, then you can skip this task.

Before you use either Systems Manager Run Command or State Manager to configure integration with CloudWatch, verify that your instances meet the minimum requirements. For more information, see [Systems Manager Prerequisites](#) in the *AWS Systems Manager User Guide*.

Verify Internet Access

Your Amazon EC2 Windows Server instances and managed instances must have outbound internet access in order to send log and event data to CloudWatch. For more information about how to configure internet access, see [Internet Gateways](#) in the *Amazon VPC User Guide*.

Next Step

After you complete the preliminary tasks for configuring integration with CloudWatch, you can perform the procedure required to complete the integration. For more information, see [Configure Instances for CloudWatch \(p. 536\)](#).

Configure Instances for CloudWatch

Important

The unified CloudWatch Agent has replaced the SSM Agent as the tool for sending log data to Amazon CloudWatch Logs. Support for using the SSM Agent to send log data will be deprecated in the near future. We recommend that you begin using the unified CloudWatch Agent for your log collection processes as soon as possible. For more information, see the following topics:

- [Send Logs to CloudWatch Logs \(CloudWatch Agent\)](#) in the *AWS Systems Manager User Guide*
- [Migrate Windows Server Instance Log Collection to the CloudWatch Agent](#) in the *AWS Systems Manager User Guide*
- [Collect Metrics from Amazon Elastic Compute Cloud Instances and On-Premises Servers with the CloudWatch Agent](#) in the *Amazon CloudWatch User Guide*

Choose from the following methods to configure integration with CloudWatch:

- [Use Run Command \(p. 537\)](#)
- [Use State Manager \(p. 537\)](#)
- [Use a Local Configuration File \(p. 538\)](#)

Before You Begin

Verify that you have completed the prerequisites. For more information, see [Preliminary Tasks for Configuring Integration with CloudWatch \(p. 528\)](#).

Use Systems Manager Run Command

Run Command enables you to manage the configuration of your instances on demand. You specify a Systems Manager document, specify parameters, and execute the command on one or more instance. The SSM agent on the instance processes the command and configures the instance as specified.

You can use Run Command to configure integration with CloudWatch. After you configure integration, the SSM Agent sends all the logs you configured in your JSON file to CloudWatch. The time frame varies for when the information is sent. For the application, system, security, and event tracing (Windows) logs, the system sends all information generated within the first minute of integration being enabled. Logs that occurred before this time are not included. For any custom log files and Internet Information Services (IIS) logs, State Manager reads the log files from the beginning.

If you previously enabled CloudWatch integration by using the EC2Config service, Run Command settings override the EC2Config settings stored on the instance. By default, these settings are stored in the following file on the instance: C:\Program Files\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json.

To configure integration with CloudWatch using Run Command

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Systems Manager Services**, **Run Command**.
3. Choose **Run a command**.
4. For **Command document**, choose **AWS-ConfigureCloudWatch**.
5. For **Target instances**, choose the instances to integrate with CloudWatch. If you do not see an instance in this list, it might not be configured for Run Command. For more information, see [Systems Manager Prerequisites](#) in the *AWS Systems Manager User Guide*.
6. For **Status**, choose **Enabled**.
7. For **Properties**, copy and paste your JSON content.
8. Complete the remaining optional fields and choose **Run**.

Use the following procedure to view the results of command execution in the Amazon EC2 console.

To view command output in the console

1. Select a command.
2. Choose the **Output** tab.
3. Choose **View Output**. The command output page shows the results of your command execution.

Use Systems Manager State Manager

State Manager enables you to manage the configuration of your Windows instances while they are running. You create a configuration document, which describes configuration tasks (for example, sending performance counters to CloudWatch and logs to CloudWatch Logs), and associate the configuration document with one or more running Windows instances. The SSM agent on the instance processes the configuration document and configures the instance as specified.

You can use Systems Manager State Manager (formerly called SSM Config) to configure integration with CloudWatch. After you configure integration, the SSM Agent sends all the logs you configured in your JSON file to CloudWatch. The time frame varies for when the information is sent. For the application, system, security, and event tracing (Windows) logs, the system sends all information generated within the first minute of integration being enabled. Logs that occurred before this time are not included. If you disable logging and then later re-enable logging, State Manager sends logs from where it left off. For any

custom log files and Internet Information Services (IIS) logs, State Manager reads the log files from the beginning.

If you previously enabled CloudWatch integration by using the EC2Config service, State Manager settings override the EC2Config settings stored on the instance. By default, these settings are stored in the following file on the instance: C:\Program Files\Amazon\Ec2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json.

To configure integration with CloudWatch using State Manager

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Systems Manager Services, State Manager**.
3. Choose **Create an association**.
4. For **Document**, choose **AWS-ConfigureCloudWatch**.
5. For **Document Version**, choose **Default version at runtime**.
6. For **Targets**, choose the instances to integrate with CloudWatch. If you do not see an instance in this list, it might not be configured for Run Command. For more information, see [Systems Manager Prerequisites](#) in the *AWS Systems Manager User Guide*.
7. For **Schedule**, choose how often you want Systems Manager to apply this policy. This indicates how often you want Systems Manager to ensure the integration with CloudWatch is still valid. This does not affect the frequency when the SSM Agent sends data to CloudWatch.
8. For **Parameters, Status**, choose **Enabled**. For **Properties**, copy and paste your JSON content.
9. (Optional) To send command output to an Amazon S3 bucket, choose **Advanced, Write to S3**.

Important
The Amazon EC2 console truncates output after 2500 characters. Configure an Amazon S3 bucket before executing commands using Systems Manager so that you can view the full output, beyond 2500 characters, in your bucket. For more information, see [Create a Bucket](#) in the *Amazon Simple Storage Service Getting Started Guide*.
10. Choose **Create Association**.
11. Choose the association you just created, and then choose **Apply Association Now**.

Use a Local Configuration File

The procedure to configure your instances using a local configuration file depends on the agent and the version of the agent that you're using. The SSM Agent is the only agent compatible with Windows Server 2016. With Windows Server 2008 to Windows Server 2012 R2, you must determine which version of EC2Config is running on your instance. For more information, see [Installing the Latest Version of EC2Config \(p. 384\)](#).

Agent

- [Use SSM Agent to Configure CloudWatch \(p. 538\)](#)
- [Use EC2Config 4.x to Configure CloudWatch \(p. 539\)](#)
- [Use EC2Config 3.x or Earlier to Configure CloudWatch \(p. 540\)](#)

Use SSM Agent to Configure CloudWatch

The following procedure describes how to configure CloudWatch using the SSM Agent on Amazon EC2 Windows Server 2016 instances.

To configure CloudWatch using SSM Agent

1. Download the latest version of the SSM Agent to your instance. For more information, see [Installing SSM Agent on Windows](#).

2. Open the `AWS.EC2.Windows.CloudWatch.json` file, and change `.IsEnabled` to `true`. This tells the agent to start sending data to CloudWatch immediately after it is started or restarted.
3. Save the file with the same name in the following folder on your Windows Server 2016 instance: `C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch\`.
4. Start or restart the SSM agent (`AmazonSSMAgent.exe`) using the Windows Services control panel or by sending the following command in PowerShell:

```
PS C:\> Restart-Service AmazonSSMAgent
```

After the SSM agent restarts, it detects the local configuration file and configures the instance for CloudWatch integration. If you change parameters and settings in the local configuration file, you need to restart the SSM agent to pick up the changes. If you want to disable CloudWatch integration on the instance, change `.IsEnabled` to `false` and save your changes in the configuration file.

Use EC2Config 4.x to Configure CloudWatch

To configure CloudWatch using EC2Config 4.x

1. Download the latest version of EC2Config to your instance. For more information, see [Installing the Latest Version of EC2Config \(p. 384\)](#).
2. (Optional) If you have an existing JSON file from an EC2Config 3.x integration with CloudWatch, open the file, and add the `.IsEnabled` section. This tells the agent to start sending data to CloudWatch immediately after it is started or restarted.

The `.IsEnabled` section must be located on the same level as the `EngineConfiguration` section. The following example illustrates this:

```
{
    "IsEnabled": true,
    "EngineConfiguration": {
        "PollInterval": "00:00:15",
        "Components": [
            {
                "Id": "OsCpuUtilization",
                "FullName": "AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent",
                "Parameters": {
                    "CategoryName": "Process",
                    ...
                }
            }
        ]
    }
}
```

3. Save the file with the same name in the following folder on your Windows Server 2008 through Windows Server 2012 R2 instance: `C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch\`.
4. Start or restart the SSM agent (`AmazonSSMAgent.exe`) using the Windows Services control panel or using the following PowerShell command:

```
PS C:\> Restart-Service AmazonSSMAgent
```

After the SSM agent restarts, it detects the local configuration file and configures the instance for CloudWatch integration. If you change parameters and settings in the local configuration file, you need to restart the SSM agent to pick up the changes. If you want to disable CloudWatch integration on the instance, change `.IsEnabled` to `false` and save your changes in the configuration file.

Use EC2Config 3.x or Earlier to Configure CloudWatch

Use the following procedure if you need to run an older version of EC2Config on your instances and continue to integrate with CloudWatch.

To configure CloudWatch using EC2Config 3.x or earlier

1. Connect to your Windows instance.
2. From the **Start** menu, choose **All Programs**, and then choose **EC2ConfigService Settings**.
3. On the **General** tab of the **Ec2 Service Properties** dialog box, under **CloudWatch Logs**, choose **Enable CloudWatch Logs integration**, and then choose **OK**.
4. If you made changes to the `AWS.EC2.Windows.CloudWatch.json` file, then you must restart the EC2Config service. For more information, see [Stopping, Restarting, Deleting, or Uninstalling EC2Config \(p. 385\)](#).

To enable CloudWatch Logs using user data

You can enable CloudWatch Logs by adding the following script to the user data field when you launch an instance. EC2Config will run this script every time your instance is restarted to make sure that CloudWatch Logs integration is enabled. To run this script only when an instance is first launched, remove `<persist>true</persist>` from the script.

```
<powershell>
$EC2SettingsFile="C:\Program Files\Amazon\Ec2ConfigService\Settings\Config.xml"
$xml = [xml](get-content $EC2SettingsFile)
$xmlElement = $xml.get_DocumentElement()
$xmlElementToModify = $xmlElement.Plugins

foreach ($element in $xmlElementToModify.Plugin)
{
    if ($element.name -eq "AWS.EC2.Windows.CloudWatch.Plugin")
    {
        $element.State="Enabled"
    }
}
$xml.Save($EC2SettingsFile)
</powershell>
<persist>true</persist>
```

Logging Amazon EC2 and Amazon EBS API Calls with AWS CloudTrail

Amazon EC2 and Amazon EBS are integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon EC2 and Amazon EBS. CloudTrail captures all API calls for Amazon EC2 and Amazon EBS as events, including calls from the console and from code calls to the APIs. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon EC2 and Amazon EBS. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Amazon EC2 and Amazon EBS, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the [AWS CloudTrail User Guide](#).

Amazon EC2 and Amazon EBS Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon EC2 and Amazon EBS, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing Events with CloudTrail Event History](#).

For an ongoing record of events in your AWS account, including events for Amazon EC2 and Amazon EBS, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- Overview for Creating a Trail
 - CloudTrail Supported Services and Integrations
 - Configuring Amazon SNS Notifications for CloudTrail
 - Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

All Amazon EC2 and Amazon EBS actions are logged by CloudTrail and are documented in the [Amazon EC2 API Reference](#). For example, calls to the [RunInstances](#), [DescribeInstances](#), or [CreateImage](#) actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or IAM user credentials.
 - Whether the request was made with temporary security credentials for a role or federated user.
 - Whether the request was made by another AWS service.

For more information, see the [CloudTrail userIdentity Element](#).

Understanding Amazon EC2 and Amazon EBS Log File Entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

The following log file record shows that a user terminated an instance.

```
{  
    "Records": [  
        {  
            "eventVersion": "1.03",  
            "userIdentity": {  
                "type": "Root",  
                "principalId": "123456789012",  
                "arn": "arn:aws:iam::123456789012:root".  
            }  
        }  
    ]  
}
```

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Understanding Amazon EC2 and
Amazon EBS Log File Entries

```
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
    "eventTime": "2016-05-20T08:27:45Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "TerminateInstances",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7botocore/1.4.1",
    "requestParameters": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-1a2b3c4d"
                }
            ]
        }
    },
    "responseElements": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-1a2b3c4d",
                    "currentState": {
                        "code": 32,
                        "name": "shutting-down"
                    },
                    "previousState": {
                        "code": 16,
                        "name": "running"
                    }
                }
            ]
        }
    },
    "requestID": "be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
    "eventID": "6e12345-2a4e-417c-aa78-7594fEXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
]
```

Network and Security

Amazon EC2 provides the following network and security features.

Features

- [Amazon EC2 Key Pairs and Windows Instances \(p. 543\)](#)
- [Amazon EC2 Security Groups for Windows Instances \(p. 548\)](#)
- [Controlling Access to Amazon EC2 Resources \(p. 562\)](#)
- [Amazon EC2 Instance IP Addressing \(p. 643\)](#)
- [Bring Your Own IP Addresses \(BYOIP\) \(p. 657\)](#)
- [Elastic IP Addresses \(p. 660\)](#)
- [Elastic Network Interfaces \(p. 665\)](#)
- [Enhanced Networking on Windows \(p. 684\)](#)
- [Placement Groups \(p. 692\)](#)
- [Network Maximum Transmission Unit \(MTU\) for Your EC2 Instance \(p. 697\)](#)
- [Virtual Private Clouds \(p. 700\)](#)
- [EC2-Classic \(p. 701\)](#)

Amazon EC2 Key Pairs and Windows Instances

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. With Windows instances, you use the private key to obtain the administrator password and then log in using RDP. For more information about key pairs and Linux instances, see [Amazon EC2 Key Pairs](#) in the *Amazon EC2 User Guide for Linux Instances*.

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 \(p. 544\)](#).

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 \(p. 544\)](#).

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.

Contents

- [Creating a Key Pair Using Amazon EC2 \(p. 544\)](#)
- [Importing Your Own Public Key to Amazon EC2 \(p. 544\)](#)
- [Retrieving the Public Key for Your Key Pair on Linux \(p. 546\)](#)
- [Retrieving the Public Key for Your Key Pair on Windows \(p. 546\)](#)
- [Retrieving the Public Key for Your Key Pair From Your Instance \(p. 546\)](#)
- [Verifying Your Key Pair's Fingerprint \(p. 547\)](#)
- [Deleting Your Key Pair \(p. 547\)](#)
- [Connecting to Your Windows Instance if You Lose Your Private Key \(p. 548\)](#)

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.

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.

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
- 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, C:\keys\my-key-pair.pub. 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, C:\keys\my-key-pair.pem. 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/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtJMxLvvwJryDUilBMTjYtwB+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:

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/public-keys/0/
openssh-key
```

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtJMxLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
```

```
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUzofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7OzlPnWOyN0qFU0XA246RA8QFYiCNYWI3f05p6KLxEXAMPLE 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 \(p. 453\)](#).

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:

```
C:\> 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:

```
C:\> 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)

Connecting to Your Windows Instance if You Lose Your Private Key

When you connect to a newly launched Windows instance, you decrypt the password for the Administrator account using the private key for the key pair that you specified when you launched the instance.

If you lose the Administrator password and you no longer have the private key, you can no longer access this Windows instance. You can replace the instance with a new instance that you launch with a new key pair and reset the Administrator password. For more information, see [Resetting a Lost or Expired Windows Administrator Password \(p. 974\)](#).

Amazon EC2 Security Groups for Windows Instances

A *security group* acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you can specify one or more security groups; otherwise, we use the default security group. You can add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group. 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.

When you launch an instance in a VPC, you must specify a security group that's created for that VPC. After you launch an instance, you can change its security groups. Security groups are associated with network interfaces. Changing an instance's security groups changes the security groups associated with the primary network interface (eth0). For more information, see [Changing an Instance's Security Groups](#) in the *Amazon VPC User Guide*. You can also change the security groups associated with any other network interface. For more information, see [Changing the Security Group \(p. 679\)](#).

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.

If you need to allow traffic to a Linux instance, see [Amazon EC2 Security Groups for Linux Instances](#) in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- [Security Group Rules \(p. 549\)](#)
 - [Connection Tracking \(p. 550\)](#)
- [Default Security Groups \(p. 551\)](#)
- [Custom Security Groups \(p. 552\)](#)
- [Working with Security Groups \(p. 552\)](#)

- [Creating a Security Group \(p. 552\)](#)
- [Describing Your Security Groups \(p. 553\)](#)
- [Adding Rules to a Security Group \(p. 553\)](#)
- [Updating Security Group Rules \(p. 555\)](#)
- [Deleting Rules from a Security Group \(p. 555\)](#)
- [Deleting a Security Group \(p. 556\)](#)
- [Security Group Rules Reference \(p. 556\)](#)
 - [Web Server Rules \(p. 556\)](#)
 - [Database Server Rules \(p. 557\)](#)
 - [Rules to Connect to Instances from Your Computer \(p. 558\)](#)
 - [Rules to Connect to Instances from an Instance with the Same Security Group \(p. 559\)](#)
 - [Rules for Path MTU Discovery \(p. 559\)](#)
 - [Rules for Ping/ICMP \(p. 559\)](#)
 - [DNS Server Rules \(p. 560\)](#)
 - [Amazon EFS Rules \(p. 560\)](#)
 - [Elastic Load Balancing Rules \(p. 561\)](#)

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.
- 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 \(p. 550\)](#).
- You can add and remove rules at any time. Your changes are automatically applied to the instances associated with the security group.

Note

The effect of some rule changes may depend on how the traffic is tracked. For more information, see [Connection Tracking \(p. 550\)](#).

- 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.
 - An individual IPv6 address. You must use the /128 prefix length; for example 2001:db8:1234:1a00::123/128.
 - A range of IPv4 addresses, in CIDR block notation, for example, 203.0.113.0/24.
 - A range of IPv6 addresses, in CIDR block notation, for example, 2001:db8:1234:1a00::/64.
 - The prefix list ID for the AWS service; for example, p1-1a2b3c4d. 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
 - A different security group for the same VPC
 - A different security group for a peer VPC in a VPC peering connection
- **(Optional) Description:** You can add a description for the rule; 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 \(p. 643\)](#). 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 3389 (RDP) from IP address 203.0.113.1 and another rule that allows access to TCP port 3389 from everyone, everyone has access to TCP port 3389.

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.

To ensure that traffic is immediately interrupted when you remove a security group rule, or to ensure that all inbound traffic is subject to firewall rules, you can use a network ACL for your subnet — network ACLs are stateless and therefore do not automatically allow response traffic. For more information, see [Network ACLs](#) in the *Amazon VPC User Guide*.

Default Security Groups

Your AWS account automatically has a *default security group* for the default VPC in each region. If you don't specify a security group when you launch an instance, the instance is automatically associated with the default security group for the VPC.

A default security group is named `default`, and it has an ID assigned by AWS. The following are the default rules for each default security group:

- Allows all inbound traffic from other instances associated with the default security group (the security group specifies itself as a source security group in its inbound rules)
- Allows all outbound traffic from the instance.

You can add or remove inbound and outbound rules for any default security group.

You can't delete a default security group. If you try to delete a default security group, you'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:

a-z, A-Z, 0-9, spaces, and ._-:/()#@[]+=&;{}!\$*

A security group name cannot start with sg-. A security group name must be unique for the VPC.

The following are the default rules for a security group that you create:

- Allows no inbound traffic
- Allows all outbound traffic

After you've created a security group, you can change its inbound rules to reflect the type of inbound traffic that you want to reach the associated instances. You can also change its outbound rules.

For more information about the rules you can add to a security group, see [Security Group Rules Reference \(p. 556\)](#).

Working with Security Groups

You can create, view, update, and delete security groups and security group rules using the Amazon EC2 console.

Tasks

- [Creating a Security Group \(p. 552\)](#)
- [Describing Your Security Groups \(p. 553\)](#)
- [Adding Rules to a Security Group \(p. 553\)](#)
- [Updating Security Group Rules \(p. 555\)](#)
- [Deleting Rules from a Security Group \(p. 555\)](#)
- [Deleting a Security Group \(p. 556\)](#)

Creating a Security Group

You can create a custom security group using the Amazon EC2 console. 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. For **VPC**, choose the ID of the VPC.

6. You can start adding rules, or you can choose **Create** to create the security group now (you can always add rules later). For more information about adding rules, see [Adding Rules to a Security Group \(p. 553\)](#).

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. For **VPC**, choose the ID of the 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, you can change its security groups. For more information, see [Changing an Instance's Security Groups](#) in the *Amazon VPC User Guide*.

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 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. (Optional) Select **VPC ID** from the filter list, then choose the ID of the VPC.
4. Select a security group. We display general information in the **Description** tab, inbound rules on the **Inbound** tab, outbound rules on the **Outbound** tab, and tags on the **Tags** 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 \(p. 556\)](#).

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 0.0.0.0/0 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 (0.0.0.0/0) 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 \(p. 556\)](#).

5. Choose **Save**.
6. You can also specify outbound rules. On the **Outbound tab**, choose **Edit**, **Add Rule**, and do the following:
 - For **Type**, select the protocol.
 - If you select a custom TCP or UDP protocol, specify the port range in **Port Range**.
 - If you select a custom ICMP protocol, choose the ICMP type name from **Protocol**, and, if applicable, the code name from **Port Range**.
 - For **Destination**, choose one of the following:
 - **Custom**: in the provided field, you must specify an IP address in CIDR notation, a CIDR block, or another security group.
 - **Anywhere**: automatically adds the 0.0.0.0/0 IPv4 CIDR block. This option enables outbound traffic to all IP addresses.

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)

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)

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.

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)

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.

Examples

- [Web Server Rules \(p. 556\)](#)
- [Database Server Rules \(p. 557\)](#)
- [Rules to Connect to Instances from Your Computer \(p. 558\)](#)
- [Rules to Connect to Instances from an Instance with the Same Security Group \(p. 559\)](#)
- [Rules for Path MTU Discovery \(p. 559\)](#)
- [Rules for Ping/ICMP \(p. 559\)](#)
- [DNS Server Rules \(p. 560\)](#)
- [Amazon EFS Rules \(p. 560\)](#)
- [Elastic Load Balancing Rules \(p. 561\)](#)

Web Server Rules

The following inbound rules allow HTTP and HTTPS access from any IP address. If your VPC is enabled for IPv6, you can add rules to control inbound HTTP and HTTPS traffic from IPv6 addresses.

Protocol type	Protocol number	Port	Source IP	Notes
TCP	6	80 (HTTP)	0.0.0.0/0	Allows inbound HTTP access from any IPv4 address
TCP	6	443 (HTTPS)	0.0.0.0/0	Allows inbound HTTPS access from any IPv4 address
TCP	6	80 (HTTP)	::/0	Allows inbound HTTP access from any IPv6 address
TCP	6	443 (HTTPS)	::/0	Allows inbound HTTPS access from any IPv6 address

Database Server Rules

The following inbound rules are examples of rules you might add for database access, depending on what type of database you're running on your instance. For more information about Amazon RDS instances, see the [Amazon RDS User Guide](#).

For the source IP, specify one of the following:

- A specific IP address or range of IP addresses in your local network
- A security group ID for a group of instances that access the database

Protocol type	Protocol number	Port	Notes
TCP	6	1433 (MS SQL)	The default port to access a Microsoft SQL Server database, for example, on an Amazon RDS instance
TCP	6	3306 (MySQL/Aurora)	The default port to access a MySQL or Aurora database, for example, on an Amazon RDS instance
TCP	6	5439 (Redshift)	The default port to access an Amazon Redshift cluster database.
TCP	6	5432 (PostgreSQL)	The default port to access a PostgreSQL database, for example, on an Amazon RDS instance

Protocol type	Protocol number	Port	Notes
TCP	6	1521 (Oracle)	The default port to access an Oracle database, for example, on an Amazon RDS instance

You can optionally restrict outbound traffic from your database servers, for example, 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 outbound HTTPS access to any IPv6 address

Rules to Connect to Instances from Your Computer

To connect to your instance, your security group must have inbound rules that allow SSH access (for Linux instances) or RDP access (for Windows instances).

Protocol type	Protocol number	Port	Source IP
TCP	6	22 (SSH)	The public IPv4 address of your computer, or a range of IP addresses in 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.

Rules to Connect to Instances from an Instance with the Same Security Group

To allow instances that are associated with the same security group to communicate with each other, you must explicitly add rules for this.

The following table describes the inbound rule for a security group that enables associated instances to communicate with each other. The rule allows all types of traffic.

Protocol type	Protocol number	Ports	Source IP
-1 (All)	-1 (All)	-1 (All)	The ID of the security group

Rules for 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

Rules for Ping/ICMP

The ping command is a type of ICMP traffic. To ping your instance, you must add the following inbound ICMP rule.

Protocol type	Protocol number	ICMP type	ICMP code	Source IP
ICMP	1	8 (Echo)	N/A	The public IPv4 address of your

Protocol type	Protocol number	ICMP type	ICMP code	Source IP
				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 Rules

If you've set up your EC2 instance as a DNS server, you must ensure that TCP and UDP traffic can reach your DNS server over port 53.

For the source IP, specify one of the following:

- An IP address or range of IP addresses in a network
- The ID of a security group for the set of instances in your network that require access to the DNS server

Protocol type	Protocol number	Port
TCP	6	53
UDP	17	53

Amazon EFS Rules

If you're using an Amazon EFS file system with your Amazon EC2 instances, the security group that you associate with your Amazon EFS mount targets must allow traffic over the NFS protocol.

Protocol type	Protocol number	Ports	Source IP	Notes
TCP	6	2049 (NFS)	The ID of the security group.	Allows inbound NFS access from resources (including the mount target) associated with this security group.

To mount an Amazon EFS file system on your Amazon EC2 instance, you must connect to your instance. Therefore, the security group associated with your instance must have rules that allow inbound SSH from your local computer or local network.

Protocol type	Protocol number	Ports	Source IP	Notes
TCP	6	22 (SSH)	The IP address range of your local computer, or the range of IP addresses for your network.	Allows inbound SSH access from your local computer.

Elastic Load Balancing Rules

If you're using a load balancer, the security group associated with your load balancer must have rules that allow communication with your instances or targets.

Inbound				
Protocol type	Protocol number	Port	Source IP	Notes
TCP	6	The listener port	For an Internet-facing load-balancer: 0.0.0.0/0 (all IPv4 addresses) 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 \(p. 562\)](#)
- [Amazon EC2 Permission Attributes \(p. 562\)](#)
- [IAM and Amazon EC2 \(p. 563\)](#)
- [IAM Policies for Amazon EC2 \(p. 564\)](#)
- [IAM Roles for Amazon EC2 \(p. 633\)](#)
- [Authorizing Inbound Traffic for Your Windows Instances \(p. 640\)](#)

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 Windows Instances \(p. 640\)](#).

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 \(p. 57\)](#).

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 \(p. 796\)](#).

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 \(p. 564\)](#)
- [IAM Roles for Amazon EC2 \(p. 633\)](#)
- [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. 565)
Define actions in your policy	Actions for Amazon EC2 (p. 566)
Define specific resources in your policy	Amazon Resource Names for Amazon EC2 (p. 566)
Apply conditions to the use of the resources	Condition Keys for Amazon EC2 (p. 569)
Work with the available resource-level permissions for Amazon EC2	Supported Resource-Level Permissions for Amazon EC2 API Actions (p. 573)
Test your policy	Checking That Users Have the Required Permissions (p. 573)
Example policies for a CLI or SDK	Example Policies for Working with the AWS CLI or an AWS SDK (p. 601)
Example policies for the Amazon EC2 console	Example Policies for Working in the Amazon EC2 Console (p. 625)

Policy Structure

The following topics explain the structure of an IAM policy.

Contents

- [Policy Syntax \(p. 565\)](#)
- [Actions for Amazon EC2 \(p. 566\)](#)
- [Amazon Resource Names for Amazon EC2 \(p. 566\)](#)
- [Condition Keys for Amazon EC2 \(p. 569\)](#)
- [Checking That Users Have the Required Permissions \(p. 573\)](#)

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 \(p. 566\)](#).
- **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 \(p. 566\)](#). For more information about which API actions support which ARNs, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 573\)](#). 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 \(p. 569\)](#).

For more information about example IAM policy statements for Amazon EC2, see [Example Policies for Working with the AWS CLI or an AWS SDK \(p. 601\)](#).

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 \(p. 573\)](#).

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> : <i>account</i> :*
Customer gateway	arn:aws:ec2: <i>region</i> : <i>account</i> :customer-gateway/ <i>cgw-id</i> Where <i>cgw-id</i> is cgw-xxxxxxx
DHCP options set	arn:aws:ec2: <i>region</i> : <i>account</i> :dhcp-options/ <i>dhcp-options-id</i> Where <i>dhcp-options-id</i> is dopt-xxxxxxxx
Elastic GPU	arn:aws:ec2: <i>region</i> : <i>account</i> :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> : <i>account</i> :instance/ <i>instance-id</i> Where <i>instance-id</i> is i-xxxxxxxx or i-xxxxxxxxxxxxxxxxxxxx
Instance profile	arn:aws:iam:: <i>account</i> :instance-profile/ <i>instance-profile-name</i>

Resource Type	ARN
	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:region:account:internet-gateway/ <i>igw-id</i> Where <i>igw-id</i> is igw-xxxxxxxx
Key pair	arn:aws:ec2:region: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:region:account:launch-template/ <i>launch-template-id</i> Where <i>launch-template-id</i> is lt-xxxxxxxxxxxxxxxxxx
NAT gateway	arn:aws:ec2:region:account:natgateway/ <i>natgateway-id</i> Where <i>natgateway-id</i> is nat-xxxxxxxxxxxxxxxxxxxx
Network ACL	arn:aws:ec2:region:account:network-acl/ <i>nacl-id</i> Where <i>nacl-id</i> is acl-xxxxxxxx
Network interface	arn:aws:ec2:region:account:network-interface/ <i>eni-id</i> Where <i>eni-id</i> is eni-xxxxxxxx
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, my-cluster)
Reserved Instance	arn:aws:ec2:region:account:reserved-instances/ <i>reservation-id</i> Where <i>reservation-id</i> is xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
Route table	arn:aws:ec2:region:account:route-table/ <i>route-table-id</i> Where <i>route-table-id</i> is rtb-xxxxxxxx
Security group	arn:aws:ec2:region:account:security-group/ <i>security-group-id</i> Where <i>security-group-id</i> is sg-xxxxxxxx
Snapshot	arn:aws:ec2:region::snapshot/ <i>snapshot-id</i> Where <i>snapshot-id</i> is snap-xxxxxxxx or snap-xxxxxxxxxxxxxxxx, 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 sir-xxxxxxxx
Subnet	arn:aws:ec2:region:account:subnet/ <i>subnet-id</i> Where <i>subnet-id</i> is subnet-xxxxxxxx

Resource Type	ARN
Volume	arn:aws:ec2:region:account:volume/ <i>volume-id</i> Where <i>volume-id</i> is vol-xxxxxxxx or vol-xxxxxxxxxxxxxxxxxx
VPC	arn:aws:ec2:region:account:vpc/vpc- <i>id</i> Where <i>vpc-id</i> is vpc-xxxxxxx
VPC peering connection	arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering- <i>connection-id</i> Where <i>vpc-peering connection-id</i> is pcx-xxxxxxxx
VPN connection	arn:aws:ec2:region:account:vpn-connection/vpn- <i>connection-id</i> Where <i>vpn-connection-id</i> is vpn-xxxxxxxx
VPN gateway	arn:aws:ec2:region:account:vpn-gateway/vpn- <i>gateway-id</i> Where <i>vpn-gateway-id</i> is vgw-xxxxxxxx

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 \(p. 573\)](#).

Condition Key	Key-Value Pair	Evaluation Types
ec2:AccepterVpc	"ec2:AccepterVpc":"vpc-arn" Where <i>vpc-arn</i> is the VPC ARN for the accepter VPC in a VPC peering connection	ARN, Null
ec2:AuthorizedService	"ec2:AuthorizedService":"service-principal" Where <i>service-principal</i> is the service principal (for example, ecs.amazonaws.com)	String, Null
ec2:AuthorizedUser	"ec2:AuthorizedUser":"principal-arn" Where <i>principal-arn</i> is the ARN for the principal (for example, arn:aws:iam::123456789012:root)	ARN, Null
ec2:AvailabilityZone	"ec2:AvailabilityZone":"az-api-name" Where <i>az-api-name</i> is the name of the Availability Zone (for example, us-east-2a) To list your Availability Zones, use describe-availability-zones	String, Null
ec2>CreateAction	"ec2>CreateAction":"api-name" Where <i>api-name</i> is the name of the resource-creating action (for example, RunInstances)	String, Null
ec2:EbsOptimized	"ec2:EbsOptimized":"optimized-flag" Where <i>optimized-flag</i> is true false (for an instance)	Boolean, Null
ec2:ElasticGpuType	"ec2:ElasticGpuType":"elastic-gpu-type" Where <i>elastic-gpu-type</i> is the name of the elastic GPU type	String, Null
ec2:Encrypted	"ec2:Encrypted":"encrypted-flag" Where <i>encrypted-flag</i> is true false (for an EBS volume)	Boolean, Null
ec2:ImageType	"ec2:ImageType":"image-type-api-name" Where <i>image-type-api-name</i> is ami aki ari	String, Null
ec2:InstanceMarketType	"ec2:InstanceMarketType":"market-type" Where <i>market-type</i> is spot on-demand	String, Null
ec2:InstanceProfile	"ec2:InstanceProfile":"instance-profile-arn" Where <i>instance-profile-arn</i> is the instance profile ARN	ARN, Null
ec2:InstanceType	"ec2:InstanceType":"instance-type-api-name" Where <i>instance-type-api-name</i> is the name of the instance type.	String, Null

Condition Key	Key-Value Pair	Evaluation Types
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
ec2:PlacementGroup	"ec2:PlacementGroup":" <i>placement-group-arn</i> " Where <i>placement-group-arn</i> is the placement group ARN	ARN, Null
ec2:PlacementGroupStrategy	"ec2:PlacementGroupStrategy":" <i>placement-group-strategy</i> " Where <i>placement-group-strategy</i> is cluster spread	String, Null
ec2:ProductCode	"ec2:ProductCode":" <i>product-code</i> " Where <i>product-code</i> is the product code	String, Null
ec2:Public	"ec2:Public":" <i>public-flag</i> " Where <i>public-flag</i> is true false (for an AMI)	Boolean, Null
ec2:Region	"ec2:Region":" <i>region-name</i> " 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":" <i>vpc-arn</i> " Where <i>vpc-arn</i> is the VPC ARN for the requester VPC in a VPC peering connection	ARN, Null
ec2:ReservedInstancesOfferingType	"ec2:ReservedInstancesOfferingType":" <i>offering-type</i> " Where <i>offering-type</i> is No Upfront Partial Upfront All Upfront	String, Null

Condition Key	Key-Value Pair	Evaluation Types
<code>ec2:ResourceTag</code> / <code>"ec2:ResourceTag/tag-key":"tag-value"</code>	Where <code>tag-key</code> and <code>tag-value</code> are the tag-key pair	String, Null
<code>ec2:RootDeviceType</code> / <code>"ec2:RootDeviceType":"root-device-type-name"</code>	Where <code>root-device-type-name</code> is <code>ebs</code> <code>instance-store</code>	String, Null
<code>ec2:SnapshotTime</code> / <code>"ec2:SnapshotTime":"time"</code>	Where <code>time</code> is the snapshot creation time (for example, <code>2013-06-01T00:00:00Z</code>)	Date, Null
<code>ec2:Subnet</code>	<code>"ec2:Subnet":"subnet-arn"</code> Where <code>subnet-arn</code> is the subnet ARN	ARN, Null
<code>ec2:Tenancy</code>	<code>"ec2:Tenancy":"tenancy-attribute"</code> Where <code>tenancy-attribute</code> is <code>default</code> <code>dedicated</code> <code>host</code>	String, Null
<code>ec2:VolumeIops</code>	<code>"ec2:VolumeIops":"volume-iops"</code> Where <code>volume-iops</code> is the input/output operations per second (IOPS). For more information, see Amazon EBS Volume Types (p. 736) .	Numeric, Null
<code>ec2:VolumeSize</code>	<code>"ec2:VolumeSize":"volume-size"</code> Where <code>volume-size</code> is the size of the volume, in GiB	Numeric, Null
<code>ec2:VolumeType</code>	<code>"ec2:VolumeType":"volume-type-name"</code> Where <code>volume-type-name</code> is <code>gp2</code> for General Purpose SSD volumes, <code>io1</code> for Provisioned IOPS SSD volumes, <code>st1</code> for Throughput Optimized HDD volumes, <code>sc1</code> for Cold HDD volumes, or <code>standard</code> for Magnetic volumes.	String, Null
<code>ec2:Vpc</code>	<code>"ec2:Vpc":"vpc-arn"</code> Where <code>vpc-arn</code> 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*.

All Amazon EC2 actions support the `aws:RequestedRegion` and `ec2:Region` condition keys. For more information, see [Example: Restricting Access to a Specific Region \(p. 602\)](#).

The `ec2:SourceInstanceARN` key can be used for conditions that specify the ARN of the instance from which a request is made. This condition key is available AWS-wide and is not service-specific. For policy examples, see [Allows an EC2 Instance to Attach or Detach Volumes](#) and [Example: Allowing a Specific Instance to View Resources in Other AWS Services \(p. 624\)](#). 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 \(p. 601\)](#).

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 \(p. 601\)](#).

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 [Example: Read-Only Access \(p. 601\)](#). 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*.

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:AccepterVpc 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
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/tag-key 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/tag-key

API Action	Resources	Condition Keys
		ec2:RootDeviceType ec2:Tenancy
	Security group arn:aws:ec2: <i>region</i> :account:security-group/* arn:aws:ec2: <i>region</i> :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: <i>region</i> :account:vpc/* arn:aws:ec2: <i>region</i> :account:vpc/vpc- <i>id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Tenancy
AttachVolume	Instance arn:aws:ec2: <i>region</i> :account:instance/* arn:aws:ec2: <i>region</i> :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: <i>region</i> :account:volume/* arn:aws:ec2: <i>region</i> :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

API Action	Resources	Condition Keys
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>
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

Amazon Elastic Compute Cloud
User Guide for Windows Instances
IAM Policies

API Action	Resources	Condition Keys
	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
	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

API Action	Resources	Condition Keys
	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
	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

API Action	Resources	Condition Keys
	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
	Reserved Instance arn:aws:ec2:region:account:reserved-instances/* arn:aws:ec2:region:account:reserved-instances/ <i>reservation-id</i>	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 arn:aws:ec2:region:account:route-table/* arn:aws:ec2:region:account:route-table/ <i>route-table-id</i>	ec2:CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc aws:RequestTag/ <i>tag-key</i> aws:TagKeys

API Action	Resources	Condition Keys
	Security group arn:aws:ec2:region:account:security-group/* arn:aws:ec2:region:account:security-group/ <i>security-group-id</i>	ec2:CreateAction ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc aws:RequestTag/ <i>tag-key</i> aws:TagKeys
	Snapshot arn:aws:ec2:region::snapshot/* arn:aws:ec2:region::snapshot/ <i>snapshot-id</i>	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
	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

API Action	Resources	Condition Keys
	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
	VPN connection arn:aws:ec2:region:account:vpn-connection/* arn:aws:ec2:region:account:vpn-connection/vpn-connection- <i>id</i>	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- <i>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
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- <i>id</i> 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- <i>id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i>
DeleteDhcpOptions	DHCP options set arn:aws:ec2:region:account:dhcp-options/* arn:aws:ec2:region:account:dhcp-options/dhcp-options- <i>id</i>	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- <i>id</i>	ec2:Region ec2:ResourceTag/ <i>tag-key</i>

API Action	Resources	Condition Keys
DeleteLaunchTemplate	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>
DeleteLaunchTemplateVersion	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>
DeleteNetworkAcl	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> ec2:Vpc
DeleteNetworkAclEntry	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> ec2:Vpc
DeleteRoute	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
DeleteRouteTable	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
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

API Action	Resources	Condition Keys
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
	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

API Action	Resources	Condition Keys
	NAT gateway arn:aws:ec2:region:account:natgateway/* arn:aws:ec2:region:account:natgateway/ natgateway-id	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
	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

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: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/vpc- <i>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/vpn-connection- <i>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/vpn-gateway- <i>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	Assocation 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 arn:aws:ec2: <i>region</i> ::snapshot/* arn:aws:ec2: <i>region</i> ::snapshot/ <i>snapshot-id</i>	ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Owner ec2:ParentVolume ec2:Region ec2:SnapshotTime ec2:ResourceTag/ <i>tag-key</i> ec2:VolumeSize
	Subnet arn:aws:ec2: <i>region:account</i> :subnet/* arn:aws:ec2: <i>region:account</i> :subnet/ <i>subnet-id</i>	ec2:AvailabilityZone ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:Vpc
	Volume arn:aws:ec2: <i>region:account</i> :volume/*	ec2:AvailabilityZone ec2:Encrypted ec2:IsLaunchTemplateResource ec2:LaunchTemplate ec2:ParentSnapshot ec2:Region ec2:Volumelops ec2:VolumeSize ec2:VolumeType aws:RequestTag/ <i>tag-key</i> aws:TagKeys

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
Launch 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
	arn:aws:ec2: <i>region</i> :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: <i>region</i> :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
Launch 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:VolumeLops 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 [Launching Instances \(RunInstances\) \(p. 610\)](#).

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 \(p. 321\)](#) 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 \(p. 618\)](#).

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 \(p. 892\)](#).

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 [Tags in a Launch Template \(p. 617\)](#).

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 \(p. 601\)](#).

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 \(p. 625\)](#). For examples of IAM policies specific to Amazon VPC, see [Controlling Access to Amazon VPC Resources](#).

Examples

- [Example: Read-Only Access \(p. 601\)](#)
- [Example: Restricting Access to a Specific Region \(p. 602\)](#)
- [Working with Instances \(p. 602\)](#)
- [Working with Volumes \(p. 604\)](#)
- [Working with Snapshots \(p. 606\)](#)
- [Launching Instances \(RunInstances\) \(p. 610\)](#)
- [Example: Working with Reserved Instances \(p. 620\)](#)
- [Example: Tagging Resources \(p. 621\)](#)
- [Example: Working with IAM Roles \(p. 622\)](#)
- [Example: Working with Route Tables \(p. 623\)](#)
- [Example: Allowing a Specific Instance to View Resources in Other AWS Services \(p. 624\)](#)
- [Example: Working with Launch Templates \(p. 624\)](#)

Example: Read-Only Access

The following policy grants users permissions to use all Amazon EC2 API actions whose names begin with `Describe`. The `Resource` element uses a wildcard to indicate that users can specify all resources with these API actions. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 573\)](#).

Users don't have permission to perform any actions on the resources (unless another statement grants them permission to do so) because they're denied permission to use API actions by default.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": "ec2:Describe*",  
        "Resource": "*"  
    }  
}]
```

Example: Restricting Access to a Specific Region

The following policy denies users permission to use all Amazon EC2 API actions unless the region is EU (Frankfurt). It uses the global condition key `aws:RequestedRegion`, which is supported by all Amazon EC2 API actions.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Action": "ec2:*",  
            "Resource": "*",  
            "Condition": {  
                "StringNotEquals": {  
                    "aws:RequestedRegion": "eu-central-1"  
                }  
            }  
        }  
    ]  
}
```

Alternatively, you can use the condition key `ec2:Region`, which is specific to Amazon EC2 and is supported by all Amazon EC2 API actions.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Action": "ec2:*",  
            "Resource": "*",  
            "Condition": {  
                "StringNotEquals": {  
                    "ec2:Region": "eu-central-1"  
                }  
            }  
        }  
    ]  
}
```

Working with Instances

Examples

- [Example: Describe, Launch, Stop, Start, and Terminate All Instances \(p. 602\)](#)
- [Example: Describe All Instances, and Stop, Start, and Terminate Only Particular Instances \(p. 603\)](#)

Example: Describe, Launch, Stop, Start, and Terminate All Instances

The following policy grants users permissions to use the API actions specified in the `Action` element. The `Resource` element uses a `*` wildcard to indicate that users can specify all resources with these API actions. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 573\)](#).

The users don't have permission to use any other API actions (unless another statement grants them permission to do so) because users are denied permission to use API actions by default.

```
{
```

```

"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeInstances", "ec2:DescribeImages",
            "ec2:DescribeKeyPairs", "ec2:DescribeSecurityGroups",
            "ec2:DescribeAvailabilityZones",
            "ec2:RunInstances", "ec2:TerminateInstances",
            "ec2:StopInstances", "ec2:StartInstances"
        ],
        "Resource": "*"
    }
]
}

```

Example: Describe All Instances, and Stop, Start, and Terminate Only Particular Instances

The following policy allows users to describe all instances, to start and stop only instances i-1234567890abcdef0 and i-0598c7d356eba48d7, and to terminate only instances in the US East (N. Virginia) Region (us-east-1) with the resource tag "purpose=test".

The first statement uses a * wildcard for the Resource element to indicate that users can specify all resources with the action; in this case, they can list all instances. The * wildcard is also necessary in cases where the API action does not support resource-level permissions (in this case, ec2:DescribeInstances). For more information about which ARNs you can use with which Amazon EC2 API actions, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 573\)](#).

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"
                }
            }
        }
    ]
}

```

```
    }
]
}
```

Working with Volumes

Examples

- [Example: Attaching and Detaching Volumes \(p. 604\)](#)
- [Example: Creating a Volume \(p. 604\)](#)
- [Example: Creating a Volume with Tags \(p. 605\)](#)

Example: Attaching and Detaching Volumes

When an API action requires a caller to specify multiple resources, you must create a policy statement that allows users to access all required resources. If you need to use a `Condition` element with one or more of these resources, you must create multiple statements as shown in this example.

The following policy allows users to attach volumes with the tag "volume_user=*iam-user-name*" to instances with the tag "department=dev", and to detach those volumes from those instances. If you attach this policy to an IAM group, the `aws:username` policy variable gives each IAM user in the group permission to attach or detach volumes from the instances with a tag named `volume_user` that has his or her IAM user name as a value.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/department": "dev"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/volume_user": "${aws:username}"
                }
            }
        }
    ]
}
```

Example: Creating a Volume

The following policy allows users to use the [CreateVolume](#) API action. The user is allowed to create a volume only if the volume is encrypted and only if the volume size is less than 20 GiB.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateVolume"  
            ],  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",  
            "Condition":{  
                "NumericLessThan": {  
                    "ec2:VolumeSize" : "20"  
                },  
                "Bool":{  
                    "ec2:Encrypted" : "true"  
                }  
            }  
        }  
    ]  
}
```

Example: Creating a Volume with Tags

The following policy includes the `aws:RequestTag` condition key that requires users to tag any volumes they create with the tags `costcenter=115` and `stack=prod`. The `aws:TagKeys` condition key uses the `ForAllValues` modifier to indicate that only the keys `costcenter` and `stack` are allowed in the request (no other tags can be specified). If users don't pass these specific tags, or if they don't specify tags at all, the request fails.

For resource-creating actions that apply tags, users must also have permissions to use the `CreateTags` action. The second statement uses the `ec2:CreateAction` condition key to allow users to create tags only in the context of `CreateVolume`. Users cannot tag existing volumes or any other resources. For more information, see [Resource-Level Permissions for Tagging \(p. 599\)](#).

```
{  
    "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"
                }
            }
        }
    ]
}
```

Working with Snapshots

Examples

- [Example: Creating a Snapshot \(p. 606\)](#)
- [Example: Creating a Snapshot with Tags \(p. 607\)](#)
- [Example: Modifying Permission Settings for Snapshots \(p. 609\)](#)

Example: 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",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/size": "lessThan20GiB"
                }
            }
        }
    ]
}
```

```

    "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
    "Condition": {
        "NumericLessThan": {
            "ec2:VolumeSize": "20"
        },
        "Bool": {
            "ec2:Encrypted": "true"
        }
    }
}
]
}

```

Example: Creating a Snapshot with Tags

The following policy includes the `aws:RequestTag` condition key that requires the customer to apply the tags `costcenter=115` and `stack=prod` to any new snapshot. The `aws:TagKeys` condition key uses the `ForAllValues` modifier to indicate that only the keys `costcenter` and `stack` 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: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"
                }
            }
        }
    ]
}
```

Example: Modifying Permission Settings for Snapshots

The following policy allows modification of a snapshot only if the snapshot is tagged with User:*username*, where *username* is the customer's AWS account user name. The request fails if this condition is not met.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2: ModifySnapshotAttribute",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "aws:TagKeys": "stack"
                }
            }
        }
    ]
}
```

```
        "Condition":{  
            "StringEquals":{  
                "ec2:ResourceTag/user-name":"${aws:username}"  
            }  
        }  
    }  
}
```

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 a VPC requires a subnet, and creates a network interface. Launching from an Amazon EBS-backed AMI creates a volume. Therefore, the user must have permissions to use these Amazon EC2 resources. You can create a policy statement that requires users to specify an optional parameter on RunInstances, or restricts users to particular values for a parameter.

For more information about the resource-level permissions that are required to launch an instance, see [Resource-Level Permissions for RunInstances \(p. 595\)](#).

By default, users don't have permissions to describe, start, stop, or terminate the resulting instances. One way to grant the users permission to manage the resulting instances is to create a specific tag for each instance, and then create a statement that enables them to manage instances with that tag. For more information, see [Working with Instances \(p. 602\)](#).

Resources

- [AMIs \(p. 610\)](#)
- [Instance Types \(p. 611\)](#)
- [Subnets \(p. 612\)](#)
- [EBS Volumes \(p. 613\)](#)
- [Tags \(p. 614\)](#)
- [Tags in a Launch Template \(p. 617\)](#)
- [Elastic GPUs \(p. 617\)](#)
- [Launch Templates \(p. 618\)](#)

AMIs

The following policy allows users to launch instances using only the specified AMIs, `ami-9e1670f7` and `ami-45cf5c3c`. The users can't launch an instance using other AMIs (unless another statement grants the users permission to do so).

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": "ec2:RunInstances",  
        "Resource": [  
            "arn:aws:ec2:region::image/ami-9e1670f7",  
            "arn:aws:ec2:region::image/ami-45cf5c3c",  
            "arn:aws:ec2:region:account:instance/*",  
            "arn:aws:ec2:region:account:volume/*",  
            "arn:aws:ec2:region:account:key-pair/*",  
            "arn:aws:ec2:region:account:security-group/*",  
            "arn:aws:ec2:region:account:subnet/*",  
            "arn:aws:ec2:region:account:network-interface/*"  
        ]  
    }]
```

```
        }
    ]
}
```

Alternatively, the following policy allows users to launch instances from all AMIs owned by Amazon. The Condition element of the first statement tests whether ec2:Owner is amazon. The users can't launch an instance using other AMIs (unless another statement grants the users permission to do so).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:region::image/ami-*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "ec2:Owner": "amazon"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:region:account:instance/*",  
                "arn:aws:ec2:region:account:subnet/*",  
                "arn:aws:ec2:region:account:volume/*",  
                "arn:aws:ec2:region:account:network-interface/*",  
                "arn:aws:ec2:region:account:key-pair/*",  
                "arn:aws:ec2:region:account:security-group/*"  
            ]  
        }  
    ]  
}
```

Instance Types

The following policy allows users to launch instances using only the t2.micro or t2.small instance type, which you might do to control costs. The users can't launch larger instances because the Condition element of the first statement tests whether ec2:InstanceType is either t2.micro or t2.small.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:region:account:instance/*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "ec2:InstanceType": ["t2.micro", "t2.small"]  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:region:account:instance/*"  
            ]  
        }  
    ]  
}
```

```
        "arn:aws:ec2:region::image/ami-*",
        "arn:aws:ec2:region:account:subnet/*",
        "arn:aws:ec2:region:account:network-interface/*",
        "arn:aws:ec2:region:account:volume/*",
        "arn:aws:ec2:region:account:key-pair/*",
        "arn:aws:ec2:region:account:security-group/*"
    ]
}
]
```

Alternatively, you can create a policy that denies users permissions to launch any instances except `t2.micro` and `t2.small` instance types.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region:account:instance/*"
            ],
            "Condition": {
                "StringNotEquals": {
                    "ec2:InstanceType": ["t2.micro", "t2.small"]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region::image/ami-*",
                "arn:aws:ec2:region:account:network-interface/*",
                "arn:aws:ec2:region:account:instance/*",
                "arn:aws:ec2:region:account:subnet/*",
                "arn:aws:ec2:region:account:volume/*",
                "arn:aws:ec2:region:account:key-pair/*",
                "arn:aws:ec2:region:account:security-group/*"
            ]
        }
    ]
}
```

Subnets

The following policy allows users to launch instances using only the specified subnet, `subnet-12345678`. The group can't launch instances into any another subnet (unless another statement grants the users permission to do so).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region:account:subnet/subnet-12345678",
                "arn:aws:ec2:region:account:network-interface/*",
                "arn:aws:ec2:region:account:instance/*",
                "arn:aws:ec2:region:account:volume/*",
                "arn:aws:ec2:region::image/ami-*",
                "arn:aws:ec2:region:account:key-pair/*",
                "arn:aws:ec2:region:account:security-group/*"
            ]
        }
    ]
}
```

```
        }
    ]
}
```

Alternatively, you could create a policy that denies users permissions to launch an instance into any other subnet. The statement does this by denying permission to create a network interface, except where subnet subnet-12345678 is specified. This denial overrides any other policies that are created to allow launching instances into other subnets.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region:account:network-interface/*"
            ],
            "Condition": {
                "ArnNotEquals": {
                    "ec2:Subnet": "arn:aws:ec2:region:account:subnet/subnet-12345678"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region::image/ami-*",
                "arn:aws:ec2:region:account:network-interface/*",
                "arn:aws:ec2:region:account:instance/*",
                "arn:aws:ec2:region:account:subnet/*",
                "arn:aws:ec2:region:account:volume/*",
                "arn:aws:ec2:region:account:key-pair/*",
                "arn:aws:ec2:region:account:security-group/*"
            ]
        }
    ]
}
```

EBS Volumes

The following policy allows users to launch instances only if the EBS volumes for the instance are encrypted. The user must launch an instance from an AMI that was created with encrypted snapshots, to ensure that the root volume is encrypted. Any additional volume that the user attaches to the instance during launch must also be encrypted.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:***:volume/*"
            ],
            "Condition": {
                "Bool": {
                    "ec2:Encrypted": "true"
                }
            }
        },
    ]
}
```

```
{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
        "arn:aws:ec2::::image/ami-*",
        "arn:aws:ec2::::network-interface/*",
        "arn:aws:ec2::::instance/*",
        "arn:aws:ec2::::subnet/*",
        "arn:aws:ec2::::key-pair/*",
        "arn:aws:ec2::::security-group/*"
    ]
}
]
```

Tags

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 \(p. 599\)](#).

```
{
    "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": {
                    "aws:RequestTag/purpose": "test",
                    "ec2:CreateAction" : "RunInstances"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "purpose"
                }
            }
        }
    ]
}
```

Tags in a Launch Template

In the following example, users can launch instances, but only if they use a specific launch template (`lt-09477bcd97b0d310e`). The `ec2: IsLaunchTemplateResource` condition key prevents users from overriding any of the resources specified in the launch template. The second part of the statement allows users to tag instances on creation—this part of the statement is necessary if tags are specified for the instance in the launch template.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": "*",  
            "Condition": {  
                "ArnLike": {  
                    "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/  
lt-09477bcd97b0d310e"  
                },  
                "Bool": {  
                    "ec2:IsLaunchTemplateResource": "true"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2>CreateTags"  
            ],  
            "Resource": "arn:aws:ec2:region:account:instance/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:CreateAction" : "RunInstances"  
                }  
            }  
        }  
    ]  
}
```

Elastic GPUs

In the following policy, users can launch an instance and specify an elastic GPU to attach to the instance. Users can launch instances in any region, but they can only attach an elastic GPU during a launch in the `us-east-2` region.

The `ec2:ElasticGpuType` condition key uses the `ForAnyValue` modifier to indicate that only the elastic GPU types `eg1.medium` and `eg1.large` are allowed in the request.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances"  
            ],  
            "Resource": [  
                "arn:aws:ec2:*:account:elastic-gpu/*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ElasticGpuType": "eg1.medium" || "eg1.large"  
                }  
            }  
        }  
    ]  
}
```

```

        "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"
        }
    }
}
]
}

```

Example: Working with Reserved Instances

The following policy gives users permission to view, modify, and purchase Reserved Instances in your account.

It is not possible to set resource-level permissions for individual Reserved Instances. This policy means that users have access to all the Reserved Instances in the account.

The Resource element uses a * wildcard to indicate that users can specify all resources with the action; in this case, they can list and modify all Reserved Instances in the account. They can also purchase Reserved Instances using the account credentials. The * wildcard is also necessary in cases where the API action does not support resource-level permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",
                "ec2:PurchaseReservedInstancesOffering", "ec2:DescribeAvailabilityZones",
                "ec2:DescribeReservedInstancesOfferings"
            ],
            "Resource": "*"
        }
    ]
}
```

To allow users to view and modify the Reserved Instances in your account, but not purchase new Reserved Instances.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",
                "ec2:DescribeAvailabilityZones"
            ],
            "Resource": "*"
        }
    ]
}
```

Example: Tagging Resources

The following policy allows users to use the `CreateTags` action to apply tags to an instance only if the tag contains the key `environment` and the value `production`. The `ForAllValues` modifier is used with the `aws:TagKeys` condition key to indicate that only the key `environment` is allowed in the request (no other tags are allowed). The user cannot tag any other resource types.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags"  
            ],  
            "Resource": "arn:aws:ec2:region:account:instance/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/environment": "production"  
                },  
                "ForAllValues:StringEquals": {  
                    "aws:TagKeys": [  
                        "environment"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

The following policy allows users to tag any taggable resource that already has a tag with a key of `owner` and a value of the IAM username. In addition, users must specify a tag with a key of `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"]  
                }  
            }  
        }  
    ]  
}
```

Example: Working with IAM Roles

The following policy allows users to attach, replace, and detach an IAM role to instances that have the tag `department=test`. Replacing or detaching an IAM role requires an association ID, therefore the policy also grants users permission to use the `ec2:DescribeIamInstanceProfileAssociations` action.

IAM users must have permission to use the `iam:PassRole` action in order to pass the role to the instance.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:AssociateIamInstanceProfile",  
                "ec2:ReplaceIamInstanceProfileAssociation",  
                "ec2:DetachIamInstanceProfile",  
                "ec2:DescribeIamInstanceProfileAssociations",  
                "iam:PassRole"  
            ],  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/department": "test"  
                }  
            }  
        }  
    ]  
}
```

```
"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 \(p. 633\)](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AssociateIamInstanceProfile",
                "ec2:ReplaceIamInstanceProfileAssociation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeIamInstanceProfileAssociations",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::account:role/TestRole-*"
        }
    ]
}
```

Example: Working with Route Tables

The following policy allows users to add, remove, and replace routes for route tables that are associated with VPC `vpc-ec43eb89` only. To specify a VPC for the `ec2:Vpc` condition key, you must specify the full ARN of the VPC.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "ec2:DeleteRoute",
            "ec2:CreateRoute",
            "ec2:ReplaceRoute"
        ],
        "Resource": [
            "arn:aws:ec2:region:account:route-table/*"
        ],
        "Condition": {
            "StringEquals": {
                "ec2:Vpc": "arn:aws:ec2:region:account:vpc/vpc-ec43eb89"
            }
        }
    }
]
```

Example: Allowing a Specific Instance to View Resources in Other AWS Services

The following is an example of a policy that you might attach to an IAM role. The policy allows an instance to view resources in various AWS services. It uses the `ec2:SourceInstanceARN` condition key to specify that the instance from which the request is made must be instance `i-093452212644b0dd6`. If the same IAM role is associated with another instance, the other instance cannot perform any of these actions.

The `ec2:SourceInstanceARN` key is an AWS-wide condition key, therefore it can be used for other service actions, not just Amazon EC2.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVolumes",
                "s3>ListAllMyBuckets",
                "dynamodb>ListTables",
                "rds:DescribeDBInstances"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "ArnEquals": {
                    "ec2:SourceInstanceARN": "arn:aws:ec2:region:account:instance/
i-093452212644b0dd6"
                }
            }
        }
    ]
}
```

Example: Working with Launch Templates

The following policy allows users to create a launch template version and modify a launch template, but only for a specific launch template (`lt-09477bcd97b0d3abc`). Users cannot work with other launch templates.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "ec2:CreateLaunchTemplateVersion",  
                "ec2:ModifyLaunchTemplate"  
            ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:ec2:region:account:launch-template/lt-09477bcd97b0d3abc"  
        }  
    ]  
}
```

The following policy allows users to delete any launch template and launch template version, provided that the launch template has the tag Purpose=Testing.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "ec2>DeleteLaunchTemplate",  
                "ec2>DeleteLaunchTemplateVersions"  
            ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:ec2:region:account:launch-template/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/Purpose": "Testing"  
                }  
            }  
        }  
    ]  
}
```

Example Policies for Working in the Amazon EC2 Console

You can use IAM policies to grant users permissions to view and work with specific resources in the Amazon EC2 console. You can use the example policies in the previous section; however, they are designed for requests that are made with the AWS CLI or an AWS SDK. The console uses additional API actions for its features, so these policies may not work as expected. For example, a user that has permission to use only the `DescribeVolumes` API action will encounter errors when trying to view volumes in the console. This section demonstrates policies that enable users to work with specific parts of the console.

Tip

To help you work out which API actions are required to perform tasks in the console, you can use a service such as AWS CloudTrail. For more information, see the [AWS CloudTrail User Guide](#).

If your policy does not grant permission to create or modify a specific resource, the console displays an encoded message with diagnostic information. You can decode the message using the `DecodeAuthorizationMessage` API action for AWS STS, or the `decode-authorization-message` command in the AWS CLI.

Examples

- [Example: Read-Only Access \(p. 626\)](#)
- [Example: Using the EC2 Launch Wizard \(p. 627\)](#)
- [Example: Working with Volumes \(p. 629\)](#)

- [Example: Working with Security Groups \(p. 630\)](#)
- [Example: Working with Elastic IP Addresses \(p. 631\)](#)
- [Example: Working with Reserved Instances \(p. 632\)](#)

For additional information about creating policies for the Amazon EC2 console, see the following AWS Security Blog post: [Granting Users Permission to Work in the Amazon EC2 Console](#).

Example: Read-Only Access

To allow users to view all resources in the Amazon EC2 console, you can use the same policy as the following example: [Example: Read-Only Access \(p. 601\)](#). Users cannot perform any actions on those resources or create new resources, unless another statement grants them permission to do so.

View instances, AMIs, and snapshots

Alternatively, you can provide read-only access to a subset of resources. To do this, replace the * wildcard in the ec2:Describe API action with specific ec2:Describe actions for each resource. The following policy allows users to view all instances, AMIs, and snapshots in the Amazon EC2 console. The ec2:DescribeTags action allows users to view public AMIs. The console requires the tagging information to display public AMIs; however, you can remove this action to allow users to view only private AMIs.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": [  
            "ec2:DescribeInstances", "ec2:DescribeImages",  
            "ec2:DescribeTags", "ec2:DescribeSnapshots"  
        ],  
        "Resource": "*"  
    }]  
}
```

Note

The Amazon EC2 ec2:Describe* API actions do not support resource-level permissions, so you cannot control which individual resources users can view in the console. Therefore, the * wildcard is necessary in the Resource element of the above statement. For more information about which ARNs you can use with which Amazon EC2 API actions, see [Supported Resource-Level Permissions for Amazon EC2 API Actions \(p. 573\)](#).

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": "*"  
    }]
```

```
        "Resource": "*"
    }
]
```

Example: Using the EC2 Launch Wizard

The Amazon EC2 launch wizard is a series of screens with options to configure and launch an instance. Your policy must include permission to use the API actions that allow users to work with the wizard's options. If your policy does not include permission to use those actions, some items in the wizard cannot load properly, and users cannot complete a launch.

Basic launch wizard access

To complete a launch successfully, users must be given permission to use the `ec2:RunInstances` API action, and at least the following API actions:

- `ec2:DescribeImages`: To view and select an AMI.
- `ec2:DescribeVpcs`: To view the available network options.
- `ec2:DescribeSubnets`: To view all available subnets for the chosen VPC.
- `ec2:DescribeSecurityGroups`: 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`: To view and select a specific Availability Zone.
- `ec2:DescribeNetworkInterfaces`: 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 \(p. 599\)](#). 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.

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 [Launching Instances \(RunInstances\) \(p. 610\)](#). 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"
                }
            }
        }
    ]
}
```

Example: Working with Volumes

The following policy grants users permission to view and create volumes, and attach and detach volumes to specific instances.

Users can attach any volume to instances that have the tag "purpose=test", and also detach volumes from those instances. To attach a volume using the Amazon EC2 console, it is helpful for users to have permission to use the `ec2:DescribeInstances` action, as this allows them to select an instance from a pre-populated list in the **Attach Volume** dialog box. However, this also allows users to view all instances on the **Instances** page in the console, so you can omit this action.

In the first statement, the `ec2:DescribeAvailabilityZones` action is necessary to ensure that a user can select an Availability Zone when creating a volume.

Users cannot tag the volumes that they create (either during or after volume creation).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVolumes",
                "ec2:DescribeAvailabilityZones",
                "ec2>CreateVolume",
                "ec2:DescribeInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:region:111122223333:instance/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/purpose": "test"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ]
        }
    ]
}
```

```
        ],
        "Resource": "arn:aws:ec2:region:111122223333:volume/*"
    }
}
```

Example: Working with Security Groups

View security groups and add and remove rules

The following policy grants users permission to view security groups in the Amazon EC2 console, and to add and remove inbound and outbound rules for existing security groups that have the tag `Department=Test`.

In the first statement, the `ec2:DescribeTags` action allows users to view tags in the console, which makes it easier for users to identify the security groups that they are allowed to modify.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeSecurityGroups", "ec2:DescribeTags"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupIngress", "ec2:RevokeSecurityGroupIngress",
                "ec2:AuthorizeSecurityGroupEgress", "ec2:RevokeSecurityGroupEgress"
            ],
            "Resource": [
                "arn:aws:ec2:region:111122223333:security-group/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Department": "Test"
                }
            }
        }
    ]
}
```

Working with the Create Security Group dialog box

You can create a policy that allows users to work with the **Create Security Group** dialog box in the Amazon EC2 console. To use this dialog box, users must be granted permission to use at least the following API actions:

- `ec2:CreateSecurityGroup`: To create a new security group.
- `ec2:DescribeVpcs`: To view a list of existing VPCs in the **VPC** list.

With these permissions, users can create a new security group successfully, but they cannot add any rules to it. To work with rules in the **Create Security Group** dialog box, you can add the following API actions to your policy:

- `ec2:AuthorizeSecurityGroupIngress`: To add inbound rules.
- `ec2:AuthorizeSecurityGroupEgress`: To add outbound rules to VPC security groups.

- **ec2:RevokeSecurityGroupIngress:** To modify or delete existing inbound rules. This is useful to allow users to use the **Copy to new** feature in the console. This feature opens the **Create Security Group** dialog box and populates it with the same rules as the security group that was selected.
- **ec2:RevokeSecurityGroupEgress:** To modify or delete outbound rules for VPC security groups. This is useful to allow users to modify or delete the default outbound rule that allows all outbound traffic.
- **ec2>DeleteSecurityGroup:** To cater for when invalid rules cannot be saved. The console first creates the security group, and then adds the specified rules. If the rules are invalid, the action fails, and the console attempts to delete the security group. The user remains in the **Create Security Group** dialog box so that they can correct the invalid rule and try to create the security group again. This API action is not required, but if a user is not granted permission to use it and attempts to create a security group with invalid rules, the security group is created without any rules, and the user must add them afterward.

Currently, the `ec2:CreateSecurityGroup` API action does not support resource-level permissions; however, you can apply resource-level permissions to the `ec2:AuthorizeSecurityGroupIngress` and `ec2:AuthorizeSecurityGroupEgress` actions to control how users can create rules.

The following policy grants users permission to use the **Create Security Group** dialog box, and to create inbound and outbound rules for security groups that are associated with a specific VPC (`vpc-1a2b3c4d`). Users can create security groups for EC2-Classic or another VPC, but they cannot add any rules to them. Similarly, users cannot add any rules to any existing security group that's not associated with VPC `vpc-1a2b3c4d`. Users are also granted permission to view all security groups in the console. This makes it easier for users to identify the security groups to which they can add inbound rules. This policy also grants users permission to delete security groups that are associated with VPC `vpc-1a2b3c4d`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": [  
            "ec2:DescribeSecurityGroups", "ec2:CreateSecurityGroup", "ec2:DescribeVpcs"  
        ],  
        "Resource": "*"  
    },  
    {  
        "Effect": "Allow",  
        "Action": [  
            "ec2>DeleteSecurityGroup", "ec2:AuthorizeSecurityGroupIngress",  
            "ec2:AuthorizeSecurityGroupEgress"  
        ],  
        "Resource": "arn:aws:ec2:region:111122223333:security-group/*",  
        "Condition": {  
            "ArnEquals": {  
                "ec2:Vpc": "arn:aws:ec2:region:111122223333:vpc/vpc-1a2b3c4d"  
            }  
        }  
    }  
]
```

Example: Working with Elastic IP Addresses

To allow users to view Elastic IP addresses in the Amazon EC2 console, you must grant users permission to use the `ec2:DescribeAddresses` action.

To allow users to work with Elastic IP addresses, you can add the following actions to your policy.

- **ec2:AllocateAddress:** To allocate an Elastic IP address.

- `ec2:ReleaseAddress`: To release an Elastic IP address.
- `ec2:AssociateAddress`: To associate an Elastic IP address with an instance or a network interface.
- `ec2:DescribeNetworkInterfaces` and `ec2:DescribeInstances`: To work with the **Associate address** screen. The screen displays the available instances or network interfaces to which you can associate an Elastic IP address.
- `ec2:DisassociateAddress`: To disassociate an Elastic IP address from an instance or a network interface.

The following policy allows users to view, allocate, and associate Elastic IP addresses with instances. Users cannot associate Elastic IP addresses with network interfaces, disassociate Elastic IP addresses, or release them.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeAddresses",  
                "ec2:AllocateAddress",  
                "ec2:DescribeInstances",  
                "ec2:AssociateAddress"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Example: Working with Reserved Instances

The following policy can be attached to an IAM user. It gives the user access to view and modify Reserved Instances in your account, as well as purchase new Reserved Instances in the AWS Management Console.

This policy allows users to view all the Reserved Instances, as well as On-Demand Instances, in the account. It's not possible to set resource-level permissions for individual Reserved Instances.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",  
                "ec2:PurchaseReservedInstancesOffering", "ec2:DescribeInstances",  
                "ec2: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 \(p. 573\)](#) and the following example: [Example: Working with IAM Roles \(p. 622\)](#).

Topics

- [Instance Profiles \(p. 633\)](#)
- [Retrieving Security Credentials from Instance Metadata \(p. 634\)](#)
- [Granting an IAM User Permission to Pass an IAM Role to an Instance \(p. 634\)](#)
- [Working with IAM Roles \(p. 635\)](#)

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 \(p. 452\)](#).

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"  
            ]  
        }  
    ]  
}
```

```
        "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 \(p. 635\)](#)
- [Launching an Instance with an IAM Role \(p. 637\)](#)
- [Attaching an IAM Role to an Instance \(p. 638\)](#)
- [Detaching an IAM Role \(p. 639\)](#)
- [Replacing an IAM Role \(p. 640\)](#)

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 (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"
        }
    ]
}
```

```
        "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": "AROAIIZKPBKS2LEXAMPLE",
            "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 (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 (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 (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 (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": "AIPAJLNLDX3AMYZNWYYAY",  
            "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 (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 (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 (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 (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 Windows 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 RDP. 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 \(p. 551\)](#) and [Custom Security Groups \(p. 552\)](#). 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 RDP traffic from your computer's public IPv4 address. To allow RDP 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 Linux instance, see [Authorizing Inbound Traffic for Your Linux Instances](#) in the *Amazon EC2 User Guide for Linux 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 RDP. 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 Windows Instances \(p. 548\)](#).

Windows Firewall may also block incoming traffic. If you're having trouble setting up access to your instance, you may have to disable Windows Firewall. For more information, see [Remote Desktop can't connect to the remote computer \(p. 986\)](#).

Adding a Rule for Inbound RDP Traffic to a Windows 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 Windows instance from your IP address using RDP.

To add a rule to a security group for inbound RDP traffic over IPv4 (console)

1. In the navigation pane of the Amazon EC2 console, choose **Instances**. Select your instance and look at the **Description** tab; **Security groups** lists the security groups that are associated with the instance. Choose **view inbound rules** to display a list of the rules that are in effect for the instance.
2. In the navigation pane, choose **Security Groups**. Select one of the security groups associated with your instance.
3. In the details pane, on the **Inbound** tab, choose **Edit**. In the dialog, choose **Add Rule**, and then choose **RDP** 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 \(p. 641\)](#).

5. Choose **Save**.

If you launched an instance with an IPv6 address and want to connect to your instance using its IPv6 address, you must add rules that allow inbound IPv6 traffic over RDP.

To add a rule to a security group for inbound RDP traffic over IPv6 (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Security Groups**. Select the security group for your instance.
3. Choose **Inbound, Edit, Add Rule**.
4. For **Type**, choose **RDP**.
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 3389 --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="3389"; ToPort="3389";  
IpRanges="cidr_ip_range" }
```

```
PS C:\> Grant-EC2SecurityGroupIngress -GroupId security_group_id -IpPermission @($ip1)
```

Assigning a Security Group to an Instance

You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group.

After you launch an instance, you can change its security groups. For more information, see [Changing an Instance's Security Groups](#) in the *Amazon VPC User Guide*.

Amazon EC2 Instance IP Addressing

Amazon EC2 and Amazon VPC support both the IPv4 and IPv6 addressing protocols. By default, Amazon EC2 and Amazon VPC use the IPv4 addressing protocol; you can't disable this behavior. When you create a VPC, you must specify an IPv4 CIDR block (a range of private IPv4 addresses). You can optionally assign an IPv6 CIDR block to your VPC and subnets, and assign IPv6 addresses from that block to instances in your subnet. IPv6 addresses are reachable over the Internet. For more information about IPv6, see [IP Addressing in Your VPC](#) in the *Amazon VPC User Guide*.

Contents

- [Private IPv4 Addresses and Internal DNS Hostnames \(p. 643\)](#)
- [Public IPv4 Addresses and External DNS Hostnames \(p. 644\)](#)
- [Elastic IP Addresses \(IPv4\) \(p. 645\)](#)
- [Amazon DNS Server \(p. 645\)](#)
- [IPv6 Addresses \(p. 645\)](#)
- [Working with IP Addresses for Your Instance \(p. 646\)](#)
- [Multiple IP Addresses \(p. 650\)](#)

Private IPv4 Addresses and Internal DNS Hostnames

A private IPv4 address is an IP address that's not reachable over the Internet. You can use private IPv4 addresses for communication between instances in the same VPC. For more information about the standards and specifications of private IPv4 addresses, see [RFC 1918](#). We allocate private IPv4 addresses to instances using DHCP.

Note

You can create a VPC with a publicly routable CIDR block that falls outside of the private IPv4 address ranges specified in RFC 1918. However, for the purposes of this documentation, we refer to private IPv4 addresses (or 'private IP addresses') as the IP addresses that are within the IPv4 CIDR range of your VPC.

When you launch an instance, we allocate a primary private IPv4 address for the instance. Each instance is also given an internal DNS hostname that resolves to the primary private IPv4 address; for example, `ip-10-251-50-12.ec2.internal`. You can use the internal DNS hostname for communication between instances in the same network, but we can't resolve the DNS hostname outside the network that the instance is in.

An instance 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 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 \(p. 650\)](#).

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.

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 a default VPC, we assign it a public IP address by default. When you launch an instance into a nondefault VPC, the subnet has an attribute that determines whether instances launched into that subnet receive a public IP address from the public IPv4 address pool. By default, we don't assign a public IP address to instances launched in a nondefault subnet.

You can control whether your instance receives a public IP address as follows:

- Modifying the public IP addressing attribute of your subnet. For more information, see [Modifying the Public IPv4 Addressing Attribute for Your Subnet](#) in the *Amazon VPC User Guide*.
- Enabling or disabling the public IP addressing feature during launch, which overrides the subnet's public IP addressing attribute. For more information, see [Assigning a Public IPv4 Address During Instance Launch \(p. 647\)](#).

A public IP address is assigned to your instance from Amazon's pool of public IPv4 addresses, and is not associated with your AWS account. When a public IP address is disassociated from your instance, it is released back into the public IPv4 address pool, and you cannot reuse it.

You cannot manually associate or disassociate a public IP address from your instance. Instead, in certain cases, we release the public IP address from your instance, or assign it a new one:

- We release your instance's public IP address when it is stopped or terminated. Your stopped instance receives a new public IP address when it is restarted.
- We release your instance's public IP address when you associate an Elastic IP address with it. When you disassociate the Elastic IP address from your instance, it receives a new public IP address.
- If the public IP address of your instance in a VPC has been released, it will not receive a new one if there is more than one network interface attached to your instance.
- If your instance's public IP address is released while it has a secondary private IP address that is associated with an Elastic IP address, the instance does not receive a new public IP address.

If you require a persistent public IP address that can be associated to and from instances as you require, use an Elastic IP address instead.

If you use dynamic DNS to map an existing DNS name to a new instance's public IP address, it might take up to 24 hours for the IP address to propagate through the Internet. As a result, new instances might not receive traffic while terminated instances continue to receive requests. To solve this problem, use an

Elastic IP address. You can allocate your own Elastic IP address, and associate it with your instance. For more information, see [Elastic IP Addresses \(p. 660\)](#).

If you assign an Elastic IP address to an instance, it receives an IPv4 DNS hostname if DNS hostnames are enabled. For more information, see [Using DNS with Your VPC](#) in the *Amazon VPC User Guide*.

Note

Instances that access other instances through their public NAT IP address are charged for regional or Internet data transfer, depending on whether the instances are in the same region.

Elastic IP Addresses (IPv4)

An Elastic IP address is a public IPv4 address that you can allocate to your account. You can associate it to and 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 \(p. 660\)](#).

We do not support Elastic IP addresses for IPv6.

Amazon DNS Server

Amazon provides a DNS server that resolves Amazon-provided IPv4 DNS hostnames to IPv4 addresses. The Amazon DNS server is located at the base of your VPC network range plus two. For more information, see [Amazon DNS Server](#) in the *Amazon VPC User Guide*.

IPv6 Addresses

You can optionally associate an IPv6 CIDR block with your VPC, and associate IPv6 CIDR blocks with your subnets. The IPv6 CIDR block for your VPC is automatically assigned from Amazon's pool of IPv6 addresses; you cannot choose the range yourself. For more information, see the following topics in the *Amazon VPC User Guide*:

- [VPC and Subnet Sizing for IPv6](#)
- [Associating an IPv6 CIDR Block with Your VPC](#)
- [Associating an IPv6 CIDR Block with Your Subnet](#)

IPv6 addresses are globally unique, and therefore reachable over the Internet. Your instance receives an IPv6 address if an IPv6 CIDR block is associated with your VPC and subnet, and if one of the following is true:

- Your subnet is configured to automatically assign an IPv6 address to an instance during launch. For more information, see [Modifying the IPv6 Addressing Attribute for Your Subnet](#).
- You assign an IPv6 address to your instance during launch.
- You assign an IPv6 address to the primary network interface of your instance after launch.
- You assign an IPv6 address to a network interface in the same subnet, and attach the network interface to your instance after launch.

When your instance receives an IPv6 address during launch, the address is associated with the primary network interface (eth0) of the instance. You can disassociate the IPv6 address from the network interface. We do not support IPv6 DNS hostnames for your instance.

An IPv6 address persists when you stop and start 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 \(p. 667\)](#).

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 \(p. 646\)](#)
- [Determining Your IPv6 Addresses \(p. 647\)](#)
- [Assigning a Public IPv4 Address During Instance Launch \(p. 647\)](#)
- [Assigning an IPv6 Address to an Instance \(p. 648\)](#)
- [Unassigning an IPv6 Address From an Instance \(p. 649\)](#)

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 \(p. 452\)](#).

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. 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. 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. 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:

```
PS C:\> Invoke-RestMethod http://169.254.169.254/latest/meta-data/local-ipv4
```

3. Use the following command to access the public IP address:

```
PS C:\> Invoke-RestMethod 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

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/`):

```
PS C:\> Invoke-RestMethod http://169.254.169.254/latest/meta-data/network/interfaces/`macs/`mac-address`/ipv6s
```

Assigning a Public IPv4 Address During Instance Launch

Each subnet has an attribute that determines whether instances launched into that subnet are assigned a public IP address. By default, nondefault subnets have this attribute set to false, and default subnets have this attribute set to true. When you launch an instance, a public IPv4 addressing feature is also available for you to control whether your instance is assigned a public IPv4 address; you can override the default behavior of the subnet's IP addressing attribute. The public IPv4 address is assigned from Amazon's pool of public IPv4 addresses, and is assigned to the network interface with the device index of `eth0`. This feature depends on certain conditions at the time you launch your instance.

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 \(p. 644\)](#). 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 \(p. 660\)](#).

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 \(p. 315\)](#). 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 \(p. 660\)](#). 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.

IPv6 is supported on all current generation instance types and the C3, R3, and I2 previous generation instance types.

To assign an IPv6 address to an instance during launch

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

2. Select an AMI and an instance type that supports IPv6, and choose **Next: Configure Instance Details**.
3. On the **Configure Instance Details** page, for **Network**, select a VPC and for **Subnet**, select a subnet. For **Auto-assign IPv6 IP**, choose **Enable**.
4. Follow the remaining steps in the wizard to launch your instance.

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

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 \(p. 667\)](#).

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 \(p. 650\)](#)
- [Working with Multiple IPv4 Addresses \(p. 651\)](#)
- [Working with Multiple IPv6 Addresses \(p. 654\)](#)

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.
- You associate security groups with network interfaces, not the individual IP addresses. Therefore, each IP address you specify in a network interface is subject to the security group of its network interface.
- Multiple IP addresses can be assigned and unassigned to network interfaces attached to running or stopped instances.
- Secondary private IPv4 addresses that are assigned to a network interface can be reassigned to another one if you explicitly allow it.
- An IPv6 address cannot be reassigned to another network interface; you must first unassign the IPv6 address from the existing network interface.
- When assigning multiple IP addresses to a network interface using the command line tools or API, the entire operation fails if one of the IP addresses can't be assigned.
- Primary private IPv4 addresses, secondary private IPv4 addresses, Elastic IP addresses, and IPv6 addresses remain with 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 \(p. 651\)](#)
- [Configuring the Operating System on Your Instance to Recognize the Secondary Private IPv4 Address \(p. 653\)](#)
- [Associating an Elastic IP Address with the Secondary Private IPv4 Address \(p. 653\)](#)
- [Viewing Your Secondary Private IPv4 Addresses \(p. 653\)](#)
- [Unassigning a Secondary Private IPv4 Address \(p. 654\)](#)

Assigning a Secondary Private IPv4 Address

You can assign the secondary private IPv4 address to the network interface for an instance as you launch the instance, or after the instance is running. This section includes the following procedures.

- [To assign a secondary private IPv4 address when launching an instance \(p. 651\)](#)
- [To assign a secondary IPv4 address during launch using the command line \(p. 652\)](#)
- [To assign a secondary private IPv4 address to a network interface \(p. 652\)](#)
- [To assign a secondary private IPv4 to an existing instance using the command line \(p. 652\)](#)

To assign a secondary private IPv4 address when launching an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. Select an AMI, then choose an instance type and choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, for **Network**, select a VPC and for **Subnet**, select a subnet.
5. In the **Network Interfaces** section, do the following, and then choose **Next: Add Storage**:
 - To add another network interface, choose **Add Device**. The console enables you to specify up to two network interfaces when you launch an instance. After you launch the instance, choose **Network Interfaces** in the navigation pane to add additional network interfaces. The total number of network interfaces that you can attach varies by instance type. For more information, see [IP Addresses Per Network Interface Per Instance Type \(p. 667\)](#).

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 \(p. 661\)](#).

- 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 \(p. 653\)](#).

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.

For information about configuring a Windows instance, see [Configuring a Secondary Private IPv4 Address for Your Windows Instance \(p. 443\)](#).

Associating an Elastic IP Address with the Secondary Private IPv4 Address

To associate an Elastic IP address with a secondary private IPv4 address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Actions**, and then select **Associate address**.
4. For **Network interface**, select the network interface, and then select the secondary IP address from the **Private IP** list.
5. Choose **Associate**.

To associate an Elastic IP address with a secondary private IPv4 address using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).
 - [associate-address](#) (AWS CLI)
 - [Register-EC2Address](#) (AWS Tools for Windows PowerShell)

Viewing Your Secondary Private IPv4 Addresses

To view the private IPv4 addresses assigned to a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface with private IP addresses to view.
4. On the **Details** tab in the details pane, check the **Primary private IPv4 IP** and **Secondary private IPv4 IPs** fields for the primary private IPv4 address and any secondary private IPv4 addresses assigned to the network interface.

To view the private IPv4 addresses assigned to an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instance with private IPv4 addresses to view.
4. On the **Description** tab in the details pane, check the **Private IPs** and **Secondary private IPs** fields for the primary private IPv4 address and any secondary private IPv4 addresses assigned to the instance through its network interface.

Unassigning a Secondary Private IPv4 Address

If you no longer require a secondary private IPv4 address, you can unassign it from the instance or the network interface. When a secondary private IPv4 address is unassigned from a network interface, the Elastic IP address (if it exists) is also disassociated.

To unassign a secondary private IPv4 address from an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select an instance, choose **Actions, Networking, Manage IP Addresses**.
4. Under **IPv4 Addresses**, choose **Unassign** for the IPv4 address to unassign.
5. Choose **Yes, Update**.

To unassign a secondary private IPv4 address from a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select the network interface, choose **Actions, Manage IP Addresses**.
4. Under **IPv4 Addresses**, choose **Unassign** for the IPv4 address to unassign.
5. Choose **Yes, Update**.

To unassign a secondary private IPv4 address using the command line

- You can use one of the following commands. For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).
 - `unassign-private-ip-addresses` (AWS CLI)
 - `Unregister-EC2PrivateIpAddress` (AWS Tools for Windows PowerShell)

Working with Multiple IPv6 Addresses

You can assign multiple IPv6 addresses to your instance, view the IPv6 addresses assigned to your instance, and unassign IPv6 addresses from your instance.

Contents

- [Assigning Multiple IPv6 Addresses \(p. 654\)](#)
- [Viewing Your IPv6 Addresses \(p. 656\)](#)
- [Unassigning an IPv6 Address \(p. 656\)](#)

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 \(p. 110\)](#).
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 \(p. 667\)](#).
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 \(p. 556\)](#). 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 in the Amazon EC2 console to assign multiple IPv6 addresses to an existing instance. This assigns the IPv6 addresses to the primary network interface (eth0) for the instance. To assign a specific IPv6 address to the instance, ensure that the IPv6 address is not already assigned to another instance or network interface.

To assign multiple IPv6 addresses to an existing instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance, choose **Actions, Networking, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Assign new IP** for each IPv6 address you want to add. You can specify an IPv6 address from the range of the subnet, or leave the **Auto-assign** value to let Amazon choose an IPv6 address for you.
5. Choose **Yes, Update**.

Alternatively, you can assign multiple IPv6 addresses to an existing network interface. The network interface must have been created in a subnet that has an associated IPv6 CIDR block. To assign a specific IPv6 address to the network interface, ensure that the IPv6 address is not already assigned to another network interface.

To assign multiple IPv6 addresses to a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select your network interface, choose **Actions, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Assign new IP** for each IPv6 address you want to add. You can specify an IPv6 address from the range of the subnet, or leave the **Auto-assign** value to let Amazon choose an IPv6 address for you.

5. Choose **Yes, Update**.

CLI Overview

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

- **Assign an IPv6 address during launch:**
 - Use the `--ipv6-addresses` or `--ipv6-address-count` options with the [run-instances](#) command (AWS CLI)
 - Define `-NetworkInterface` and specify the `Ipv6Addresses` or `Ipv6AddressCount` parameters with the [New-EC2Instance](#) command (AWS Tools for Windows PowerShell).
- **Assign an IPv6 address to a network interface:**
 - [assign-ipv6-addresses](#) (AWS CLI)
 - [Register-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell)

Viewing Your IPv6 Addresses

You can view the IPv6 addresses for an instance or for a network interface.

To view the IPv6 addresses assigned to an instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select your instance. In the details pane, review the **IPv6 IPs** field.

To view the IPv6 addresses assigned to a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select your network interface. In the details pane, review the **IPv6 IPs** field.

CLI Overview

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

- **View the IPv6 addresses for an instance:**
 - [describe-instances](#) (AWS CLI)
 - [Get-EC2Instance](#) (AWS Tools for Windows PowerShell).
- **View the IPv6 addresses for a network interface:**
 - [describe-network-interfaces](#) (AWS CLI)
 - [Get-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

Unassigning an IPv6 Address

You can unassign an IPv6 address from the primary network interface of an instance, or you can unassign an IPv6 address from a network interface.

To unassign an IPv6 address from an instance

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

2. In the navigation pane, choose **Instances**.
3. Select your instance, choose **Actions, Networking, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Unassign** for the IPv6 address to unassign.
5. Choose **Yes, Update**.

To unassign an IPv6 address from a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Network Interfaces**.
3. Select your network interface, choose **Actions, Manage IP Addresses**.
4. Under **IPv6 Addresses**, choose **Unassign** for the IPv6 address to unassign.
5. Choose **Save**.

CLI Overview

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

- [unassign-ipv6-addresses](#) (AWS CLI)
- [Unregister-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell).

Bring Your Own IP Addresses (BYOIP)

You can bring part or all of your public IPv4 address range from your on-premises network to your AWS account. You continue to own the address range, but AWS advertises it on the internet. After you bring the address range to AWS, it appears in your account as an address pool. You can create an Elastic IP address from your address pool and use it with your AWS resources, such as EC2 instances, NAT gateways, and Network Load Balancers.

Important

BYOIP is not available in all Regions. For a list of supported Regions, see the [FAQ for Bring Your Own IP](#).

Requirements

- The address range must be registered with your regional internet registry (RIR), such as the American Registry for Internet Numbers (ARIN) or Réseaux IP Européens Network Coordination Centre (RIPE). It must be registered to a business or institutional entity and may not be registered to an individual person.
- For ARIN, the supported network types are "Direct Allocation" and "Direct Assignment".
- For RIPE, the supported allocation statuses are "ALLOCATED PA", "LEGACY", and "ASSIGNED PI".
- The most specific address range that you can specify is /24.
- You can bring each address range to one region at a time.
- You can bring 5 address ranges per region to your AWS account.
- The addresses in the IP address range must have a clean history. We may investigate the reputation of the IP address range and reserve the right to reject an IP address range if it contains an IP address that has poor reputation or is associated with malicious behavior.

Prepare to Bring Your Address Range to Your AWS Account

To ensure that only you can bring your address range to your AWS account, you must authorize Amazon to advertise the address range and provide proof that you own the address range.

A Route Origin Authorization (ROA) is a document that you can create through your RIR. It contains the address range, the ASNs that are allowed to advertise the address range, and an expiration date. An ROA authorizes Amazon to advertise an address range under a specific AS number. However, it does not authorize your AWS account to bring the address range to AWS. To authorize your AWS account to bring an address range to AWS, you must publish a self-signed X509 certificate in the RDAP remarks for the address range. The certificate contains a public key, which AWS uses to verify the authorization-context signature that you provide. You should keep your private key secure and use it to sign the authorization-context message.

The commands in the following procedure require OpenSSL version 1.0.2 or later.

To prepare to bring your address range to your AWS account

1. Create an ROA to authorize Amazon ASNs 16509 and 14618 to advertise your address range. You must set the maximum length to the size of the smallest prefix that you want to bring (for example, /24). It might take up to 24 hours for the ROA to become available to Amazon. For more information, see the following:
 - ARIN — [ROA Requests](#)
 - RIPE — [Managing ROAs](#)
2. Generate an RSA 2048-bit key pair as follows:

```
openssl genrsa -out private.key 2048
```

3. Create a public X509 certificate from the key pair using the following command. In this example, the certificate expires in 365 days, after which time it cannot be trusted. Therefore, be sure to set the expiration appropriately. When prompted for information, you can accept the default values.

```
openssl req -new -x509 -key private.key -days 365 | tr -d "\n" > publickey.cer
```

4. Create a signed message. The format of the message is as follows, where the date is the expiry date of the message:

```
1 | aws | account | cidr | YYYMMDD | SHA256 | RSAPSS
```

The following command signs the message using the key pair that you created and saves it as `base64_urlsafe_signature`:

```
echo "1|aws|123456789012|198.51.100.0/24|20191201|SHA256|RSAPSS" | tr -d "\n" | openssl dgst -sha256 -sigopt rsa_padding_mode:pss -sigopt rsa_pss_saltlen:-1 -sign private.key -keyform PEM | openssl base64 | tr -- '+=/ '-_-' | tr -d "\n" > base64_urlsafe_signature
```

5. Update the RDAP record for your RIR with the X509 certificate. Be sure to copy the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- from the certificate. Be sure that you have removed newline characters, if you haven't already done so using the `tr -d "\n"` commands in the previous steps. To view your certificate, run the following command:

```
cat publickey.cer
```

For ARIN, add the certificate in the "Public Comments" section for your address range.

For RIPE, add the certificate as a new "desc" field for your address range.

Provision the Address Range for use with AWS

When you provision an address range for use with AWS, you are confirming that you own the address range and authorizing Amazon to advertise it. We also verify that you own the address range.

To provision the address range, use the following [provision-byoip-cidr](#) command. The message in the --cidr-authorization-context parameter is the signed message that you created in the previous section, not the ROA message.

```
aws ec2 provision-byoip-cidr --cidr address-range --cidr-authorization-context  
Message="message",Signature="signature"
```

Provisioning an address range is an asynchronous operation, so the call returns immediately, but the address range is not ready to use until its status changes from pending-provision to provisioned. To monitor the status of the address range, use the following [describe-byoip-cidrs](#) command:

```
aws ec2 describe-byoip-cidrs --max-results 5
```

Advertise the Address Range through AWS

After the address range is provisioned, it is ready to be advertised. We recommend that you stop advertising the address range from other locations before you advertise it through AWS. If you keep advertising your IP address range from other locations, we can't reliably support it or troubleshoot issues. Specifically, we can't guarantee that traffic to the address range will enter our network.

To minimize down time, you can configure your AWS resources to use an address from your address pool before it is advertised, and then simultaneously stop advertising it from the current location and start advertising it through AWS. For more information about allocating an Elastic IP address from your address pool, see [Allocating an Elastic IP Address \(p. 661\)](#).

To advertise the address range, use the following [advertise-byoip-cidr](#) command:

```
aws ec2 advertise-byoip-cidr --cidr address-range
```

Important

You can run the [advertise-byoip-cidr](#) command at most once every 10 seconds, even if you specify different address ranges each time.

To stop advertising the address range, use the following [withdraw-byoip-cidr](#) command:

```
aws ec2 withdraw-byoip-cidr --cidr address-range
```

Important

You can run the [withdraw-byoip-cidr](#) command at most once every 10 seconds, even if you specify different address ranges each time.

Deprovision the Address Range

To stop using your address range with AWS, release any Elastic IP addresses still allocated from the address pool, stop advertising the address range, and deprovision the address range.

To release each Elastic IP address, use the following [release-address](#) command:

```
aws ec2 release-address --allocation-id eipalloc-12345678
```

To stop advertising the address range, use the following [withdraw-byoip-cidr](#) command:

```
aws ec2 withdraw-byoip-cidr --cidr address-range
```

To deprovision the address range, use the following [deprovision-byoip-cidr](#) command:

```
aws ec2 deprovision-byoip-cidr --cidr address-range
```

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 \(p. 660\)](#)
- [Working with Elastic IP Addresses \(p. 661\)](#)
- [Using Reverse DNS for Email Applications \(p. 665\)](#)
- [Elastic IP Address Limit \(p. 665\)](#)

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, and you cannot convert a public IPv4 address to an Elastic IP address. For more information, see [Public IPv4 Addresses and External DNS Hostnames \(p. 644\)](#).
- 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.
- When you allocate an Elastic IP address from an IP address pool that you have brought to your AWS account, it does not count toward your Elastic IP address limits.

Working with Elastic IP Addresses

The following sections describe how you can work with Elastic IP addresses.

Tasks

- [Allocating an Elastic IP Address \(p. 661\)](#)
- [Describing Your Elastic IP Addresses \(p. 662\)](#)
- [Tagging an Elastic IP Address \(p. 662\)](#)
- [Associating an Elastic IP Address with a Running Instance \(p. 663\)](#)
- [Disassociating an Elastic IP Address and Reassociating with a Different Instance \(p. 663\)](#)
- [Releasing an Elastic IP Address \(p. 664\)](#)
- [Recovering an Elastic IP Address \(p. 664\)](#)

Allocating an Elastic IP Address

You can allocate an Elastic IP address from Amazon's pool of public IPv4 addresses, or from a custom IP address pool that you have brought to your AWS account. For more information about bringing your own IP address range to your AWS account, see [Bring Your Own IP Addresses \(BYOIP\) \(p. 657\)](#).

You can allocate an Elastic IP address using the Amazon EC2 console or the command line.

To allocate an Elastic IP address from Amazon's pool of 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 **Elastic IPs**.
3. Choose **Allocate new address**.
4. For **IPv4 address pool**, choose **Amazon pool**.
5. Choose **Allocate**, and close the confirmation screen.

To allocate an Elastic IP address from an IP address pool that you own 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. For **IPv4 address pool**, choose **Owned by me** and then select the IP address pool.

To see the IP address range of the selected address pool and the number of IP addresses already allocated from the address pool, see **Address ranges**.

5. For **IPv4 address**, do one of the following:

- To let Amazon EC2 select an IP address from the address pool, choose **No preference**.
 - To select a specific IP address from the address pool, choose **Select an address** and then type the IP address.
6. Choose **Allocate**, and 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.

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.

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)

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).

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 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.
- 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

- (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
```

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

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 Windows \(p. 684\)](#).

Contents

- [Network Interface Basics \(p. 666\)](#)
- [IP Addresses Per Network Interface Per Instance Type \(p. 667\)](#)
- [Scenarios for Network Interfaces \(p. 673\)](#)
- [Best Practices for Configuring Network Interfaces \(p. 674\)](#)
- [Working with Network Interfaces \(p. 675\)](#)
- [Requester-Managed Network Interfaces \(p. 683\)](#)

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 \(p. 667\)](#).

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 \(p. 644\)](#).

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 \(p. 645\)](#).

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 \(p. 650\)](#). 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

Amazon Elastic Compute Cloud
 User Guide for Windows Instances
 IP Addresses Per Network Interface Per Instance Type

Instance Type	Maximum Network Interfaces	IPv4 Addresses per Interface	IPv6 Addresses per Interface
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
c5n.large	3	10	10
c5n.xlarge	4	15	15
c5n.2xlarge	4	15	15
c5n.4xlarge	8	30	30
c5n.9xlarge	8	30	30
c5n.18xlarge	15	50	50
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
d2.8xlarge	8	30	30
f1.2xlarge	4	15	15
f1.4xlarge	8	30	30
f1.16xlarge	8	50	50
g2.2xlarge	4	15	IPv6 not supported
g2.8xlarge	8	30	IPv6 not supported
g3s.xlarge	4	15	15
g3.4xlarge	8	30	30
g3.8xlarge	8	30	30
g3.16xlarge	15	50	50
h1.2xlarge	4	15	15

Amazon Elastic Compute Cloud
 User Guide for Windows Instances
 IP Addresses Per Network Interface Per Instance Type

Instance Type	Maximum Network Interfaces	IPv4 Addresses per Interface	IPv6 Addresses per Interface
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
m2.4xlarge	8	30	IPv6 not supported
m3.medium	2	6	IPv6 not supported
m3.large	3	10	IPv6 not supported
m3.xlarge	4	15	IPv6 not supported
m3.2xlarge	4	30	IPv6 not supported
m4.large	2	10	10
m4.xlarge	4	15	15
m4.2xlarge	4	15	15
m4.4xlarge	8	30	30

Amazon Elastic Compute Cloud
User Guide for Windows Instances
IP Addresses Per Network Interface Per Instance Type

Instance Type	Maximum Network Interfaces	IPv4 Addresses per Interface	IPv6 Addresses per Interface
m4.10xlarge	8	30	30
m4.16xlarge	8	30	30
m5.large	3	10	10
m5.xlarge	4	15	15
m5.2xlarge	4	15	15
m5.4xlarge	8	30	30
m5.12xlarge	8	30	30
m5.24xlarge	15	50	50
m5a.large	3	10	10
m5a.xlarge	4	15	15
m5a.2xlarge	4	15	15
m5a.4xlarge	8	30	30
m5a.12xlarge	8	30	30
m5a.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
p3dn.24xlarge	15	50	50
r3.large	3	10	10
r3.xlarge	4	15	15
r3.2xlarge	4	15	15

Amazon Elastic Compute Cloud
 User Guide for Windows Instances
 IP Addresses Per Network Interface Per Instance Type

Instance Type	Maximum Network Interfaces	IPv4 Addresses per Interface	IPv6 Addresses per Interface
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
r5a.large	3	10	10
r5a.xlarge	4	15	15
r5a.2xlarge	4	15	15
r5a.4xlarge	8	30	30
r5a.12xlarge	8	30	30
r5a.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	3	4	4

Instance Type	Maximum Network Interfaces	IPv4 Addresses per Interface	IPv6 Addresses per Interface
t2.medium	3	6	6
t2.large	3	12	12
t2.xlarge	3	15	15
t2.2xlarge	3	15	15
t3.nano	2	2	2
t3.micro	2	2	2
t3.small	3	4	4
t3.medium	3	6	6
t3.large	3	12	12
t3.xlarge	4	15	15
t3.2xlarge	4	15	15
u-6tb1.metal	5	30	30
u-9tb1.metal	5	30	30
u-12tb1.metal	5	30	30
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

Note

If f1.16xlarge, g3.16xlarge, h1.16xlarge, i3.16xlarge, and r4.16xlarge instances use more than 31 IPv4 or IPv6 addresses per interface, they cannot access the instance

metadata, VPC DNS, and Time Sync services from the 32nd IP address onwards. If access to these services is needed from all IP addresses on the interface, we recommend using a maximum of 31 IP addresses per interface.

Scenarios for Network Interfaces

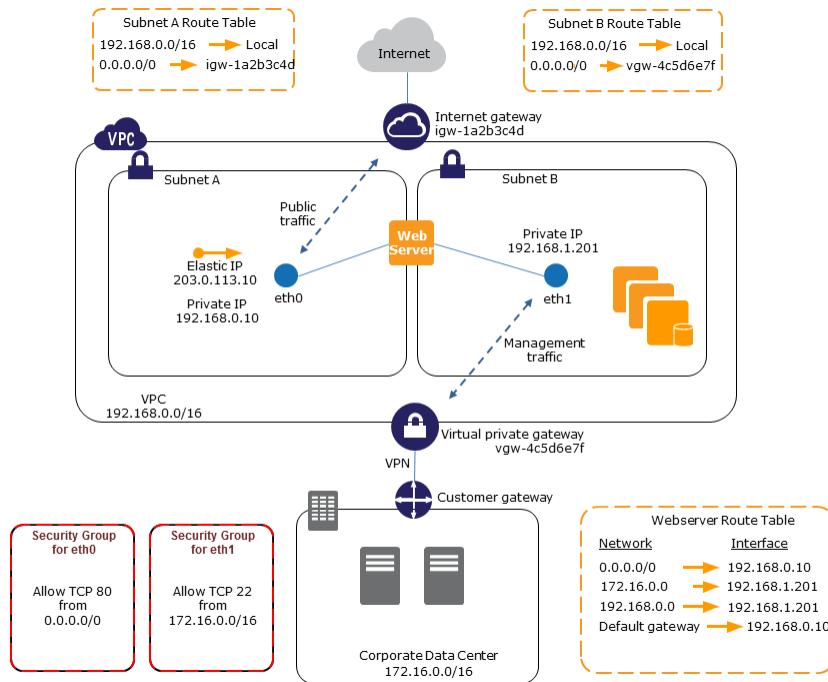
Attaching multiple network interfaces to an instance is useful when you want to:

- 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 RDP 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.
- If you have multiple subnets in an Availability Zone for the same VPC, you can move a network interface from an instance in one of these subnets to an instance in another one of these subnets.
- 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 \(p. 651\)](#). If you need to use multiple network interfaces, you must configure the network interfaces to use static routing. For more information, see [Configure a Secondary Elastic Network Interface \(p. 447\)](#).

Working with Network Interfaces

You can work with network interfaces using the Amazon EC2 console or the command line.

Contents

- [Creating a Network Interface \(p. 675\)](#)
- [Deleting a Network Interface \(p. 676\)](#)
- [Viewing Details about a Network Interface \(p. 676\)](#)
- [Attaching a Network Interface When Launching an Instance \(p. 677\)](#)
- [Attaching a Network Interface to a Stopped or Running Instance \(p. 678\)](#)
- [Detaching a Network Interface from an Instance \(p. 678\)](#)
- [Changing the Security Group \(p. 679\)](#)
- [Changing the Source or Destination Checking \(p. 679\)](#)
- [Associating an Elastic IP Address \(IPv4\) \(p. 680\)](#)
- [Disassociating an Elastic IP Address \(IPv4\) \(p. 680\)](#)
- [Assigning an IPv6 Address \(p. 681\)](#)
- [Unassigning an IPv6 Address \(p. 681\)](#)
- [Changing Termination Behavior \(p. 681\)](#)
- [Adding or Editing a Description \(p. 682\)](#)
- [Adding or Editing Tags \(p. 682\)](#)

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 \(p. 667\)](#).

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 \(p. 644\)](#).

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 \(p. 890\)](#).

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 \(p. 682\)](#).

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.

Enhanced Networking on Windows

Enhanced networking uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on [supported instance types \(p. 684\)](#). 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 \(p. 684\)](#)
- [Enabling Enhanced Networking on Your Instance \(p. 684\)](#)
- [Enabling Enhanced Networking with the Elastic Network Adapter \(ENA\) on Windows Instances \(p. 685\)](#)
- [Enabling Enhanced Networking with the Intel 82599 VF Interface on Windows Instances \(p. 688\)](#)
- [Operating System Optimizations \(p. 691\)](#)

Enhanced Networking Types

Depending on your instance type, enhanced networking can be enabled using one of the following mechanisms:

Elastic Network Adapter (ENA)

The Elastic Network Adapter (ENA) supports network speeds of up to 100 Gbps for supported instance types.

C5, C5d, C5n, F1, G3, H1, I3, m4.16xlarge, M5, M5a, M5d, P2, P3, R4, R5, R5a, R5d, T3, u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, X1, X1e, and z1d instances use the Elastic Network Adapter for enhanced networking.

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.

For information about the supported network speed for each instance type, see [Amazon EC2 Instance Types](#).

Enabling Enhanced Networking on Your Instance

If your instance type supports the Elastic Network Adapter for enhanced networking, follow the procedures in [Enabling Enhanced Networking with the Elastic Network Adapter \(ENA\) on Windows Instances \(p. 685\)](#).

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 Windows Instances \(p. 688\)](#).

Enabling Enhanced Networking with the Elastic Network Adapter (ENA) on Windows Instances

Amazon EC2 provides enhanced networking capabilities through the Elastic Network Adapter (ENA).

Contents

- [Requirements \(p. 685\)](#)
- [Testing Whether Enhanced Networking Is Enabled \(p. 685\)](#)
- [Enabling Enhanced Networking on Windows \(p. 686\)](#)
- [Amazon ENA Driver Versions \(p. 687\)](#)
- [Subscribing to Notifications \(p. 413\)](#)

Requirements

To prepare for enhanced networking using the ENA, set up your instance as follows:

- Select from the following supported instance types: C5, C5d, C5n, F1, G3, H1, I3, m4.16xlarge, M5, M5a, M5d, P2, P3, R4, R5, R5a, R5d, T3, u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, X1, X1e, and z1d.
- Ensure that the instance has internet connectivity.
- Install and configure the [AWS CLI](#) or the [AWS Tools for Windows PowerShell](#) on any computer you choose, preferably your local desktop or laptop. For more information, see [Accessing Amazon EC2 \(p. 3\)](#). Enhanced networking cannot be managed from the Amazon EC2 console.
- If you have important data on the instance that you want to preserve, you should back that data up now by creating an AMI from your instance. Updating kernels and kernel modules, as well as enabling the enaSupport attribute, might render incompatible instances or operating systems unreachable; if you have a recent backup, your data will still be retained if this happens.

Testing Whether Enhanced Networking Is Enabled

To test whether enhanced networking is already enabled, verify that the driver is installed on your instance and that the enaSupport attribute is set.

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
```

Enabling Enhanced Networking on Windows

If you launched your instance and it does not have enhanced networking enabled already, you must download and install the required network adapter driver on your instance, and then set the `enaSupport` instance attribute to activate enhanced networking. You can only enable this attribute on supported instance types and only if the ENA driver is installed. For more information, see [Enhanced Networking Types \(p. 684\)](#).

To enable enhanced networking

1. Connect to your instance and log in as the local administrator.
2. [Windows Server 2016] Run the following EC2Launch PowerShell script to configure the instance after the driver is installed.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -Schedule
```

3. From the instance, install the driver as follows:
 - a. [Download](#) the latest driver to the instance.
 - b. Extract the zip archive.
 - c. Install the driver by running the `install.ps1` PowerShell script.
4. 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.
5. Enable ENA support on your instance as follows:
 - a. From your local computer, check the EC2 instance ENA support attribute on your instance by running one of the following commands. If the attribute is not enabled, the output will be "[]" or blank. `EnaSupport` is set to `false` by default.
 - [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
```

- b. To enable ENA support, run one of the following commands:

- [modify-instance-attribute \(AWS CLI\)](#)

```
aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

- [Edit-EC2InstanceAttribute \(AWS Tools for Windows PowerShell\)](#)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -EnaSupport $true
```

If you encounter problems when you restart the instance, you can also disable ENA support using one of the following commands:

- [modify-instance-attribute \(AWS CLI\)](#)

```
aws ec2 modify-instance-attribute --instance-id instance_id --no-ena-support
```

- [Edit-EC2InstanceAttribute \(AWS Tools for Windows PowerShell\)](#)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -EnaSupport $false
```

- Verify that the attribute has been set to true using `describe-instances` or `Get-EC2Instance` as shown previously. You should now see the following output:

```
[  
    true  
]
```

- 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 using the AWS OpsWorks console so that the instance state remains in sync.
- On the instance, validate that the ENA driver is installed and enabled as follows:
 - Right-click the network icon and choose **Open Network and Sharing Center**.
 - Choose the Ethernet adapter (for example, **Ethernet 2**).
 - Choose **Details**. For **Network Connection Details**, check that **Description** is **Amazon Elastic Network Adapter**.
- (Optional) Create an AMI from the instance. The AMI inherits the `enaSupport` attribute from the instance. Therefore, you can use this AMI to launch another instance with ENA enabled by default. For more information, see [Creating a Custom Windows AMI \(p. 65\)](#).

Amazon ENA Driver Versions

Windows AMIs include the Amazon ENA driver to enable enhanced networking. The following table summarizes the changes for each release.

Driver version	Details	Release date
1.5.0	<ul style="list-style-type: none">Improved stability and performance fixes.Receive Buffers can now be configured up to a value of 8192 in Advanced Properties of the ENA NIC.Default Receive Buffers of 1k.	October 2018

Driver version	Details	Release date
1.2.3	Includes reliability fixes and unifies support for Windows Server 2008 R2 through Windows Server 2016.	February 2018
1.0.9	Includes some reliability fixes. Applies only to Windows Server 2008 R2. Not recommended for other versions of Windows Server.	December 2016
1.0.8	The initial release. Included in AMIs for Windows Server 2008 R2, Windows Server 2012 RTM, Windows Server 2012 R2, and Windows Server 2016.	July 2016

Subscribing to Notifications

Amazon SNS can notify you when new versions of EC2 Windows Drivers are released. Use the following procedure to subscribe to these notifications.

To subscribe to EC2 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 this region because the SNS notifications that you are subscribing to are in this region.
3. In the navigation pane, choose **Subscriptions**.
4. Choose **Create subscription**.
5. In the **Create subscription** dialog box, do the following:
 - a. For **TopicARN**, copy the following Amazon Resource Name (ARN):
`arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers`
 - b. For **Protocol**, choose `Email`.
 - c. For **Endpoint**, type an email address that you can use to receive the notifications.
 - d. Choose **Create subscription**.
6. You'll receive a confirmation email. Open the email and follow the directions to complete your subscription.

Whenever new EC2 Windows drivers are released, we send notifications to subscribers. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from Amazon EC2 Windows driver notification

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v2/home>.
2. In the navigation pane, choose **Subscriptions**.
3. Select the checkbox for the subscription and then choose **Actions, Delete subscriptions**. When prompted for confirmation, choose **Delete**.

Enabling Enhanced Networking with the Intel 82599 VF Interface on Windows Instances

Amazon EC2 provides enhanced networking capabilities through the Intel 82599 VF interface, which uses the Intel `ixgbevf` driver.

Contents

- [Requirements \(p. 689\)](#)
- [Testing Whether Enhanced Networking is Enabled \(p. 689\)](#)
- [Enabling Enhanced Networking on Windows \(p. 690\)](#)

Requirements

To prepare for enhanced networking using the Intel 82599 VF interface, set up your instance as follows:

- Select from the following supported instance types: C3, C4, D2, I2, M4 (excluding m4.16xlarge), and R3.
- Launch the instance from a 64-bit HVM AMI. You can't enable enhanced networking on Windows Server 2008 and Windows Server 2003. Enhanced networking is already enabled for Windows Server 2012 R2 and Windows Server 2016 AMIs. Windows Server 2012 R2 includes Intel driver 1.0.15.3 and we recommend that you upgrade that driver to the latest version using the Pnputil.exe utility.
- Ensure that the instance has internet connectivity.
- Install and configure the [AWS CLI](#) or the [AWS Tools for Windows PowerShell](#) on any computer you choose, preferably your local desktop or laptop. For more information, see [Accessing Amazon EC2 \(p. 3\)](#). Enhanced networking cannot be managed from the Amazon EC2 console.
- If you have important data on the instance that you want to preserve, you should back that data up now by creating an AMI from your instance. Updating kernels and kernel modules, as well as enabling the `sriovNetSupport` attribute, might render incompatible instances or operating systems unreachable; if you have a recent backup, your data will still be retained if this happens.

Testing Whether Enhanced Networking is Enabled

Enhanced networking with the Intel 82599 VF interface is enabled if the driver is installed on your instance and the `sriovNetSupport` attribute is set.

Driver

To verify that the driver is installed, connect to your instance and open Device Manager. You should see "Intel(R) 82599 Virtual Function" listed under **Network adapters**.

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 (srivNetSupport)

To check whether an AMI already has the enhanced networking `srivNetSupport` attribute set, use one of the following commands:

- [describe-image-attribute](#) (AWS CLI)

```
aws ec2 describe-image-attribute --image-id ami_id --attribute srivNetSupport
```

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 srivNetSupport
```

If the attribute isn't set, `SriovNetSupport` is empty; otherwise, it is set as follows:

```
"SriovNetSupport": {  
    "Value": "simple"  
},
```

Enabling Enhanced Networking on Windows

If you launched your instance and it does not have enhanced networking enabled already, you must download and install the required network adapter driver on your instance, and then set the `srivNetSupport` instance attribute to activate enhanced networking. You can only enable this attribute on supported instance types. For more information, see [Enhanced Networking Types \(p. 684\)](#).

Important

To view the latest version of the Intel driver in the Windows AMIs, see [Details About AWS Windows AMI Versions \(p. 81\)](#).

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 and log in as the local administrator.
2. [Windows Server 2016] Run the following EC2Launch PowerShell script to configure the instance after the driver is installed.

```
PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -  
Schedule
```

Important

The administrator password will reset when you enable the initialize instance EC2Launch script. You can modify the configuration file to disable the administrator password reset by specifying it in the settings for the initialization tasks. For steps on how to disable password reset, see [Configure Initialization Tasks \(p. 376\)](#).

3. From the instance, install the driver as follows:

- a. Download the Intel network adapter driver for your operating system:

- [Windows Server 2008 R2](#)
- [Windows Server 2012](#)

- [Windows Server 2012 R2](#)
 - [Windows Server 2016](#)
- b. In the **Download** folder, locate the PROWinx64.exe file. Rename this file PROWinx64.zip.
 - c. Open a context (right-click) menu on PROWinx64.zip and choose **Extract All**. Specify a destination path and choose **Extract**.
 - d. Open a command prompt window, go to the folder with the extracted files, and use the pnputil utility to add and install the INF file in the driver store.

Windows Server 2016

```
pnputil -i -a PROXGB\Winx64\NDIS65\vxn65x64.inf
```

Windows Server 2012 R2

```
pnputil -i -a PROXGB\Winx64\NDIS64\vxn64x64.inf
```

Windows Server 2012

```
pnputil -i -a PROXGB\Winx64\NDIS63\vxn63x64.inf
```

Windows Server 2008 R2

```
pnputil -a PROXGB\Winx64\NDIS62\vxn62x64.inf
```

4. 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.
 5. 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 --sriov-net-support simple
```
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)
- ```
Edit-EC2InstanceAttribute -InstanceId instance_id -SriovNetSupport "simple"
```
6. (Optional) Create an AMI from the instance, as described in [Creating a Custom Windows AMI \(p. 65\)](#). 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.
 7. 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.

Operating System Optimizations

To achieve the maximum network performance on instances with enhanced networking, you may need to modify the default operating system configuration. We recommend the following configuration

changes for applications that require high network performance. Other optimizations (such as turning on checksum offloading and enabling RSS, for example) are already in place on official Windows AMIs.

Note

TCP chimney offloading should be disabled in most use cases, and has been deprecated as of Windows Server 2016.

In addition to these operating system optimizations, you should also consider the maximum transmission unit (MTU) of your network traffic, and adjust according to your workload and network architecture. For more information, see [Network Maximum Transmission Unit \(MTU\) for Your EC2 Instance \(p. 697\)](#).

Configure RSS CPU Affinity

Receive side scaling (RSS) is used to distribute network traffic CPU load across multiple processors. By default, Amazon official Windows AMIs are configured with RSS enabled and allow up to eight RSS queues. By defining CPU affinity for RSS queues as well as other processes that demand CPU, it is possible to spread the CPU load on multi-core systems and allow more network traffic to be processed. To avoid contention when using multiple elastic network interfaces, use the **Set-NetAdapterRss** command to manually assign different sets of processors to handle RSS queues for different interfaces.

For example, in a 16-core instance with two elastic network interfaces, the following two commands spread the load evenly:

```
Set-NetAdapterRss -Name NIC1 -NumaNode 0 -BaseProcessorNumber 1 -MaxProcessorNumber 8
Set-NetAdapterRss -Name NIC2 -NumaNode 0 -BaseProcessorNumber 9 -MaxProcessorNumber 16
```

For more information, see [Set-NetAdapterRss](#).

Set Device Processor Affinity

Hardware devices need access to the CPU to raise interrupts and communicate with the host OS. In the event of heavy network traffic becoming CPU bound, it may be possible to smooth out the load by allocating separate cores to network devices, other hardware devices, and applications (such as SQL) running on the OS.

Configure interrupt affinity on the ENA devices to ensure that interrupt requests are all handled by the same NUMA node (setting `DevicePolicy` to `0x01` or `0x04`). For more information, see [Interrupt Affinity](#) in the Microsoft documentation. When using multiple elastic network interfaces, we recommend that you configure them to use different NUMA groups, to further spread out the IRQ load.

To minimize performance bottlenecks that are caused by processor interrupts, it is important to consider configuration changes for other sources of load on the machine. For example, [SQL Server may be configured to bind affinity to certain cores](#). When spreading load, keep in mind that inter-node NUMA communication is much more expensive than intra-node. In a hypothetical server with 64 processors distributed into two NUMA nodes, two elastic network interfaces, a web server, and a SQL database, it's optimal to divide CPUs up logically, rather than allowing the operating system to handle the distribution.

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 \(p. 693\)](#)
- [Spread Placement Groups \(p. 693\)](#)
- [Placement Group Rules and Limitations \(p. 693\)](#)
- [Creating a Placement Group \(p. 694\)](#)
- [Launching Instances in a Placement Group \(p. 695\)](#)
- [Changing the Placement Group for an Instance \(p. 695\)](#)
- [Deleting a Placement Group \(p. 696\)](#)

Cluster Placement Groups

A cluster placement group is a logical grouping of instances within a single Availability Zone. A placement group can span peered VPCs in the same region. The chief benefit of a cluster placement group, in addition to a 10 Gbps flow limit, is the non-blocking, non-oversubscribed, fully bi-sectional nature of the connectivity. In other words, all nodes within the placement group can talk to all other nodes within the placement group at the full line rate of 10 Gbps flows and 25 aggregate without any slowing due to over-subscription.

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 \(p. 684\)](#).

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.
- For instances that are enabled for enhanced networking, traffic between instances within the same region that is addressed using IPv4 or IPv6 addresses can use up to 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.

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, M5a, and M5d
 - Compute optimized: C3, C4, C5, C5d, C5n, and cc2.8xlarge
 - Memory optimized: cr1.8xlarge, R3, R4, R5, R5a, R5d, X1, X1e, and z1d
 - Storage optimized: D2, H1, hs1.8xlarge, I2, and I3
 - Accelerated computing: F1, G2, G3, P2, and 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 network connectivity that meets your requirements.
- For instances that are enabled for enhanced networking, the following rules apply:
 - Instances within a cluster placement group can use up to 10 Gbps for single-flow traffic.
 - 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 a Custom Windows AMI \(p. 65\)](#).

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 \(p. 695\)](#) 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 \(9001 MTU\) \(p. 697\)](#)
- [Path MTU Discovery \(p. 698\)](#)
- [Check the Path MTU Between Two Hosts \(p. 698\)](#)
- [Check and Set the MTU on Your Windows Instance \(p. 699\)](#)
- [Troubleshooting \(p. 700\)](#)

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 \(p. 692\)](#).

You can use jumbo frames for traffic between your VPCs and your on-premises networks over AWS Direct Connect. For more information, see [Setting Network MTU](#) in the *AWS Direct Connect User Guide*.

The following instances support jumbo frames:

- General purpose: M3, M4, M5, M5a, M5d, T2, and T3
- Compute optimized: C3, C4, C5, C5d, C5n, and CC2
- Memory optimized: CR1, R3, R4, R5, R5a, R5d, X1, and z1d
- Storage optimized: D2, H1, HS1, I2, and I3
- Accelerated computing: F1, G2, G3, P2, and P3
- Bare metal: i3.metal, u-6tb1.metal, u-9tb1.metal, and u-12tb1.metal

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 [Rules for Path MTU Discovery \(p. 559\)](#).

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 **mturoute.exe** command, which you can download and install from <http://www.elifulkerson.com/projects/mturoute.php>.

To check path MTU using mturoute.exe

1. Download **mturoute.exe** from <http://www.elifulkerson.com/projects/mturoute.php>.
2. Open a Command Prompt window and change to the directory where you downloaded **mturoute.exe**.
3. 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 www.elifulkerson.com.

```
.\\mturoute.exe www.elifulkerson.com
* ICMP Fragmentation is not permitted. *
* Speed optimization is enabled. *
* Maximum payload is 10000 bytes. *
+ ICMP payload of 1472 bytes succeeded.
- ICMP payload of 1473 bytes is too big.
Path MTU: 1500 bytes.
```

In this example, the path MTU is 1500.

Check and Set the MTU on Your Windows Instance

Some drivers are configured to use jumbo frames, and others are configured to use standard frame sizes. You might want to use jumbo frames for network traffic within your VPC or standard frames for Internet traffic. Whatever your use case, we recommend that you verify that your instances behave as expected.

ENet Driver

You can change the MTU setting using Device Manager or the **Set-NetAdapterAdvancedProperty** command.

To get the current MTU setting using the **Get-NetAdapterAdvancedProperty** command, use the following command. Check the entry for the interface name MTU. A value of 9001 indicates that Jumbo frames are enabled. Jumbo frames are disabled by default.

```
Get-NetAdapterAdvancedProperty -Name "Ethernet"
```

Enable jumbo frames as follows:

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "MTU" -RegistryValue 9001
```

Disable jumbo frames as follows:

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "MTU" -RegistryValue 1500
```

Intel SRIOV 82599 Driver

You can change the MTU setting using Device Manager or the **Set-NetAdapterAdvancedProperty** command.

To get the current MTU setting using the **Get-NetAdapterAdvancedProperty** command, use the following command. Check the entry for the interface name *JumboPacket. A value of 9014 indicates that Jumbo frames are enabled. (Note that the MTU size includes the header and the payload.) Jumbo frames are disabled by default.

```
Get-NetAdapterAdvancedProperty -Name "Ethernet"
```

Enable jumbo frames as follows:

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" - RegistryValue 9014
```

Disable jumbo frames as follows:

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" - RegistryValue 1514
```

AWS PV Driver

You cannot change the MTU setting using Device Manager, but you can change it using the **netsh** command.

Get the current MTU setting using the following command. The name of the interface can vary. In the output, look for an entry with the name "Ethernet," "Ethernet 2," or "Local Area Connection". You'll need the interface name to enable or disable jumbo frames. A value of 9001 indicates that Jumbo frames are enabled.

```
netsh interface ipv4 show subinterface
```

Enable jumbo frames as follows:

```
netsh interface ipv4 set subinterface "Ethernet" mtu=9001
```

Disable jumbo frames as follows:

```
netsh interface ipv4 set subinterface "Ethernet" mtu=1500
```

Troubleshooting

If you experience connectivity issues between your EC2 instance and an Amazon Redshift cluster when using jumbo frames, see [Queries Appear to Hang](#) in the *Amazon Redshift Cluster Management Guide*

Virtual Private Clouds

Amazon Virtual Private Cloud (Amazon VPC) enables you to define a virtual network in your own logically isolated area within the AWS cloud, known as a *virtual private cloud (VPC)*. You can launch your Amazon EC2 resources, such as instances, into the subnets of your VPC. Your VPC closely resembles a traditional network that you might operate in your own data center, with the benefits of using scalable infrastructure from AWS. You can configure your VPC; you can select its IP address range, create subnets, and configure route tables, network gateways, and security settings. You can connect instances in your VPC to the internet or to your own data center.

When you create your AWS account, we create a *default VPC* for you in each region. A default VPC is a VPC that is already configured and ready for you to use. You can launch instances into your default VPC immediately. Alternatively, you can create your own *nondefault VPC* and configure it as you need.

If you created your AWS account before 2013-12-04, you might have support for the EC2-Classic platform in some regions. If you created your AWS account after 2013-12-04, it does not support EC2-Classic, so you must launch your resources in a VPC. For more information, see [EC2-Classic \(p. 701\)](#).

Amazon VPC Documentation

For more information about Amazon VPC, see the following documentation.

Guide	Description
Amazon VPC User Guide	Describes key concepts and provides instructions for using the features of Amazon VPC.
Amazon VPC Peering Guide	Describes VPC peering connections and provides instructions for using them.
Amazon VPC Network Administrator Guide	Helps network administrators configure customer gateways.

EC2-Classic

With EC2-Classic, your instances run in a single, flat network that you share with other customers. With Amazon VPC, your instances run in a virtual private cloud (VPC) that's logically isolated to your AWS account.

The EC2-Classic platform was introduced in the original release of Amazon EC2. If you created your AWS account after 2013-12-04, it does not support EC2-Classic, so you must launch your Amazon EC2 instances in a VPC.

If your account does not support EC2-Classic, we create a default VPC for you. By default, when you launch an instance, we launch it into your default VPC. Alternatively, you can create a nondefault VPC and specify it when you launch an instance.

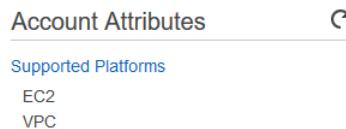
Detecting Supported Platforms

The Amazon EC2 console indicates which platforms you can launch instances into for the selected region, and whether you have a default VPC in that region.

Verify that the region you'll use is selected in the navigation bar. On the Amazon EC2 console dashboard, look for **Supported Platforms** under **Account Attributes**.

Accounts that Support EC2-Classic

The dashboard displays the following under **Account Attributes** to indicate that the account supports both the EC2-Classic platform and VPCs in this region, but the region does not have a default VPC.



The output of the [describe-account-attributes](#) command includes both the EC2 and VPC values for the `supported-platforms` attribute.

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
{
    "AccountAttributes": [
        {
            "AttributeName": "supported-platforms",
            "AttributeValues": [
                {
                    "AttributeValue": "EC2"
                },
                {
                    "AttributeValue": "VPC"
                }
            ]
        }
    ]
}
```

Accounts that Require a VPC

The dashboard displays the following under **Account Attributes** to indicate that the account requires a VPC to launch instances in this region, does not support the EC2-Classic platform in this region, and the region has a default VPC with the identifier `vpc-1a2b3c4d`.

Account Attributes

C^a

Supported Platforms

VPC

Default VPC

vpc-1a2b3c4d

The output of the [describe-account-attributes](#) command includes only the VPC value for the supported-platforms attribute.

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
{
    "AccountAttributes": [
        {
            "AttributeValues": [
                {
                    "AttributeValue": "VPC"
                }
            ],
            "AttributeName": "supported-platforms",
        }
    ]
}
```

Instance Types Available in EC2-Classic

Most of the newer instance types require a VPC. The following are the only instance types supported in EC2-Classic:

- General purpose: M1, M3, and T1
- Compute optimized: C1, C3, and CC2
- Memory optimized: CR1, M2, and R3
- Storage optimized: D2, HS1, and I2
- Accelerated computing: G2

If your account supports EC2-Classic but you have not created a nondefault VPC, you can do one of the following to launch instances that require a VPC:

- Create a nondefault VPC and launch your VPC-only instance into it by specifying a subnet ID or a network interface ID in the request. Note that you must create a nondefault VPC if you do not have a default VPC and you are using the AWS CLI, Amazon EC2 API, or AWS SDK to launch a VPC-only instance. For more information, see [Create a Virtual Private Cloud \(VPC\) \(p. 16\)](#).
- 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 a VPC. For more information, see [Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC \(p. 721\)](#).

Differences Between Instances in EC2-Classic and a VPC

The following table summarizes the differences between instances launched in EC2-Classic, instances launched in a default VPC, and instances launched in a nondefault VPC.

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Public IPv4 address (from Amazon's public IP address pool)	Your instance receives a public IPv4 address from the EC2-Classic public IPv4 address pool.	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.
Associating an Elastic IP address	You associate an Elastic IP address with an instance.	An Elastic IP address is a property of a network interface. You associate an Elastic IP address with an instance by updating the network interface attached to the instance.	An Elastic IP address is a property of a network interface. You associate an Elastic IP address with an instance by updating the network interface attached to the instance.
Reassociating an Elastic IP address	If the Elastic IP address is already associated with another instance, the address is automatically associated with the new instance.	If the Elastic IP address is already associated with another instance, the address is automatically associated with the new instance.	If the Elastic IP address is already associated with another instance, it succeeds only if you allowed reassociation.
Tagging Elastic IP addresses	You cannot apply tags to an Elastic IP address.	You can apply tags to an Elastic IP address.	You can apply tags to an Elastic IP address.
DNS hostnames	DNS hostnames are enabled by default.	DNS hostnames are enabled by default.	DNS hostnames are disabled by default.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Differences Between Instances in EC2-Classic and a VPC

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
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.
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.

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
IPv6 addressing	IPv6 addressing is not supported. You cannot assign IPv6 addresses to your instances.	You can optionally associate an IPv6 CIDR block with your VPC, and assign IPv6 addresses to instances in your VPC.	You can optionally associate an IPv6 CIDR block with your VPC, and assign IPv6 addresses to instances in your VPC.

Security Groups for EC2-Classic

If you're using EC2-Classic, you must use security groups created specifically for EC2-Classic. When you launch an instance in EC2-Classic, you must specify a security group in the same region as the instance. You can't specify a security group that you created for a VPC when you launch an instance in EC2-Classic.

After you launch an instance in EC2-Classic, you can't change its security groups. However, you can add rules to or remove rules from a security group, and those changes are automatically applied to all instances that are associated with the security group after a short period.

Your AWS account automatically has a default security group per region for EC2-Classic. If you try to delete the default security group, you'll get the following error: Client.InvalidGroup.Reserved: The security group 'default' is reserved.

You can create custom security groups. The security group name must be unique within your account for the region. To create a security group for use in EC2-Classic, choose **No VPC** for the VPC.

You can add inbound rules to your default and custom security groups. You can't change the outbound rules for an EC2-Classic security group. When you create a security group rule, you can use a different security group for EC2-Classic in the same region as the source or destination. To specify a security group for another AWS account, add the AWS account ID as a prefix; for example, 111122223333/sg-edcd9784.

In EC2-Classic, you can have up to 500 security groups in each region for each account. You can associate an instance with up to 500 security groups and add up to 100 rules to a security group.

IP Addressing and DNS

Amazon provides a DNS server that resolves Amazon-provided IPv4 DNS hostnames to IPv4 addresses. In EC2-Classic, the Amazon DNS server is located at 172.16.0.23.

If you create a custom firewall configuration in EC2-Classic, you must create a rule in your firewall that allows inbound traffic from port 53 (DNS)—with a destination port from the ephemeral range—from the address of the Amazon DNS server; otherwise, internal DNS resolution from your instances fails. If your firewall doesn't automatically allow DNS query responses, then you need to allow traffic from the IP address of the Amazon DNS server. To get the IP address of the Amazon DNS server, use the following command from within your instance:

```
ipconfig /all | findstr /c:"DNS Servers"
```

Elastic IP Addresses

If your account supports EC2-Classic, there's one pool of Elastic IP addresses for use with the EC2-Classic platform and another for use with your VPCs. You can't associate an Elastic IP address that you allocated for use with a VPC with an instance in EC2-Classic, and vice-versa. However, you can migrate an Elastic IP address you've allocated for use in the EC2-Classic platform for use with a VPC. You cannot migrate an Elastic IP address to another region.

To allocate an Elastic IP address for use in EC2-Classic using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate new address**.
4. Select **Classic**, and then choose **Allocate**. Close the confirmation screen.

Migrating an Elastic IP Address from EC2-Classic

If your account supports EC2-Classic, you can migrate Elastic IP addresses that you've allocated for use with EC2-Classic platform to be used with a VPC, within the same region. This can assist you to migrate your resources from EC2-Classic to a VPC; for example, you can launch new web servers in your VPC, and then use the same Elastic IP addresses that you used for your web servers in EC2-Classic for your new VPC web servers.

After you've migrated an Elastic IP address to a VPC, you cannot use it with EC2-Classic. However, if required, you can restore it to EC2-Classic. You cannot migrate an Elastic IP address that was originally allocated for use with a VPC to EC2-Classic.

To migrate an Elastic IP address, it must not be associated with an instance. For more information about disassociating an Elastic IP address from an instance, see [Disassociating an Elastic IP Address and Reassociating with a Different Instance \(p. 663\)](#).

You can migrate as many EC2-Classic Elastic IP addresses as you can have in your account. However, when you migrate an Elastic IP address, it counts against your Elastic IP address limit for VPCs. You cannot migrate an Elastic IP address if it will result in your exceeding your limit. Similarly, when you restore an Elastic IP address to EC2-Classic, it counts against your Elastic IP address limit for EC2-Classic. For more information, see [Elastic IP Address Limit \(p. 665\)](#).

You cannot migrate an Elastic IP address that has been allocated to your account for less than 24 hours.

You can migrate an Elastic IP address from EC2-Classic using the Amazon EC2 console or the Amazon VPC console. This option is only available if your account supports EC2-Classic.

To move an Elastic IP address using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address, and choose **Actions, Move to VPC scope**.
4. In the confirmation dialog box, choose **Move Elastic IP**.

You can restore an Elastic IP address to EC2-Classic using the Amazon EC2 console or the Amazon VPC console.

To restore an Elastic IP address to EC2-Classic using the Amazon EC2 console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address, choose **Actions, Restore to EC2 scope**.
4. In the confirmation dialog box, choose **Restore**.

After you've performed the command to move or restore your Elastic IP address, the process of migrating the Elastic IP address can take a few minutes. Use the [describe-moving-addresses](#) command to check whether your Elastic IP address is still moving, or has completed moving.

After you've moved your Elastic IP address, you can view its allocation ID on the [Elastic IPs](#) page in the **Allocation ID** field.

If the Elastic IP address is in a moving state for longer than 5 minutes, contact [Premium Support](#).

To move an Elastic IP address using the command line

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

- [move-address-to-vpc](#) (AWS CLI)
- [Move-EC2AddressToVpc](#) (AWS Tools for Windows PowerShell)

To restore an Elastic IP address to EC2-Classic using the command line

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

- [restore-address-to-classic](#) (AWS CLI)
- [Restore-EC2AddressToClassic](#) (AWS Tools for Windows PowerShell)

To describe the status of your moving addresses using the command line

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

- [describe-moving-addresses](#) (AWS CLI)
- [Get-EC2Address](#) (AWS Tools for Windows PowerShell)

Sharing and Accessing Resources Between EC2-Classic and a VPC

Some resources and features in your AWS account can be shared or accessed between EC2-Classic and a VPC, for example, through ClassicLink. For more information, see [ClassicLink \(p. 708\)](#).

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 Windows Instance in EC2-Classic to a Windows Instance in a VPC \(p. 721\)](#).

The following resources can be shared or accessed between EC2-Classic and a VPC.

Resource	Notes
AMI	
Bundle task	
EBS volume	
Elastic IP address (IPv4)	You can migrate an Elastic IP address from EC2-Classic to a VPC. You can't migrate an Elastic IP address that was originally allocated for use in a VPC to EC2-Classic. For more information,

Resource	Notes
	see Migrating an Elastic IP Address from EC2-Classic (p. 706) .
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 Windows Instance in EC2-Classic to a Windows Instance in a VPC (p. 721) .
Key pair	
Load balancer	If you're using ClassicLink, you can register a linked EC2-Classic instance with a load balancer in a VPC, provided that the VPC has a subnet in the same Availability Zone as the instance. You can't migrate a load balancer from EC2-Classic to a VPC. You can't register an instance in a VPC with a load balancer in EC2-Classic.
Placement group	
Reserved Instance	You can change the network platform for your Reserved Instances from EC2-Classic to a VPC. For more information, see Modifying Reserved Instances (p. 210) .
Security group	A linked EC2-Classic instance can use a VPC security groups through ClassicLink to control traffic to and from the VPC. VPC instances can't use EC2-Classic security groups. You can't migrate a security group from EC2-Classic to a VPC. You can copy rules from a security group for EC2-Classic to a security group for a VPC. For more information, see Creating a Security Group (p. 552) .
Snapshot	

The following resources can't be shared or moved between EC2-Classic and a VPC:

- Spot Instances

ClassicLink

ClassicLink allows you to link EC2-Classic instances to a VPC in your account, within the same region. If you associate the VPC security groups with a EC2-Classic instance, this enables communication between your EC2-Classic instance and instances in your VPC using private IPv4 addresses. ClassicLink removes

the need to make use of public IPv4 addresses or Elastic IP addresses to enable communication between instances in these platforms.

ClassicLink is available to all users with accounts that support the EC2-Classic platform, and can be used with any EC2-Classic instance. For more information about migrating your resources to a VPC, see [Migrating from a Windows Instance in EC2-Classic to a Windows Instance in a VPC \(p. 721\)](#).

There is no additional charge for using ClassicLink. Standard charges for data transfer and instance usage apply.

Contents

- [ClassicLink Basics \(p. 709\)](#)
- [ClassicLink Limitations \(p. 711\)](#)
- [Working with ClassicLink \(p. 712\)](#)
- [Example IAM Policies for ClassicLink \(p. 715\)](#)
- [API and CLI Overview \(p. 717\)](#)
- [Example: ClassicLink Security Group Configuration for a Three-Tier Web Application \(p. 719\)](#)

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 \(p. 714\)](#).

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 \(p. 714\)](#).

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 \(p. 564\)](#).

For more information about policies for working with ClassicLink, see the following example: [Example IAM Policies for ClassicLink \(p. 715\)](#).

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 \(p. 719\)](#).

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 local1.

For more information and examples, see [Configurations With ClassicLink in the Amazon VPC Peering Guide](#).

ClassicLink Limitations

To use the ClassicLink feature, you need to be aware of the following limitations:

- You can link an EC2-Classic instance to only one VPC at a time.
- If you stop your linked EC2-Classic instance, it's automatically unlinked from the VPC and the VPC security groups are no longer associated with the instance. You can link your instance to the VPC again after you've restarted it.
- You cannot link an EC2-Classic instance to a VPC that's in a different region or a different AWS account.
- You cannot use ClassicLink to link a VPC instance to a different VPC, or to a EC2-Classic resource. To establish a private connection between VPCs, you can use a VPC peering connection. For more information, see the [Amazon VPC Peering Guide](#).
- You cannot associate a VPC Elastic IP address with a linked EC2-Classic instance.
- You cannot enable EC2-Classic instances for IPv6 communication. You can associate an IPv6 CIDR block with your VPC and assign IPv6 address to resources in your VPC, however, communication between a ClassicLinked instance and resources in the VPC is over IPv4 only.
- VPCs with routes that conflict with the EC2-Classic private IP address range of 10/8 cannot be enabled for ClassicLink. This does not include VPCs with 10.0.0.0/16 and 10.1.0.0/16 IP address

ranges that already have local routes in their route tables. For more information, see [Routing for ClassicLink \(p. 710\)](#).

- VPCs configured for dedicated hardware tenancy cannot be enabled for ClassicLink. Contact AWS support to request that your dedicated tenancy VPC be allowed to be enabled for ClassicLink.

Important

EC2-Classic instances are run on shared hardware. If you've set the tenancy of your VPC to dedicated because of regulatory or security requirements, then linking an EC2-Classic instance to your VPC might not conform to those requirements, as this allows a shared tenancy resource to address your isolated resources directly using private IP addresses. If you need to enable your dedicated VPC for ClassicLink, provide a detailed reason in your request to AWS support.

- If you link your EC2-Classic instance to a VPC in the 172.16.0.0/16 range, and you have a DNS server running on the 172.16.0.23/32 IP address within the VPC, then your linked EC2-Classic instance can't access the VPC DNS server. To work around this issue, run your DNS server on a different IP address within the VPC.
- ClassicLink doesn't support transitive relationships out of the VPC. Your linked EC2-Classic instance doesn't have access to any VPN connection, VPC gateway endpoint, NAT gateway, or Internet gateway associated with the VPC. Similarly, resources on the other side of a VPN connection or an Internet gateway don't have access to a linked EC2-Classic instance.

Working with ClassicLink

You can use the Amazon EC2 and Amazon VPC consoles to work with the ClassicLink feature. You can enable or disable a VPC for ClassicLink, and link and unlink EC2-Classic instances to a VPC.

Note

The ClassicLink features are only visible in the consoles for accounts and regions that support EC2-Classic.

Tasks

- [Enabling a VPC for ClassicLink \(p. 712\)](#)
- [Linking an Instance to a VPC \(p. 713\)](#)
- [Creating a VPC with ClassicLink Enabled \(p. 713\)](#)
- [Linking an EC2-Classic Instance to a VPC at Launch \(p. 713\)](#)
- [Viewing Your ClassicLink-Enabled VPCs and Linked EC2-Classic Instances \(p. 714\)](#)
- [Enabling ClassicLink DNS Support \(p. 714\)](#)
- [Disabling ClassicLink DNS Support \(p. 714\)](#)
- [Unlinking a EC2-Classic Instance from a VPC \(p. 715\)](#)
- [Disabling ClassicLink for a VPC \(p. 715\)](#)

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 \(p. 710\)](#).

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 \(p. 315\)](#).

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**.

Example IAM Policies for ClassicLink

You can enable a VPC for ClassicLink and then link an EC2-Classic instance to the VPC. You can also view your ClassicLink-enabled VPCs, and all of your EC2-Classic instances that are linked to a VPC. You can create policies with resource-level permission for the `ec2:EnableVpcClassicLink`, `ec2:DisableVpcClassicLink`, `ec2:AttachClassicLinkVpc`, and `ec2:DetachClassicLinkVpc` actions to control how users are able to use those actions. Resource-level permissions are not supported for `ec2:Describe*` actions.

Examples

- [Full Permissions to Work with ClassicLink \(p. 715\)](#)
- [Enable and Disable a VPC for ClassicLink \(p. 716\)](#)
- [Link Instances \(p. 716\)](#)
- [Unlink Instances \(p. 717\)](#)

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/*"
            ]
        }
    ]
}
```

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-EC2ClassicLinkInstance](#) (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

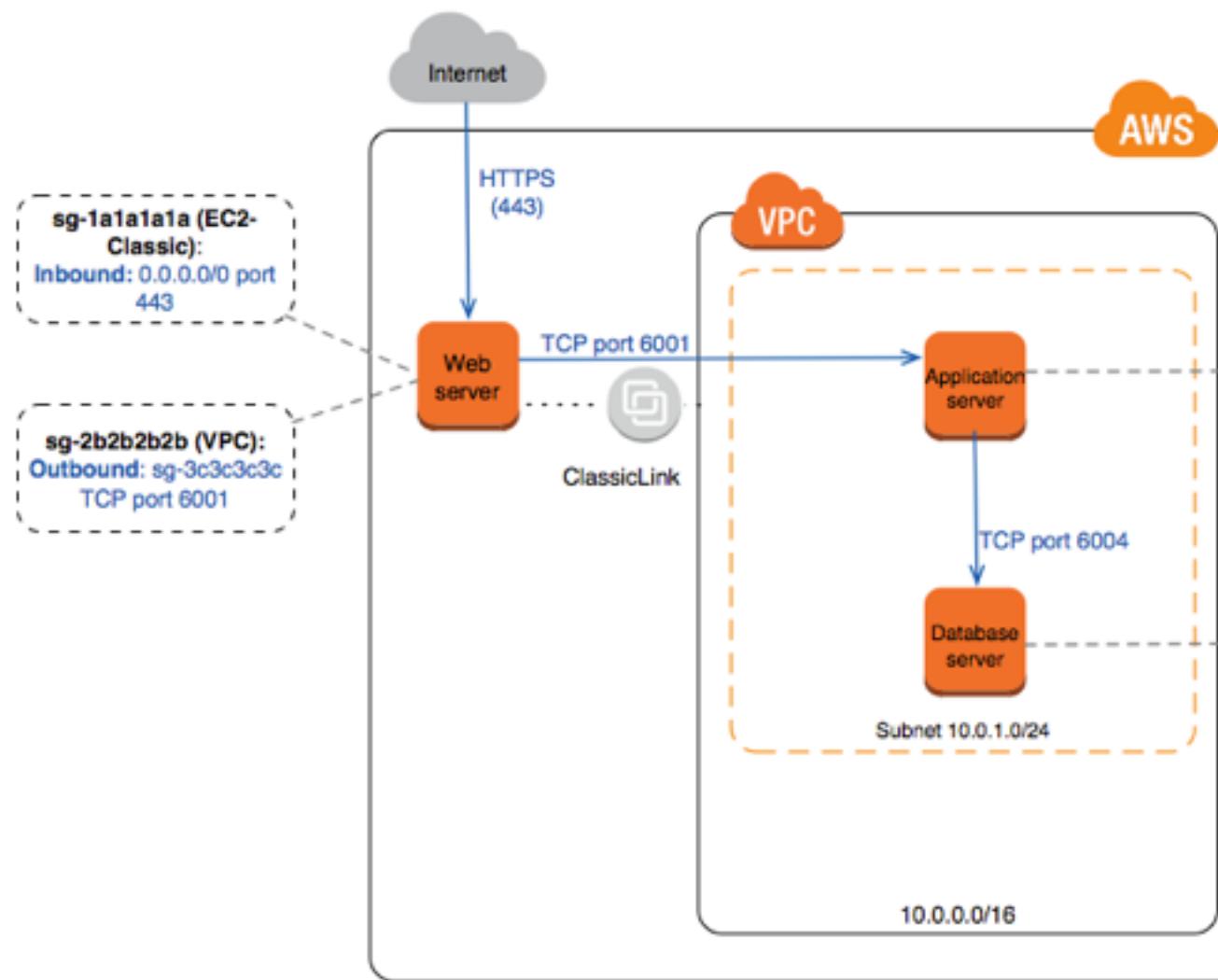
- [describe-vpc-classic-link-dns-support](#) (AWS CLI)
- [Get-EC2VpcClassicLinkDnsSupport](#) (AWS Tools for Windows PowerShell)
- [DescribeVpcClassicLinkDnsSupport](#) (Amazon EC2 Query API)

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 Windows Instance in EC2-Classic to a Windows Instance in a VPC

If you created your AWS account before 2013-12-04, you might have support for EC2-Classic in some regions. Some Amazon EC2 resources and features, such as enhanced networking and newer instance types, require a virtual private cloud (VPC). Some resources can be shared between EC2-Classic and a VPC, while some can't. For more information, see [Sharing and Accessing Resources Between EC2-Classic and a VPC \(p. 707\)](#).

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 Linux instance, see [Migrating a Linux Instance from EC2-Classic to a VPC](#) in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- [Full Migration to a VPC \(p. 722\)](#)
- [Incremental Migration to a VPC Using ClassicLink \(p. 727\)](#)

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 \(p. 722\)](#)
- [Step 2: Configure Your Security Group \(p. 722\)](#)
- [Step 3: Create an AMI from Your EC2-Classic Instance \(p. 723\)](#)
- [Step 4: Launch an Instance Into Your VPC \(p. 724\)](#)
- [Example: Migrating a Simple Web Application \(p. 725\)](#)

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:

- Your AWS 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 Windows Instances \(p. 548\)](#).

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 Windows Instances \(p. 548\)](#).

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.

For more information, see [Creating a Custom Windows AMI \(p. 65\)](#).

(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 \(p. 734\)](#)
- [Creating an Amazon EBS Volume \(p. 749\)](#)
- [Attaching an Amazon EBS Volume to an Instance \(p. 752\)](#)

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 \(p. 786\)](#)
- [Creating an Amazon EBS Snapshot \(p. 789\)](#)

- Restoring an Amazon EBS Volume from a Snapshot (p. 751)

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 \(p. 315\)](#).

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

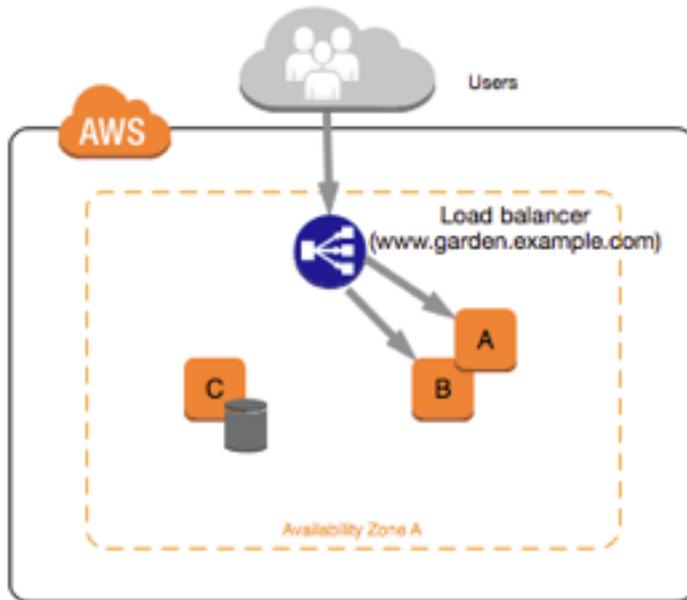
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Switch to your new AWS account.

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.

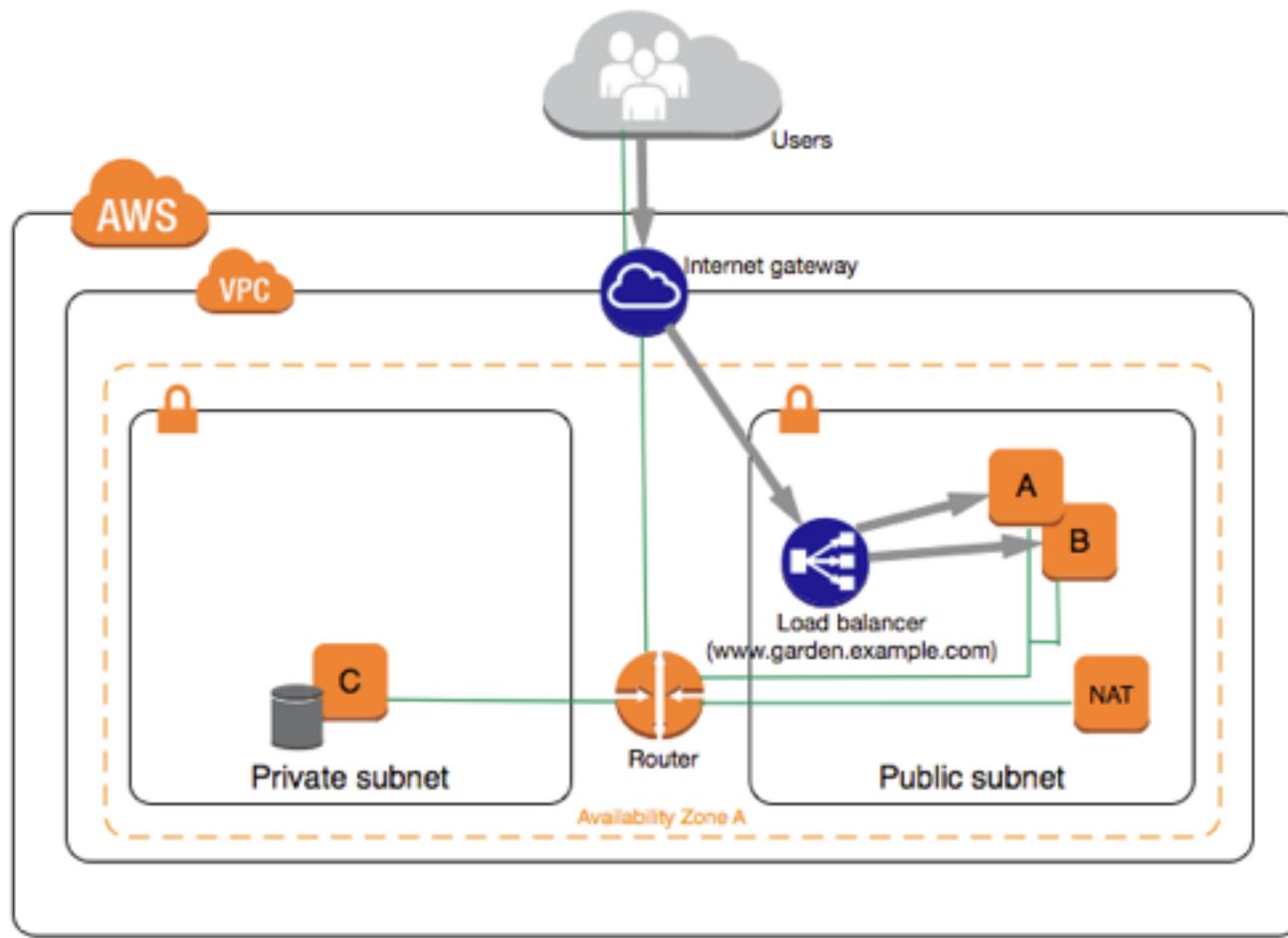
For more information about the parameters you can configure in each step of the wizard, see [Launching an Instance Using the Launch Instance Wizard \(p. 315\)](#).

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 a VPC, and create a load balancer in your public subnet to load balance the traffic between your web servers.



To migrate your web application to a VPC, you can follow these steps:

- **Create a VPC:** In this case, you can use the VPC wizard in the Amazon VPC console to create your VPC and subnets. The second wizard configuration creates a VPC with one private and one public subnet, and launches and configures a NAT device in your public subnet for you. For more information, see [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 \(p. 723\)](#).
- **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 \(p. 722\)](#).

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 \(p. 724\)](#).
- **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 a VPC. After you've migrated them, you can associate them with your new web servers in your VPC. For more information, see [Migrating an Elastic IP Address from EC2-Classic \(p. 706\)](#).
- **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 \(p. 708\)](#).

Topics

- [Step 1: Prepare Your Migration Sequence \(p. 728\)](#)
- [Step 2: Create a VPC \(p. 728\)](#)
- [Step 3: Enable Your VPC for ClassicLink \(p. 728\)](#)
- [Step 4: Create an AMI from Your EC2-Classic Instance \(p. 728\)](#)
- [Step 5: Launch an Instance Into Your VPC \(p. 729\)](#)
- [Step 6: Link Your EC2-Classic Instances to Your VPC \(p. 729\)](#)
- [Step 7: Complete the VPC Migration \(p. 730\)](#)

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 \(p. 708\)](#).

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.

For more information, see [Creating a Custom Windows AMI \(p. 65\)](#).

(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 \(p. 734\)](#)
- [Creating an Amazon EBS Volume \(p. 749\)](#)
- [Attaching an Amazon EBS Volume to an Instance \(p. 752\)](#)

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 \(p. 786\)](#)
- [Creating an Amazon EBS Snapshot \(p. 789\)](#)
- [Restoring an Amazon EBS Volume from a Snapshot \(p. 751\)](#)

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 \(p. 315\)](#).

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 a VPC. For more information, see [Migrating an Elastic IP Address from EC2-Classic \(p. 706\)](#).

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.

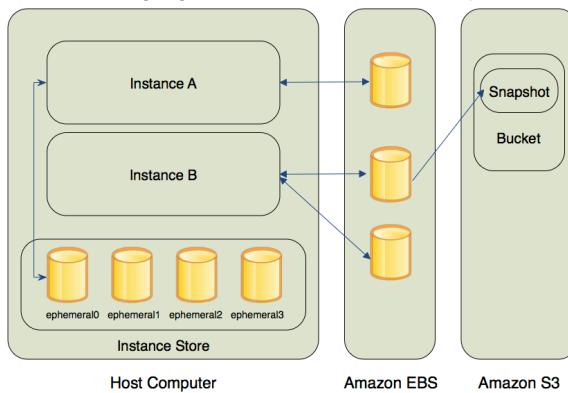
Storage

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 \(p. 732\)](#)
- [Amazon EC2 Instance Store \(p. 843\)](#)
- [Amazon Simple Storage Service \(Amazon S3\) \(p. 851\)](#)

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 \(p. 815\)](#).

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 \(p. 732\)](#).

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 \(p. 843\)](#).

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 \(Amazon S3\) \(p. 851\)](#).

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 \(p. 855\)](#).

You can also attach EBS volumes to a running instance. For more information, see [Attaching an Amazon EBS Volume to an Instance \(p. 752\)](#).

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 \(p. 815\)](#).

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 greater flexibility when defining access controls, including the ability to create, rotate, disable, and audit encryption keys that are specific to individual applications and users. 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 \(p. 733\)](#)
- [Amazon EBS Volumes \(p. 734\)](#)
- [Amazon EBS Snapshots \(p. 786\)](#)
- [Amazon EBS–Optimized Instances \(p. 807\)](#)
- [Amazon EBS Encryption \(p. 815\)](#)
- [Amazon EBS and NVMe \(p. 819\)](#)
- [Amazon EBS Volume Performance on Windows Instances \(p. 820\)](#)
- [Amazon CloudWatch Events for Amazon EBS \(p. 836\)](#)

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 \(p. 749\)](#).
- 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 16,000 IOPS and 250 MiB/s of throughput. For more information, see [General Purpose SSD \(**gp2**\) Volumes \(p. 739\)](#).
- With Provisioned IOPS SSD (**io1**) volumes, you can provision a specific level of I/O performance. **Io1** volumes support up to 64,000 IOPS and 1,000 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 \(p. 742\)](#).
- 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 \(p. 742\)](#).
- 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 \(p. 744\)](#).
- 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 Windows \(p. 753\)](#).
- 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 \(p. 815\)](#).
- 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 \(p. 786\)](#).
- 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 \(p. 789\)](#), [Restoring an Amazon EBS Volume from a Snapshot \(p. 751\)](#), and [Copying an Amazon EBS Snapshot \(p. 793\)](#).

- 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 \(p. 870\)](#).
- 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 Windows Instances \(p. 820\)](#).

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 the 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 (`iop1`), 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 \(p. 736\)](#).

Contents

- [Benefits of Using EBS Volumes \(p. 734\)](#)
- [Amazon EBS Volume Types \(p. 736\)](#)
- [Creating an Amazon EBS Volume \(p. 749\)](#)
- [Restoring an Amazon EBS Volume from a Snapshot \(p. 751\)](#)
- [Attaching an Amazon EBS Volume to an Instance \(p. 752\)](#)
- [Making an Amazon EBS Volume Available for Use on Windows \(p. 753\)](#)
- [Viewing Volume Information \(p. 756\)](#)
- [Monitoring the Status of Your Volumes \(p. 757\)](#)
- [Detaching an Amazon EBS Volume from an Instance \(p. 771\)](#)
- [Deleting an Amazon EBS Volume \(p. 773\)](#)
- [Modifying the Size, IOPS, or Type of an EBS Volume on Windows \(p. 773\)](#)

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 NTFS, 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 \(p. 758\)](#). 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 \(p. 815\)](#).

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 \(p. 815\)](#).

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 \(p. 890\)](#).

- **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 for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	<ul style="list-style-type: none">• Recommended for most workloads	<ul style="list-style-type: none">• Critical business applications that require sustained IOPS performance,	<ul style="list-style-type: none">• Streaming workloads requiring consistent,	<ul style="list-style-type: none">• Throughput-oriented storage for large volumes

	Solid-State Drives (SSD)		Hard Disk Drives (HDD)	
	<ul style="list-style-type: none"> System boot volumes Virtual desktops Low-latency interactive apps Development and test environments 	<ul style="list-style-type: none"> or more than 16,000 IOPS or 250 MiB/s of throughput per volume Large database workloads, such as: <ul style="list-style-type: none"> MongoDB Cassandra Microsoft SQL Server MySQL PostgreSQL Oracle 	<ul style="list-style-type: none"> fast throughput at a low price Big data Data warehouses Log processing Cannot be a boot volume 	<ul style="list-style-type: none"> 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	16,000***	64,000****	500	250
Max. Throughput/Volume	250 MiB/s***	1,000 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 without a volume-type argument 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

*** General Purpose SSD (gp2) volumes have a throughput limit between 128 MiB/s and 250 MiB/s depending on volume size. Volumes greater than 170 GiB and below 334 GiB deliver a maximum throughput of 250 MiB/s if burst credits are available. Volumes with 334 GiB and above deliver 250 MiB/s irrespective of burst credits. An older gp2 volume may not see full performance unless a `ModifyVolume` action is performed on it. For more information, see [Modifying the Size, IOPS, or Type of an EBS Volume on Windows](#).

**** Maximum IOPS of 64,000 is guaranteed only on [Nitro-based instances](#). Other instance families guarantee performance up to 32,000 IOPS.

† Maximum throughput of 1,000 MiB/s is guaranteed only on [Nitro-based instances](#). Other instance families guarantee up to 500 MiB/s. An older io1 volume may not see full performance unless a `ModifyVolume` action is performed on it. For more information, see [Modifying the Size, IOPS, or Type of an EBS Volume on Windows](#).

†† 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	<code>standard</code>
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

The following Amazon EBS volume considerations apply to Windows boot volumes:

- Windows boot volumes must use an MBR partition table, which limits the usable space to 2 TiB, regardless of volume size.
- Windows boot volumes of 2 TiB (2048 GiB) that have been converted to use a dynamic MBR partition table display an error when examined with Disk Manager.

The following Amazon EBS volume considerations apply to Windows data (non-boot) volumes:

- Windows volumes 2 TiB (2048 GiB) or greater must use a GPT partition table to access the entire volume.
- Amazon EBS volumes over 2048 GiB that are attached to Windows instances at launch are automatically formatted with a GPT partition table.
- Amazon EBS volumes attached to Windows instances after launch must be manually initialized with a GPT partition table. For more information, see [Making an Amazon EBS Volume Available for Use](#).

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 Windows Instances \(p. 820\)](#).

For more information about pricing for these volume types, see [Amazon EBS Pricing](#).

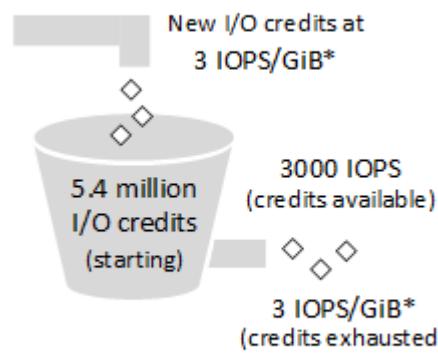
General Purpose SSD (gp2) Volumes

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 16,000 IOPS (at 5,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size. AWS designs gp2 volumes to deliver 90% of 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

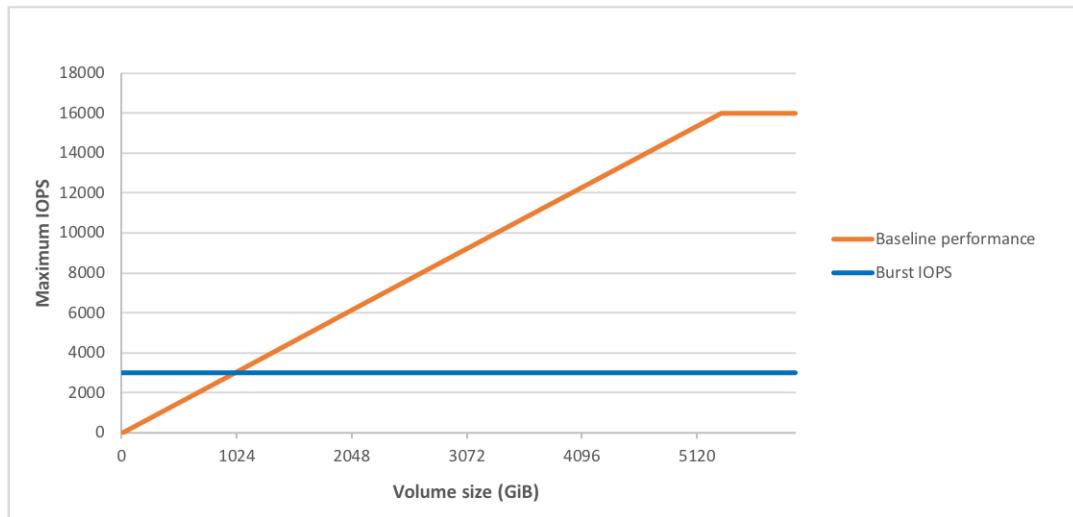
The performance of gp2 volumes is tied to volume size, which determines the baseline performance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher baseline performance levels and accumulate I/O credits faster. I/O credits represent the available bandwidth that 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 16,000 IOPS

Each volume receives an initial I/O credit balance of 5.4 million I/O credits, which is enough to sustain the maximum burst performance of 3,000 IOPS for 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)	Minimum burst duration @ 3,000 IOPS (seconds)	Seconds to fill empty credit balance
1	100	1862	54,000
100	300	2,000	18,000
250	750	2,400	7,200
334 (Min. size for max. throughput)	1002	2703	5389
500	1,500	3,600	3,600
750	2,250	7,200	2,400
1,000	3,000	N/A*	N/A*
5,334 (Min. size for max. IOPS)	16,000	N/A*	N/A*
16,384 (16 TiB, max. volume size)	16,000	N/A*	N/A*

* 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 250 MiB/s. When I/O demand drops below the baseline level and unused credits are added to the I/O credit balance, the maximum IOPS performance of the volume again exceeds the baseline. For example, a 100 GiB gp2 volume with an empty credit balance has a baseline performance of 300 IOPS and a throughput limit of 75 MiB/s (300 I/O operations per second * 256 KiB per I/O operation = 75 MiB/s). The larger a volume is, the greater the baseline performance is and the faster it replenishes the credit balance. For more information about how IOPS are measured, see [I/O Characteristics](#).

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 16,000 IOPS.

For information about using CloudWatch metrics and alarms to monitor your burst bucket balance, see [Monitoring the Burst Bucket Balance for gp2, st1, and sc1 Volumes \(p. 749\)](#).

Throughput Performance

Throughput for a gp2 volume can be calculated using the following formula, up to the throughput limit of 250 MiB/s:

$$\text{Throughput in MiB/s} = (\text{Volume size in GiB}) \times (\text{IOPS per GiB}) \times (\text{I/O size in KiB})$$

Assuming V = volume size, I = I/O size, R = I/O rate, and T = throughput, this can be simplified to:

$$T = VIR$$

The smallest volume size that achieves the maximum throughput is given by:

$$\begin{aligned} V &= \frac{T}{I R} \\ &= \frac{(250 \text{ MiB/s})}{(256 \text{ KiB})(3 \text{ IOPS/GiB})} \\ &= \frac{[(250)(2^{20})(\text{Bytes})]/(\text{s})}{(256)(2^{10})(\text{Bytes})(3 \text{ IOPS}/[(2^{30})(\text{Bytes})])} \\ &= \frac{(250)(2^{20})(2^{30})(\text{Bytes})}{(256)(2^{10})(3)} \\ &= 357913940000 \text{ Bytes} \end{aligned}$$

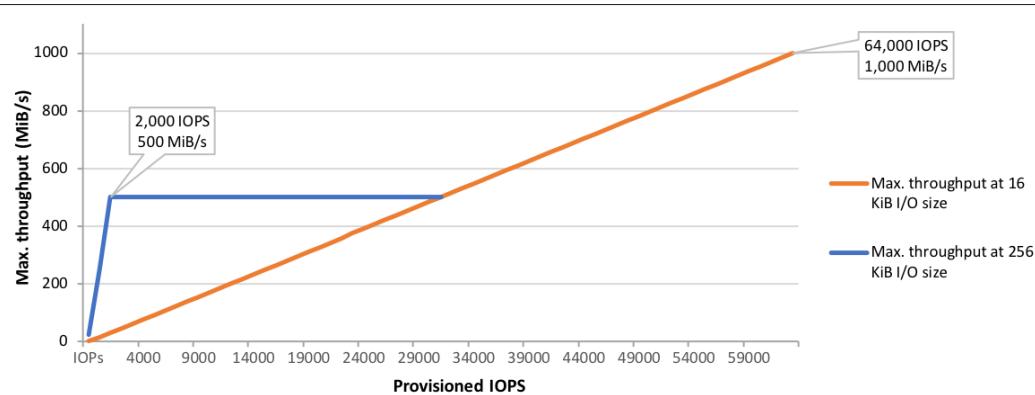
= 333# GiB (334 GiB in practice because volumes are provisioned in whole gibibytes)

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. You can provision from 100 IOPS up to 64,000 IOPS per volume on Nitro system instance families and up to 32,000 on other instance families. 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. On a supported instance type, any volume 1,280 GiB in size or greater allows provisioning up to the 64,000 IOPS maximum ($50 \times 1,280 \text{ GiB} = 64,000$).

The throughput limit of io1 volumes is 256 KiB/s for each IOPS provisioned, up to a maximum of 1,000 MiB/s (at 64,000 IOPS). Up to 32,000 IOPS, I/O size can be as high as 256 KiB, while above that a 16 KiB size is used.



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.

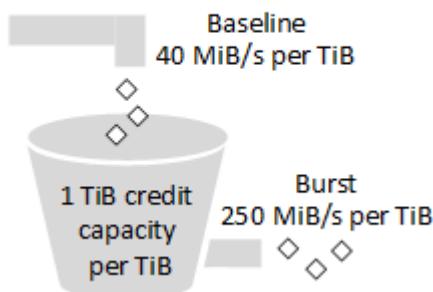
This volume type is optimized for workloads involving large, sequential I/O, and we recommend that customers with workloads performing small, random I/O use gp2. For more information, see [Inefficiency of Small Read/Writes on HDD \(p. 748\)](#).

Throughput Credits and Burst Performance

Like gp2, st1 uses a burst-bucket model for performance. Volume size determines the baseline throughput of your volume, which is the rate at which the volume accumulates throughput credits. Volume size also determines the burst throughput of your volume, which is the rate at which you can spend credits when they are available. Larger volumes have higher baseline and burst throughput. The more credits your volume has, the longer it can drive I/O at the burst level.

The following diagram shows the burst-bucket behavior for st1.

ST1 burst bucket



Subject to throughput and throughput-credit caps, the available throughput of an st1 volume is expressed by the following formula:

$$\text{(Volume size)} \times (\text{Credit accumulation rate per TiB}) = \text{Throughput}$$

For a 1-TiB st1 volume, burst throughput is limited to 250 MiB/s, the bucket fills with credits at 40 MiB/s, and it can hold up to 1 TiB-worth of credits.

Larger volumes scale these limits linearly, with throughput capped at a maximum of 500 MiB/s. After the bucket is depleted, throughput is limited to the baseline rate of 40 MiB/s per TiB.

On volume sizes ranging from 0.5 to 16 TiB, baseline throughput varies from 20 to a cap of 500 MiB/s, which is reached at 12.5 TiB as follows:

$$12.5 \text{ TiB} \times \frac{40 \text{ MiB/s}}{1 \text{ TiB}} = 500 \text{ MiB/s}$$

Burst throughput varies from 125 MiB/s to a cap of 500 MiB/s, which is reached at 2 TiB as follows:

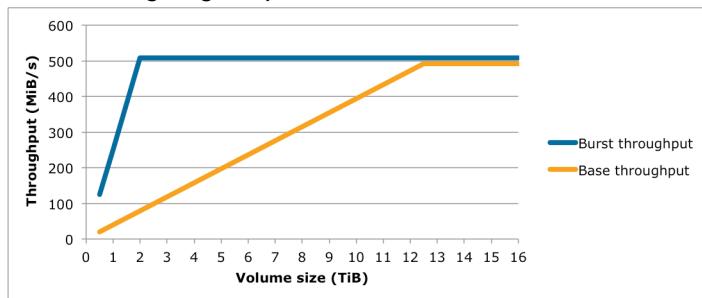
$$2 \text{ TiB} \times \frac{250 \text{ MiB/s}}{1 \text{ TiB}} = 500 \text{ MiB/s}$$

The following table states the full range of base and burst throughput values for st1:

Volume Size (TiB)	ST1 Base Throughput (MiB/s)	ST1 Burst Throughput (MiB/s)
0.5	20	125
1	40	250
2	80	500

Volume Size (TiB)	ST1 Base Throughput (MiB/s)	ST1 Burst Throughput (MiB/s)
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 \(p. 749\)](#).

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.

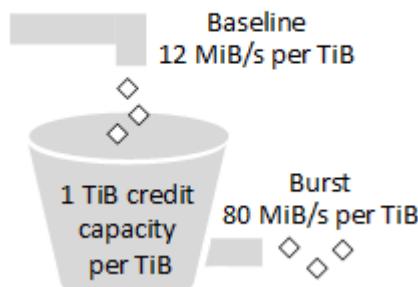
Note

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 \(p. 748\)](#).

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:

$$16 \text{ TiB} \times \frac{12 \text{ MiB/s}}{1 \text{ TiB}} = 192 \text{ MiB/s}$$

Burst throughput varies from 40 MiB/s to a cap of 250 MiB/s, which is reached at 3.125 TiB as follows:

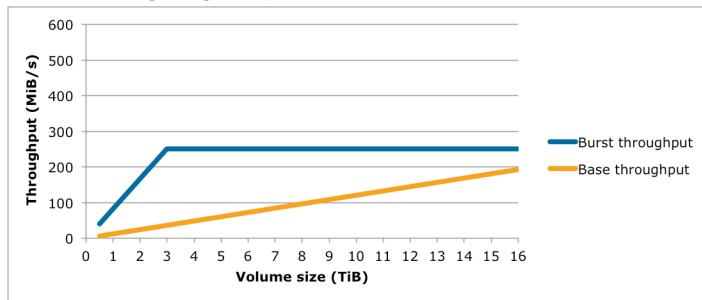
$$3.125 \text{ TiB} \times \frac{80 \text{ MiB/s}}{1 \text{ TiB}} = 250 \text{ MiB/s}$$

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

Volume Size (TiB)	SC1 Base Throughput (MiB/s)	SC1 Burst Throughput (MiB/s)
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 \(p. 749\)](#).

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 \(p. 749\)](#).

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.

$$\begin{aligned} \frac{5 \text{ TiB}}{0.000238418 \text{ TiB/s}} &= 20972 \text{ s} = 5.83 \text{ hours (optimal)} \\ 5.83 \text{ hours} + \frac{5.83 \text{ hours}}{(0.90)(0.99)} &= 6.54 \text{ hours (minimum expected)} \end{aligned}$$

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 \(p. 816\)](#). You can use IAM policies to enforce encryption on new volumes. For more information, see the example IAM policies in [Working with Volumes \(p. 604\)](#) and [Launching Instances \(RunInstances\) \(p. 610\)](#).

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 \(p. 315\)](#) and [Block Device Mapping \(p. 855\)](#). You can restore volumes from previously created snapshots. For more information, see [Restoring an Amazon EBS Volume from a Snapshot \(p. 751\)](#).

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 \(p. 822\)](#).

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 \(p. 825\)](#).

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 \(p. 881\)](#).
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 \(p. 736\)](#).

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.

Note

The following Amazon EBS volume considerations apply to Windows boot volumes:

- Windows boot volumes must use an MBR partition table, which limits the usable space to 2 TiB, regardless of volume size.
- Windows boot volumes of 2 TiB (2048 GiB) that have been converted to use a dynamic MBR partition table display an error when examined with Disk Manager.

The following Amazon EBS volume considerations apply to Windows data (non-boot) volumes:

- Windows volumes 2 TiB (2048 GiB) or greater must use a GPT partition table to access the entire volume.
- Amazon EBS volumes over 2048 GiB that are attached to Windows instances at launch are automatically formatted with a GPT partition table.
- Amazon EBS volumes attached to Windows instances after launch must be manually initialized with a GPT partition table. For more information, see [Making an Amazon EBS Volume Available for Use](#).

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 \(p. 816\)](#).

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 \(p. 786\)](#).

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 \(p. 816\)](#).

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 \(p. 825\)](#).

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 \(p. 793\)](#).

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 \(p. 736\)](#).

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 \(p. 816\)](#).

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.

Note

The following Amazon EBS volume considerations apply to Windows boot volumes:

- Windows boot volumes must use an MBR partition table, which limits the usable space to 2 TiB, regardless of volume size.
- Windows boot volumes of 2 TiB (2048 GiB) that have been converted to use a dynamic MBR partition table display an error when examined with Disk Manager.

The following Amazon EBS volume considerations apply to Windows data (non-boot) volumes:

- Windows volumes 2 TiB (2048 GiB) or greater must use a GPT partition table to access the entire volume.
- Amazon EBS volumes over 2048 GiB that are attached to Windows instances at launch are automatically formatted with a GPT partition table.
- Amazon EBS volumes attached to Windows instances after launch must be manually initialized with a GPT partition table. For more information, see [Making an Amazon EBS Volume Available for Use](#).

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 \(p. 752\)](#).
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 Windows \(p. 773\)](#).

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 \(p. 853\)](#).
- 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 \(p. 816\)](#).
- 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 Windows Instances \(p. 854\)](#).
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 Windows \(p. 753\)](#).

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 Windows

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 \(p. 786\)](#).

You can get directions for volumes on a Linux instance from [Making a Volume Available for Use on Linux](#) in the *Amazon EC2 User Guide for Linux Instances*.

You can make an EBS volume available for use using the Disk Management utility and the DiskPart command line tool.

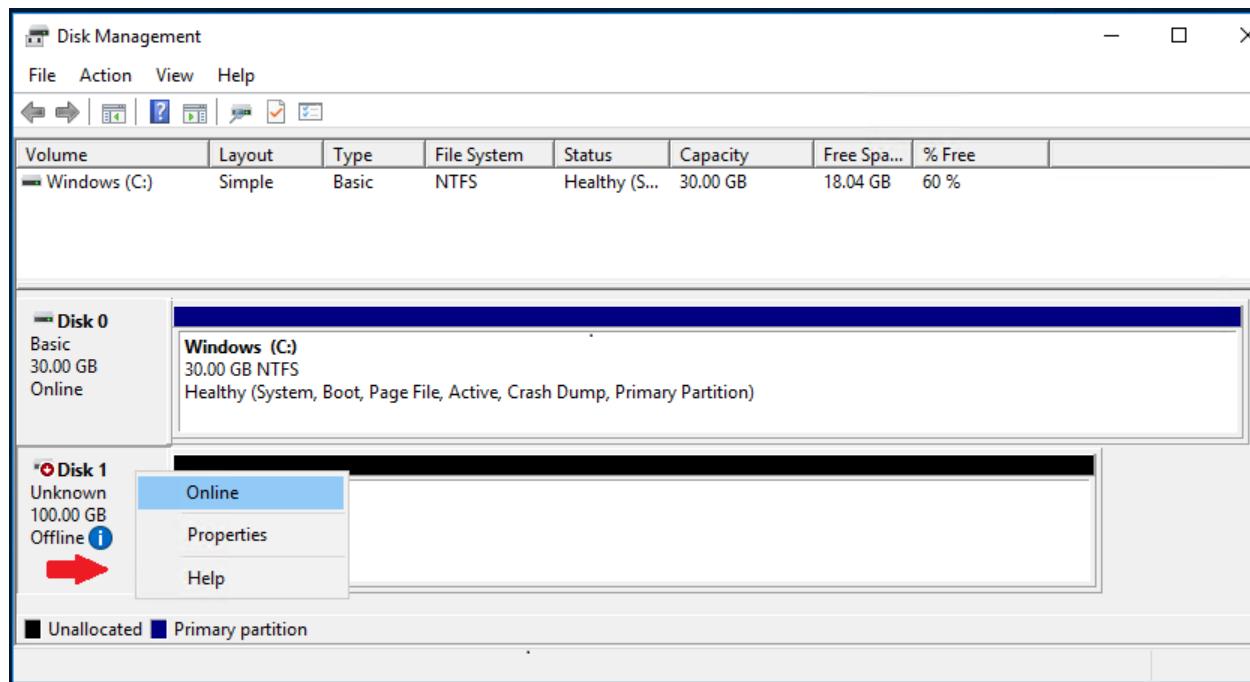
To make an EBS volume available for use using the Disk Management utility

1. Log in to your Windows instance using Remote Desktop. For more information, see, [Connecting to Your Windows Instance \(p. 359\)](#).
2. Start the Disk Management utility. On the taskbar, open the context (right-click) menu for the Windows logo and choose **Disk Management**.

Note

On Windows Server 2008, choose **Start, Administrative Tools, Computer Management, Disk Management.**

3. Bring the volume online. In the lower pane, open the context (right-click) menu for the left panel for the disk for the EBS volume. Choose **Online**.

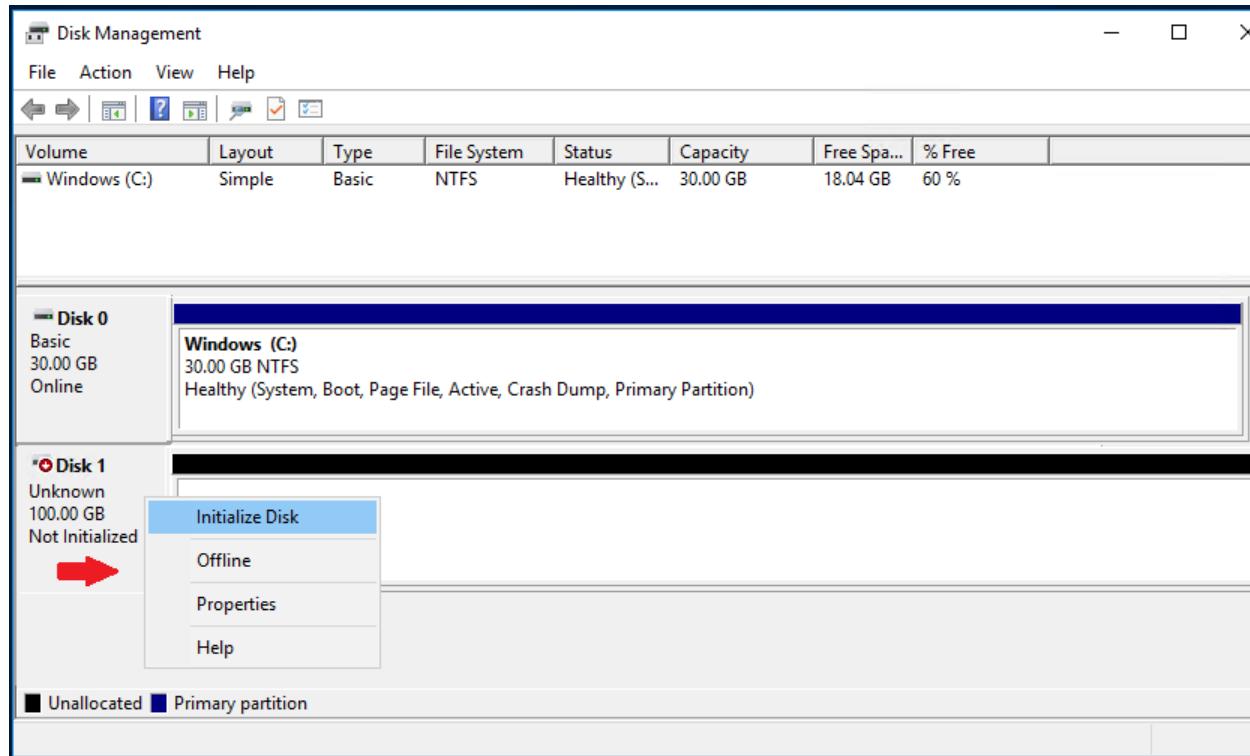


4. (Conditional) You must initialize the disk before you can use it.

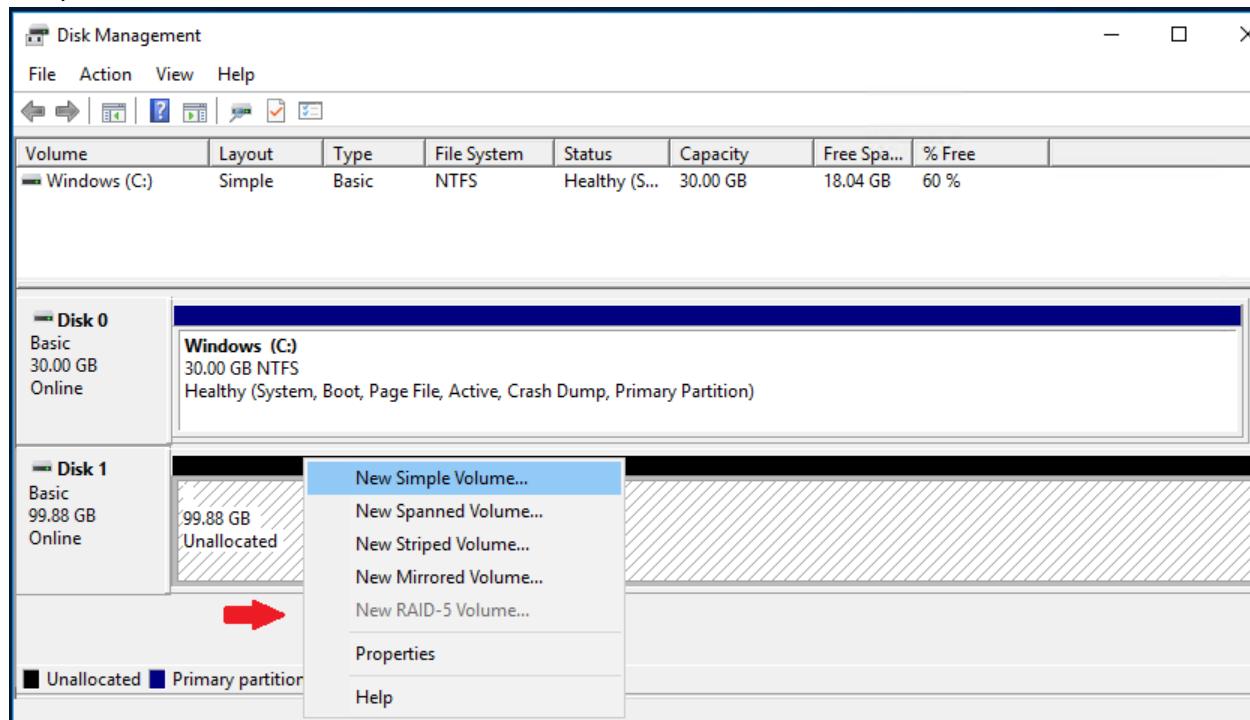
Warning

If you're mounting a volume that already has data on it (for example, a public data set, or a volume that you created from a snapshot), do not reformat the volume or you will delete the existing data.

If the disk is not initialized, initialize it as follows. Open the context (right-click) menu for the left panel for the disk and choose **Initialize Disk**. In the **Initialize Disk** dialog box, select a partition style and choose **OK**.



5. Open the context (right-click) menu for the right panel for the disk and choose **New Simple Volume**. Complete the wizard.



To make an EBS volume available for use using the DiskPart command line tool

1. Log in to your Windows instance using Remote Desktop. For more information, see [Connecting to Your Windows Instance \(p. 359\)](#).
2. Create a new script file named `diskpart.txt`.
3. Add the following commands to the script file and specify the volume label and drive letter. This script configures the volume to use the master boot record (MBR) partition structure, formats the volume as an NTFS volume, sets the volume label, and assigns it a drive letter.

Warning

If you're mounting a volume that already has data on it, do not reformat the volume or you will delete the existing data.

```
select disk 1
attributes disk clear readonly
online disk
convert mbr
create partition primary
format quick fs=ntfs label="volume_label"
assign letter="drive_letter"
```

For more information, see [DiskPart Syntax and Parameters](#).

4. Navigate to the folder in which the script is located and execute the following command:

```
C:\> diskpart /s diskpart.txt
```

Viewing Volume Information

You can view descriptive information about your EBS volumes. For example, you can view information about all volumes in a specific region or view detailed information about a single volume, including its size, volume type, whether the volume is encrypted, which master key was used to encrypt the volume, and the specific instance to which the volume is attached.

You can get additional information about your EBS volumes, such as how much disk space is available, from the operating system on the instance.

Viewing Descriptive information

To view information about an EBS volume using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Volumes**.
3. 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 \(AWS Tools for Windows PowerShell\)](#)

Viewing Free Disk Space

You can get additional information about your EBS volumes, such as how much disk space is available, from the Windows operating system on the instance. For example, you can view the free disk space by opening File Explorer and selecting **This PC**.

You can also view the free disk space using the following `dir` command and examining the last line of the output:

```
C:\> dir C:  
Volume in drive C has no label.  
Volume Serial Number is 68C3-8081  
  
Directory of C:\  
  
03/25/2018 02:10 AM <DIR> .  
03/25/2018 02:10 AM <DIR> ..  
03/25/2018 03:47 AM <DIR> Contacts  
03/25/2018 03:47 AM <DIR> Desktop  
03/25/2018 03:47 AM <DIR> Documents  
03/25/2018 03:47 AM <DIR> Downloads  
03/25/2018 03:47 AM <DIR> Favorites  
03/25/2018 03:47 AM <DIR> Links  
03/25/2018 03:47 AM <DIR> Music  
03/25/2018 03:47 AM <DIR> Pictures  
03/25/2018 03:47 AM <DIR> Saved Games  
03/25/2018 03:47 AM <DIR> Searches  
03/25/2018 03:47 AM <DIR> Videos  
0 File(s) 0 bytes  
13 Dir(s) 18,113,662,976 bytes free
```

You can also view the free disk space using the following `fsutil` command:

```
C:\> fsutil volume diskfree C:  
Total # of free bytes : 18113204224  
Total # of bytes : 32210153472  
Total # of avail free bytes : 18113204224
```

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 \(p. 758\)](#)
- [Monitoring Volumes with Status Checks \(p. 762\)](#)
- [Monitoring Volume Events \(p. 764\)](#)
- [Working with an Impaired Volume \(p. 766\)](#)
- [Working with the AutoEnableIO Volume Attribute \(p. 769\)](#)

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*.

Some of these metrics have differences on Nitro-based instances. For a list of instance types based on the Nitro system, see [Nitro-based Instances \(p. 113\)](#).

The `AWS/EBS` namespace includes the following metrics.

Metric	Description
<code>VolumeReadBytes</code>	<p>Provides information on the read 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 read operation during the period, except on volumes attached to a Nitro-based instance, where the average represents the average over the specified period. The <code>SampleCount</code> statistic reports the total number of read operations during the period, except on volumes attached to a Nitro-based instance, where the sample count represents the number of data points used in the statistical calculation. For Xen instances, data is reported only when there is read activity on the volume.</p> <p>The <code>Minimum</code> and <code>Maximum</code> statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Bytes</p>

Metric	Description
VolumeWriteBytes	<p>Provides information on the write operations in a specified period of time. The Sum statistic reports the total number of bytes transferred during the period. The Average statistic reports the average size of each write operation during the period, except on volumes attached to a Nitro-based instance, where the average represents the average over the specified period. The SampleCount statistic reports the total number of write operations during the period, except on volumes attached to a Nitro-based instance, where the sample count represents the number of data points used in the statistical calculation. For Xen instances, data is reported only when there is write activity on the volume.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Bytes</p>
VolumeReadOps	<p>The total number of read operations in a specified period of time.</p> <p>To calculate the average read operations per second (read IOPS) for the period, divide the total read operations in the period by the number of seconds in that period.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Count</p>
VolumeWriteOps	<p>The total number of write operations in a specified period of time.</p> <p>To calculate the average write operations per second (write IOPS) for the period, divide the total write operations in the period by the number of seconds in that period.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Count</p>
VolumeTotalReadTime	<p>The total number of seconds spent by all read operations that completed in a specified period of time. If multiple requests are submitted at the same time, this total could be greater than the length of the period. For example, for a period of 5 minutes (300 seconds): if 700 operations completed during that period, and each operation took 1 second, the value would be 700 seconds. For Xen instances, data is reported only when there is read activity on the volume.</p> <p>The Average statistic on this metric is not relevant for volumes attached to Nitro-based instances.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Seconds</p>

Metric	Description
VolumeTotalWriteTime	<p>The total number of seconds spent by all write operations that completed in a specified period of time. If multiple requests are submitted at the same time, this total could be greater than the length of the period. For example, for a period of 5 minutes (300 seconds): if 700 operations completed during that period, and each operation took 1 second, the value would be 700 seconds.</p> <p>For Xen instances, data is reported only when there is write activity on the volume.</p> <p>The Average statistic on this metric is not relevant for volumes attached to Nitro-based instances.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Seconds</p>
VolumeIdleTime	<p>The total number of seconds in a specified period of time when no read or write operations were submitted.</p> <p>The Average statistic on this metric is not relevant for volumes attached to Nitro-based instances.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Seconds</p>
VolumeQueueLength	<p>The number of read and write operation requests waiting to be completed in a specified period of time.</p> <p>The Sum statistic on this metric is not relevant for volumes attached to Nitro-based instances.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances.</p> <p>Units: Count</p>
VolumeThroughputPercentage	<p>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>

Metric	Description
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 Nitro-based instances.</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)	$\text{Sum}(\text{VolumeWriteBytes}) / \text{Period} / 1024$
Read Throughput (IOPS)	$\text{Sum}(\text{VolumeReadOps}) / \text{Period}$
Write Throughput (IOPS)	$\text{Sum}(\text{VolumeWriteOps}) / \text{Period}$
Avg Queue Length (Operations)	$\text{Avg}(\text{VolumeQueueLength})$
% Time Spent Idle	$\text{Sum}(\text{VolumeIdleTime}) / \text{Period} \times 100$
Avg Read Size (KiB/Operation)	<p>$\text{Avg}(\text{VolumeReadBytes}) / 1024$</p> <p>For Nitro-based instances, the following formula derives Average Read Size using CloudWatch Metric Math:</p> $(\text{Sum}(\text{VolumeReadBytes}) / \text{Sum}(\text{VolumeReadOps})) / 1024$

Graph	Description using raw metrics
	The <code>VolumeReadBytes</code> and <code>VolumeReadOps</code> metrics are available in the EBS CloudWatch console.
Avg Write Size (KiB/Operation)	$\text{Avg}(\text{VolumeWriteBytes}) / 1024$ <p>For Nitro-based instances, the following formula derives Average Write Size using CloudWatch Metric Math:</p> $(\text{Sum}(\text{VolumeWriteBytes}) / \text{Sum}(\text{VolumeWriteOps})) / 1024$ <p>The <code>VolumeWriteBytes</code> and <code>VolumeWriteOps</code> metrics are available in the EBS CloudWatch console.</p>
Avg Read Latency (ms/Operation)	$\text{Avg}(\text{VolumeTotalReadTime}) \times 1000$ <p>For Nitro-based instances, the following formula derives Average Read Latency using CloudWatch Metric Math:</p> $(\text{Sum}(\text{VolumeTotalReadTime}) / \text{Sum}(\text{VolumeReadOps})) \times 1000$ <p>The <code>VolumeTotalReadTime</code> and <code>VolumeReadOps</code> metrics are available in the EBS CloudWatch console.</p>
Avg Write Latency (ms/Operation)	$\text{Avg}(\text{VolumeTotalWriteTime}) \times 1000$ <p>For Nitro-based instances, the following formula derives Average Write Latency using CloudWatch Metric Math:</p> $(\text{Sum}(\text{VolumeTotalWriteTime}) / \text{Sum}(\text{VolumeWriteOps})) * 1000$ <p>The <code>VolumeTotalWriteTime</code> and <code>VolumeWriteOps</code> metrics are available in the EBS CloudWatch console.</p>

For the average latency graphs and average size graphs, the average is calculated over the total number of operations (read or write, whichever is applicable to the graph) that completed during the period.

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 **chkdsk**, 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 \(p. 825\)](#).

The following table lists statuses for Amazon EBS volumes.

Volume status	I/O enabled status	I/O performance status (only available for Provisioned IOPS volumes)
ok	Enabled (I/O Enabled or I/O Auto-Enabled)	Normal (Volume performance is as expected)
warning	Enabled (I/O Enabled or I/O Auto-Enabled)	Degraded (Volume performance is below expectations) Severely Degraded (Volume performance is well below expectations)
impaired	Enabled (I/O Enabled or I/O Auto-Enabled) Disabled (Volume is offline and pending recovery, or is waiting for the user to enable I/O)	Stalled (Volume performance is severely impacted) Not Available (Unable to determine I/O performance because I/O is disabled)
insufficient-data	Enabled (I/O Enabled or I/O Auto-Enabled) Insufficient Data	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 'Status Checks' tab selected for a specific volume. A warning message indicates that IO operations have been disabled since 16 hours and 58 minutes ago, due to data inconsistencies. The volume status is listed as 'impaired'. The IO status is 'Disabled' since December 23, 2013, at 7:06:41 PM UTC+2. The description is 'Awaiting Action: Enable IO'. The auto-enabled IO setting is 'Disabled' with an 'Edit' link. Below the volume details, there is a link to 'Find out more' about working with volume status checks and events, and a note about posting 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 \(p. 766\)](#).

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 \(p. 764\)](#).

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 \(p. 766\)](#).

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.

Actions ▾

Filter: Volume resources ▾ All event types ▾ Ongoing and scheduled ▾ Search Events

<input type="checkbox"/>	Resource Name	Resource Type	Resource Id	Availability Zone	Event Type	Event Status
<input type="checkbox"/>		volume	vol-0381c540	us-east-1d	potential-data-i...	Awaiting Action: Enable IO
<input checked="" type="checkbox"/>		volume	vol-3682c675	us-east-1d	potential-data-i...	IO Disabled

Event: vol-3682c675

 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 \(p. 766\)](#). 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 \(p. 767\)](#)

- [Option 2: Perform a Consistency Check on the Volume Using Another Instance \(p. 768\)](#)
- [Option 3: Delete the Volume If You No Longer Need It \(p. 769\)](#)

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**.

Volumes: vol-d882c69b

Description	Status Checks	Monitoring	Tags
IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistency may occur if you enable I/O operations now.			
Volume ID	vol-d882c69b		
Capacity	100 GiB		
Created	November 21, 2013 3:42:01 PM UTC+2		
State	available		
Volume type	io1		
Product codes	-		

- e. In **Enable Volume IO**, choose **Yes, Enable**.
3. Check the data on the volume.
 - a. Run the **chkdsk** 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

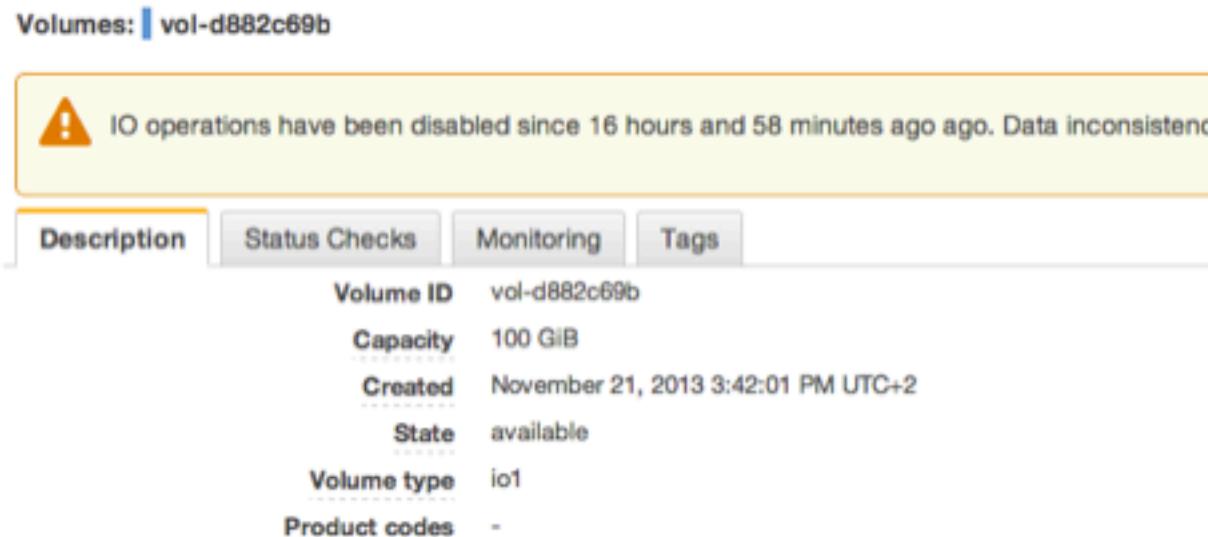
Use the following procedure to check the volume outside your production environment.

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**.



- d. In the **Enable Volume IO** dialog box, choose **Yes, Enable**.
4. Attach the volume to another instance. For information, see [Launch Your Instance \(p. 314\)](#) and [Attaching an Amazon EBS Volume to an Instance \(p. 752\)](#).
5. Check the data on the volume.
 - a. Run the **chkdsk** 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 \(p. 773\)](#).

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 \(p. 751\)](#).

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 \(p. 764\)](#).

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

 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

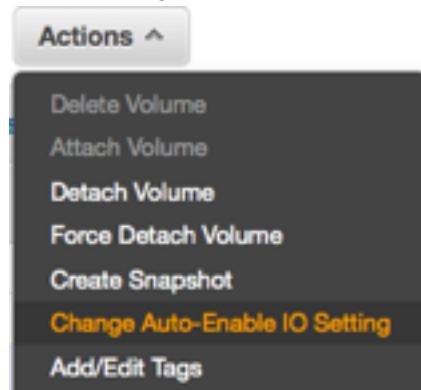
Description Awaiting Action: Enable IO

Auto-Enabled IO Disabled [Edit](#)

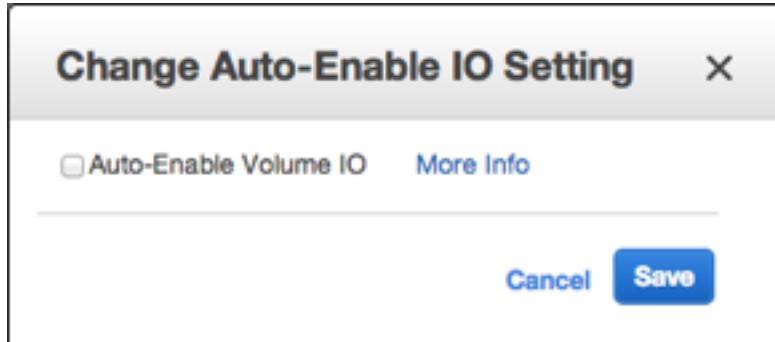
[Find out more](#) about working with volume status checks and events.
If you need technical assistance with your volume, post your issue to the [Developer Forums](#) or visit our [Support forums](#).

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 \(p. 773\)](#).

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 \(p. 756\)](#).

You can reattach a volume that you detached (without unmounting it), but it might not get the same mount point. If there were writes to the volume in progress when it was detached, the data on the volume might be out of sync.

To detach an EBS volume using the console

1. Unmount the volume. Choose **Disk Management**, right-click the volume, and then choose **Change Drive Letter and Path**. Select the mount point and choose **Remove**.

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 \(p. 771\)](#).

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 Windows

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, 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, and 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 \(p. 785\)](#).

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 \(p. 777\)](#) and [Modifying an EBS Volume from the Command Line \(p. 777\)](#).
2. **Monitor the progress of the modification.** For more information, see [Monitoring the Progress of Volume Modifications \(p. 778\)](#).
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 Windows File System after Resizing the Volume \(p. 781\)](#).
4. **If you modify the size of an NVMe volume on an instance that does not have the AWS NVMe drivers, reboot the instance to enable Windows to see the new volume size.** For more information about installing the AWS NVMe drivers, see [AWS NVMe Drivers for Windows Instances \(p. 424\)](#).

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 \(p. 774\)](#)
- [Modifying an EBS Volume from the Console \(p. 777\)](#)
- [Modifying an EBS Volume from the Command Line \(p. 777\)](#)
- [Monitoring the Progress of Volume Modifications \(p. 778\)](#)
- [Extending a Windows File System after Resizing the Volume \(p. 781\)](#)
- [Limitations When Modifying EBS Volumes \(p. 784\)](#)
- [Appendix: Starting and Stopping an Instance to Modify an EBS Volume \(p. 785\)](#)

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, Amazon EBS abstracts the massively distributed storage of a data center into virtual hard disk drives. To an operating system installed on an EC2 instance, an attached EBS volume appears to be a physical hard disk drive containing 512-byte disk sectors. The OS manages the allocation of data blocks (or clusters) onto those virtual sectors through its storage management utilities. The allocation is in conformity with a volume partitioning scheme, such as master boot record (MBR) or GUID partition table (GPT), and within the capabilities of the installed file system (ext4, NTFS, and so on).

EBS is not aware of the data contained in its virtual disk sectors; it only ensures the integrity of the sectors. This means that AWS actions and OS actions are independent of each other. When you are 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 \(p. 774\)](#), 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.

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 \(p. 775\)](#). 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
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}	$8 \text{ ZiB} = 8 \times 1024^3 \text{ TiB}$ (50 TiB certified on RHEL7)	$1 \text{ EiB} = 1024^2 \text{ TiB}$ (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 Windows Volumes

By default, Windows initializes volumes with a master boot record (MBR) partition table. Because MBR supports only volumes smaller than 2 TiB (2,048 GiB), Windows prevents you from resizing MBR volumes beyond this limit. In such a case, the **Extend Volume** option is disabled in the Windows **Disk Management** utility. If you use the AWS Management Console or AWS CLI to create an MBR-partitioned volume that exceeds the size limit, Windows cannot detect or use the additional space. For recommendations for Linux volumes, see [Recommendations for Linux Volumes](#) in the *Amazon EC2 User Guide for Linux Instances*.

To overcome this limitation, you can create a new, larger volume with a GUID partition table (GPT) and copy over the data from the original MBR volume.

To create a GPT volume

1. Create a new, empty volume of the desired size in the Availability Zone of the EC2 instance and attach it to your instance.

Note

The new volume must not be a volume restored from a snapshot.

2. Log in to your Windows system and open **Disk Management** (`diskmgmt.exe`).
3. Open the context (right-click) menu for the new disk and choose **Online**.
4. In the **Initialize Disk** window, select the new disk and choose **GPT (GUID Partition Table)**, **OK**.
5. When initialization is complete, copy the data from the original volume to the new volume, using a tool such as robocopy or teracopy.
6. In **Disk Management**, change the drive letters to appropriate values and take the old volume offline.
7. In the Amazon EC2 console, detach the old volume from the instance, reboot the instance to verify that it functions properly, and delete the old volume.

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 Windows File System after Resizing the Volume \(p. 781\)](#).
6. If you increase the size of an NVMe volume on an instance that does not have the AWS NVMe drivers, you must reboot the instance to enable Windows to see the new volume size. For more information about installing the AWS NVMe drivers, see [AWS NVMe Drivers for Windows Instances \(p. 424\)](#).

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 Windows File System after Resizing the Volume \(p. 781\)](#).

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 or AWS 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 \(p. 777\)](#) 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.

Extending a Windows File System after Resizing the Volume

Use the Windows Disk Management utility to extend the disk size to the new size of the volume. You can begin resizing the file system as soon as the volume enters the Optimizing state.

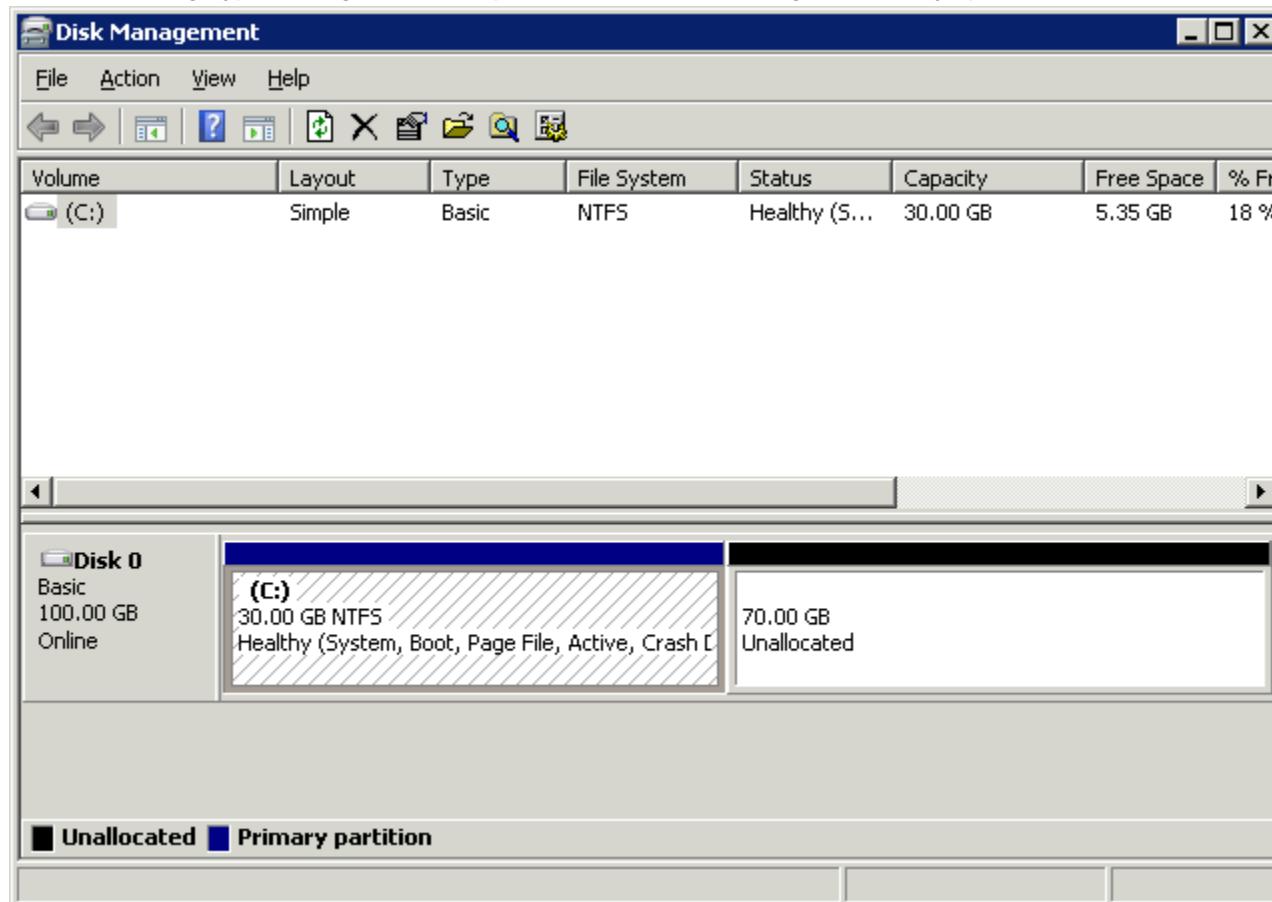
Important

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](#).

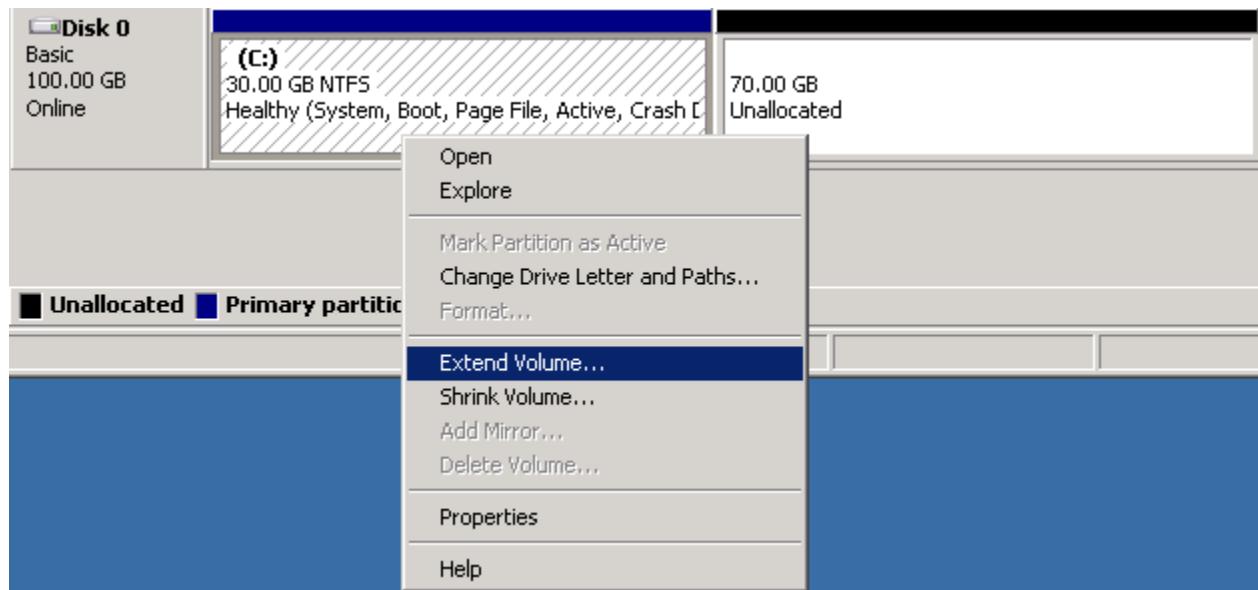
For information about extending a Linux file system, see [Extending a Linux File System after Resizing the Volume](#) in the *Amazon EC2 User Guide for Linux Instances*.

To extend a Windows file system

1. Log in to your Windows instance using Remote Desktop.
2. In the **Run** dialog, type **diskmgmt.msc** and press Enter. The Disk Management utility opens.

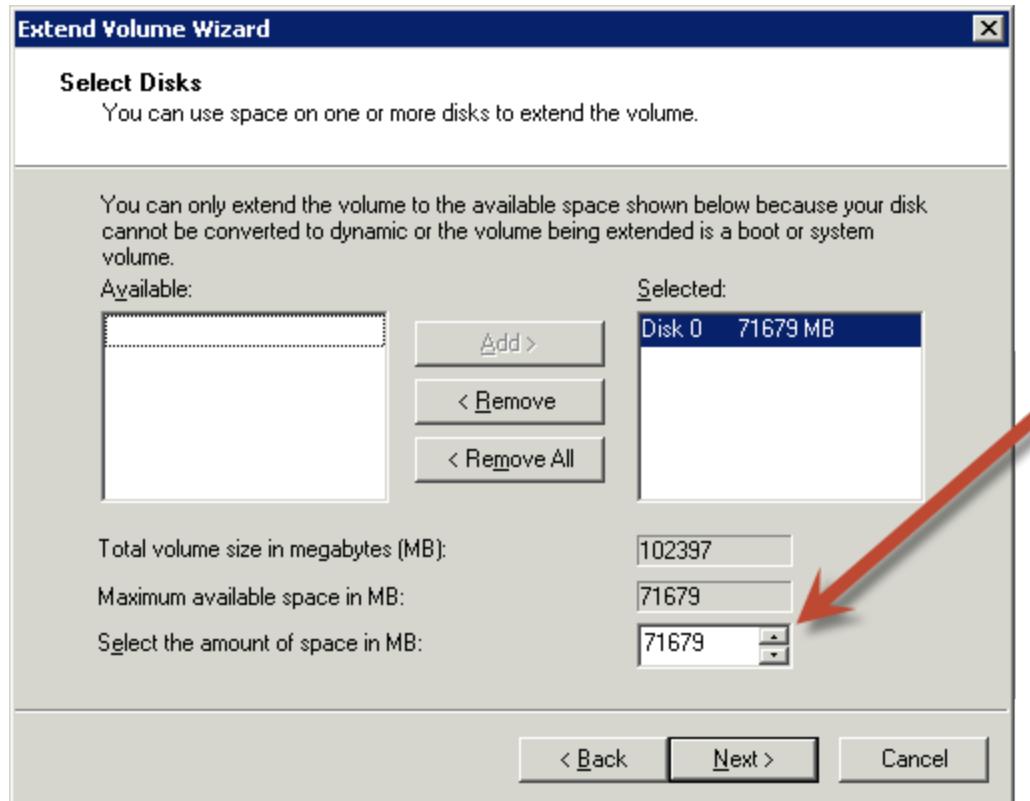


3. On the **Disk Management** menu, choose **Action, Rescan Disks**.
4. Open the context (right-click) menu for the expanded drive and choose **Extend Volume**.



5. In the Extend Volume wizard, choose **Next**. For **Select the amount of space in MB**, enter the number of megabytes by which to extend the volume. Normally, you set this to the maximum available space. The highlighted text under **Selected** is the amount of space that is added, not the final size the volume will have.

Complete the wizard.



6. If you increase the size of an NVMe volume on an instance that does not have the AWS NVMe drivers, you must reboot the instance to enable Windows to see the new volume size. For

more information about installing the AWS NVMe drivers, see [AWS NVMe Drivers for Windows Instances \(p. 424\)](#).

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 \(p. 785\)](#).
- When provisioning over 32,000 IOPS on an existing `io1` volume, you may need to do one of the following in order to see full performance:
 - Detach and attach the volume.
 - Restart the instance.
- Decreasing the size of an EBS volume is not supported. However, you can create a smaller volume and then migrate your data to it using an application-level tool such as robocopy.
- After modifying a volume, wait at least six hours and ensure that the volume is in the `in-use` or `available` state 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 \(p. 785\)](#).

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
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 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 \(p. 364\)](#).

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

5. Modify your EBS volume as described in [Modifying an EBS Volume from the Console \(p. 777\)](#) or [Modifying an EBS Volume from the Command Line \(p. 777\)](#).
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.

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 Windows File System after Resizing the Volume \(p. 781\)](#).

You may have to bring the volume online in order to use it. For more information, see [Making an Amazon EBS Volume Available for Use on Windows \(p. 753\)](#). You do not need to reformat the volume.

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 \(p. 789\)](#).

Note

Using Systems Manager Run Command, you can take application-consistent snapshots of all [Amazon Elastic Block Store \(Amazon EBS\)](#) volumes attached to your Amazon EC2 Windows instances. The snapshot process uses the Windows [Volume Shadow Copy Service \(VSS\)](#) to take image-level backups of VSS-aware applications, including data from pending transactions between these applications and the disk. Furthermore, you don't need to shut down your

instances or disconnect them when you need to back up all attached volumes. For more information, see [Using Run Command to Take VSS-Enabled Snapshots of EBS Volumes](#) in the [AWS Systems Manager User Guide](#).

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 \(p. 787\)](#)
- [Copying and Sharing Snapshots \(p. 789\)](#)
- [Encryption Support for Snapshots \(p. 789\)](#)
- [Creating an Amazon EBS Snapshot \(p. 789\)](#)
- [Deleting an Amazon EBS Snapshot \(p. 791\)](#)
- [Copying an Amazon EBS Snapshot \(p. 793\)](#)
- [Viewing Amazon EBS Snapshot Information \(p. 795\)](#)
- [Sharing an Amazon EBS Snapshot \(p. 796\)](#)
- [Automating the Amazon EBS Snapshot Lifecycle \(p. 798\)](#)

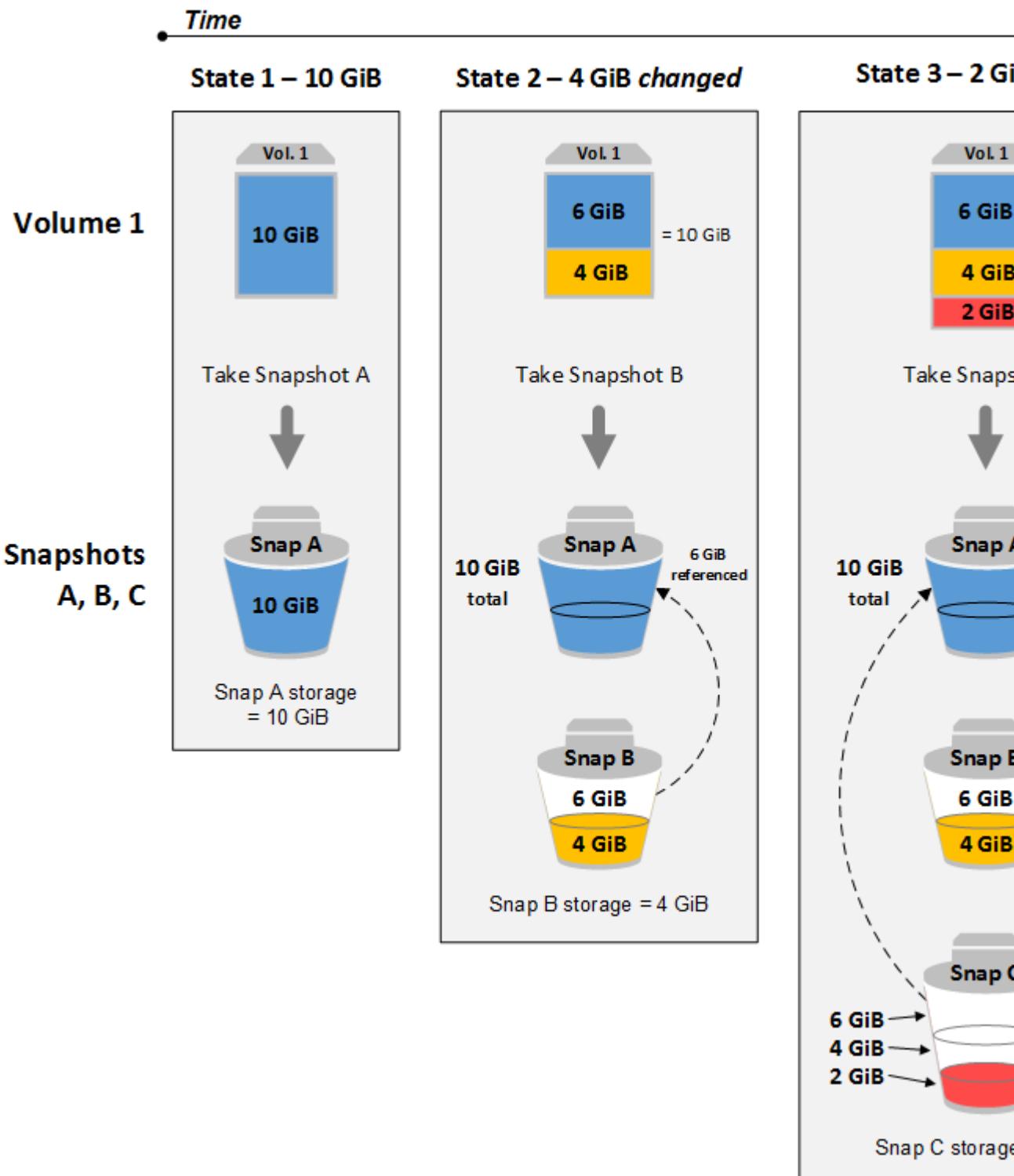
How Incremental Snapshots Work

This section provides illustrations of how an EBS snapshot captures the state of a volume at a point in time, and also how successive snapshots of a changing volume create a history of those changes.

In the diagram below, Volume 1 is shown at three points in time. A snapshot is taken of each of these three volume states.

- In State 1, the volume has 10 GiB of data. Because Snap A is the first snapshot taken of the volume, the entire 10 GiB of data must be copied.
- In State 2, the volume still contains 10 GiB of data, but 4 GiB have changed. Snap B needs to copy and store only the 4 GiB that changed after Snap A was taken. The other 6 GiB of unchanged data, which are already copied and stored in Snap A, are *referenced* by Snap B rather than (again) copied. This is indicated by the dashed arrow.
- In State 3, 2 GiB of data have been added to the volume, for a total of 12 GiB. Snap C needs to copy the 2 GiB that were added after Snap B was taken. As shown by the dashed arrows, Snap C also references 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 12 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 \(p. 791\)](#).

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 \(p. 796\)](#).

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 \(p. 751\)](#). 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 \(p. 793\)](#).

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 \(p. 796\)](#).

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 \(p. 796\)](#).

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 Windows, open Disk Management, right-click the volume to unmount, and select **Change Drive Letter and Path**. Select the mount point to remove, and then click **Remove**.

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 \(p. 890\)](#).

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)

Note

Using Systems Manager Run Command, you can take application-consistent snapshots of all [Amazon Elastic Block Store \(Amazon EBS\)](#) volumes attached to your Amazon EC2 Windows instances. The snapshot process uses the Windows [Volume Shadow Copy Service \(VSS\)](#) to take image-level backups of VSS-aware applications, including data from pending transactions between these applications and the disk. Furthermore, you don't need to shut down your

instances or disconnect them when you need to back up all attached volumes. For more information, see [Using Run Command to Take VSS-Enabled Snapshots of EBS Volumes](#) in the [AWS Systems Manager User Guide](#).

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.

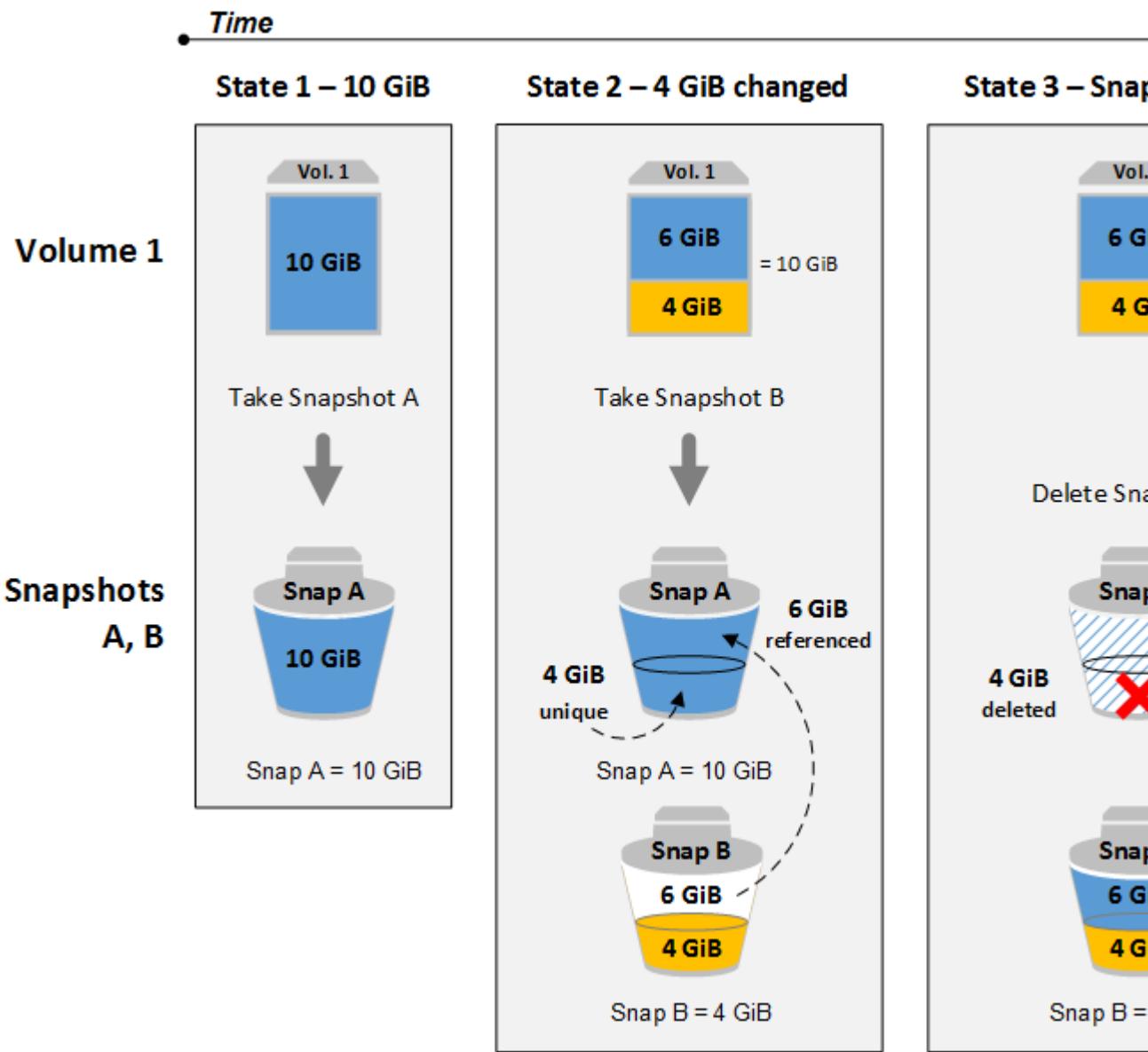
If you make periodic snapshots of a volume, the snapshots are *incremental*, which means that only the blocks on the device that have changed after your last snapshot are saved in the new snapshot. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need 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 \(p. 787\)](#) 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 Windows AMI \(p. 76\)](#).

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 \(p. 796\)](#).

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 \(p. 890\)](#).
- 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 \(p. 796\)](#).

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 \(p. 793\)](#).
- 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 Using the Console

To share a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Snapshots** in the navigation pane.
3. Select the snapshot and then choose **Actions, Modify Permissions**.
4. Make the snapshot public or share it with specific AWS accounts as follows:
 - To make the snapshot public, choose **Public**.

This option is not valid for encrypted snapshots or snapshots with an AWS Marketplace product code.

- To share the snapshot with one or more AWS accounts, choose **Private**, 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 unencrypted snapshot that was privately 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 Using the Console

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

The permissions for a snapshot are specified using the `createVolumePermission` attribute of the snapshot. To make a snapshot public, set the group to `all`. To share a snapshot with a specific AWS account, set the user to the ID of the AWS account.

To modify snapshot permissions using the command line

Use one of the following commands:

- [modify-snapshot-attribute](#) (AWS CLI)
- [Edit-EC2SnapshotAttribute](#) (AWS Tools for Windows PowerShell)

To view snapshot permissions using the command line

Use one of the following commands:

- [describe-snapshot-attribute](#) (AWS CLI)
- [Get-EC2SnapshotAttribute](#) (AWS Tools for Windows PowerShell)

For more information about these command line interfaces, see [Accessing Amazon EC2 \(p. 3\)](#).

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.

All user-defined tags on a source volume can optionally be copied to snapshots created by a policy.

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": "dynamodb:  
                *"  
        }  
    ]  
}
```

```
        "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.
 - **Copy tags**—Copy all user-defined tags on a source volume to snapshots of the volume created by this policy.
 - **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.
 - **Copy tags**—Copy all user-defined tags on a source volume to snapshots of the volume created by this policy.
 - **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  
            },  
            "CopyTags": false  
        }  
    ]  
}
```

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  
                    },  
                    "CopyTags": false,  
                    "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  
            },  
            "CopyTags": false  
        }  
    ]  
}
```

To view the updated policy, use the `get-lifecycle-policy` command. You can see that the state, the value of the tag, the snapshot interval, and the snapshot start time were changed.

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 \(p. 802\)](#). For information about displaying information about lifecycle policies using the CLI, see [Example: Display a lifecycle policy \(p. 803\)](#).

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](#).

AWS CloudFormation

When deploying resource stacks with AWS CloudFormation, you can include Amazon DLM policies in your AWS CloudFormation templates. For more information, see [Amazon Data Lifecycle Manager Resource Types Reference](#).

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 \(p. 736\)](#).

Contents

- [Instance Types that Support EBS Optimization \(p. 807\)](#)
- [Enabling Amazon EBS Optimization at Launch \(p. 814\)](#)
- [Modifying Amazon EBS Optimization for a Running Instance \(p. 815\)](#)

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.

For 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.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
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)
c4.large	Yes	500	62.5	4,000
c4.xlarge	Yes	750	93.75	6,000
c4.2xlarge	Yes	1,000	125	8,000
c4.4xlarge	Yes	2,000	250	16,000
c4.8xlarge	Yes	4,000	500	32,000
c5.large *	Yes	3,500	437.5	20,000
c5.xlarge *	Yes	3,500	437.5	20,000
c5.2xlarge *	Yes	3,500	437.5	20,000
c5.4xlarge	Yes	3,500	437.5	20,000
c5.9xlarge	Yes	7,000	875	40,000
c5.18xlarge	Yes	14,000	1,750	80,000
c5d.large *	Yes	3,500	437.5	20,000
c5d.xlarge *	Yes	3,500	437.5	20,000
c5d.2xlarge *	Yes	3,500	437.5	20,000
c5d.4xlarge	Yes	3,500	437.5	20,000
c5d.9xlarge	Yes	7,000	875	40,000
c5d.18xlarge	Yes	14,000	1,750	80,000
c5n.large *	Yes	3,500	437.5	20,000
c5n.xlarge *	Yes	3,500	437.5	20,000
c5n.2xlarge *	Yes	3,500	437.5	20,000
c5n.4xlarge	Yes	3,500	437.5	20,000
c5n.9xlarge	Yes	7,000	875	40,000
c5n.18xlarge	Yes	14,000	1,750	80,000
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
f1.4xlarge	Yes	3,500	400	44,000
f1.16xlarge	Yes	14,000	1,750	75,000

Amazon Elastic Compute Cloud
User Guide for Windows Instances
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)
g3s.xlarge	Yes	850	100	5,000
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	14,000	1,750	65,000
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	3,500	437.5	18,750
m5.xlarge *	Yes	3,500	437.5	18,750
m5.2xlarge *	Yes	3,500	437.5	18,750
m5.4xlarge	Yes	3,500	437.5	18,750
m5.12xlarge	Yes	7,000	875	40,000
m5.24xlarge	Yes	14,000	1,750	80,000
m5a.large *	Yes	2,120	265	16,000
m5a.xlarge *	Yes	2,120	265	16,000
m5a.2xlarge *	Yes	2,120	265	16,000

Amazon Elastic Compute Cloud
User Guide for Windows Instances
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)
m5a.4xlarge	Yes	2,120	265	16,000
m5a.12xlarge	Yes	5,000	675	30,000
m5a.24xlarge	Yes	10,000	1,250	60,000
m5d.large *	Yes	3,500	437.5	18,750
m5d.xlarge *	Yes	3,500	437.5	18,750
m5d.2xlarge *	Yes	3,500	437.5	18,750
m5d.4xlarge	Yes	3,500	437.5	18,750
m5d.12xlarge	Yes	7,000	875	40,000
m5d.24xlarge	Yes	14,000	1,750	80,000
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
p3dn.24xlarge	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
r5a.large *	Yes	2,210	265	16,000
r5a.xlarge *	Yes	2,210	265	16,000

Amazon Elastic Compute Cloud
User Guide for Windows Instances
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)
r5a.2xlarge *	Yes	2,210	265	16,000
r5a.4xlarge	Yes	2,210	265	16,000
r5a.12xlarge	Yes	5,000	625	30,000
r5a.24xlarge	Yes	10,000	1,250	60,000
r5d.large *	Yes	3,500	437.5	18,750
r5d.xlarge *	Yes	3,500	437.5	18,750
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
t3.nano *	Yes	1,536	192	11,800
t3.micro *	Yes	1,536	192	11,800
t3.small *	Yes	1,536	192	11,800
t3.medium *	Yes	1,536	192	11,800
t3.large *	Yes	2,048	256	15,700
t3.xlarge *	Yes	2,048	256	15,700
t3.2xlarge *	Yes	2,048	256	15,700
u-6tb1.metal	Yes	14,000	1,750	80,000
u-9tb1.metal	Yes	14,000	1,750	80,000
u-12tb1.metal	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

Instance type	EBS-optimized by default	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KB I/O)	Maximum IOPS (16 KB I/O)
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 437.5 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 as shown in the following table:

Instance type	Baseline bandwidth (Mbps)	Baseline throughput (MB/s, 128 KB I/O)	Baseline IOPS (16 KB I/O)
c5.large	525	65.625	4,000
c5.xlarge	800	100	6,000
c5.2xlarge	1,750	218.75	10,000
c5d.large	525	65.625	4,000
c5d.xlarge	800	100	6,000
c5d.2xlarge	1,750	218.75	10,000
c5n.large	525	65.625	4,000
c5n.xlarge	800	100	6,000
c5n.2xlarge	1,750	218.75	10,000
m5.large	480	60	3,600
m5.xlarge	850	106.25	6,000
m5.2xlarge	1,700	212.5	12,000
m5a.large	480	60	3,600
m5a.xlarge	800	100	6,000
m5a.2xlarge	1,166	146	8,333
m5d.large	480	60	3,600
m5d.xlarge	850	106.25	6,000
m5d.2xlarge	1,700	212.5	12,000
r5.large	480	60	3,600
r5.xlarge	850	106.25	6,000

Instance type	Baseline bandwidth (Mbps)	Baseline throughput (MB/s, 128 KB I/O)	Baseline IOPS (16 KB I/O)
r5.2xlarge	1,700	212.5	12,000
r5a.large	480	60	3,600
r5a.xlarge	800	100	6,000
r5a.2xlarge	1,166	146	8,333
r5d.large	480	60	3,600
r5d.xlarge	850	106.25	6,000
r5d.2xlarge	1,700	212.5	12,000
t3.nano	32	4	250
t3.micro	64	8	500
t3.small	128	16	1,000
t3.medium	256	32	2,000
t3.large	512	64	4,000
t3.xlarge	512	64	4,000
t3.2xlarge	512	64	4,000
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 \(p. 498\)](#).

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

Instance type	EBS-optimized by default	Maximum bandwidth (Mbps)	Maximum throughput (MB/s, 128 KB I/O)	Maximum IOPS (16 KB I/O)
i2.xlarge	No	500	62.5	4,000
i2.2xlarge	No	1,000	125	8,000
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. To keep data from instance store volumes, 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 \(p. 816\)](#).

Contents

- [Encryption Key Management \(p. 816\)](#)
- [Supported Instance Types \(p. 816\)](#)
- [Changing the Encryption State of Your Data \(p. 817\)](#)
- [Amazon EBS Encryption and CloudWatch Events \(p. 819\)](#)

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 instance types listed below. You can attach both encrypted and unencrypted volumes to these instance types simultaneously.

- General purpose: M3, M4, M5, M5d, T2, and T3
- Compute optimized: C3, C4, C5, C5d, and C5n
- Memory optimized: cr1.8xlarge, R3, R4, R5, R5d, X1, X1e, and z1d
- Storage optimized: D2, h1.2xlarge, h1.4xlarge, I2, and I3
- Accelerated computing: F1, G2, G3, P2, and P3

- Bare metal: `i3.metal`, `u-6tb1.metal`, `u-9tb1.metal`, and `u-12tb1.metal`

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 \(p. 749\)](#).
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 \(p. 752\)](#).
3. Make the destination volume available by following the procedures in [Making an Amazon EBS Volume Available for Use on Windows \(p. 753\)](#). 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

EBS volumes are exposed as NVMe block devices on Nitro-based instances (p. 113).

Note

The EBS performance guarantees stated in [Amazon EBS Product Details](#) are valid regardless of the block-device interface.

Some of these instance types also support NVMe instance store volumes. For more information, see [NVMe SSD Volumes \(p. 850\)](#).

Identifying the EBS Device

EBS uses single-root I/O virtualization (SR-IOV) to provide volume attachments on Nitro-based instances using the NVMe specification. These devices rely on standard NVMe drivers on the operating system. These drivers typically discover attached devices by scanning the PCI bus during instance boot, and create device nodes based on the order in which the devices respond, not on how the devices are specified in the block device mapping. In Linux, NVMe device names follow the pattern `/dev/nvme<x>n<y>`, where `<x>` is the enumeration order, and, for EBS, `<y>` is 1. Occasionally, devices can respond to discovery in a different order in subsequent instance starts, which causes the device name to change.

We recommend that you use stable identifiers for your EBS volumes within your instance, such as one of the following:

- For Nitro-based instances, the block device mappings that are specified in the Amazon EC2 console when you are attaching an EBS volume or during `AttachVolume` or `RunInstances` API calls are captured in the vendor-specific data field of the NVMe controller identification. With Amazon Linux AMIs later than version 2017.09.01, we provide a `udev` rule that reads this data and creates a symbolic link to the block-device mapping.
- NVMe-attached EBS volumes have the EBS volume ID set as the serial number in the device identification.
- When a device is formatted, a UUID is generated that persists for the life of the filesystem. A device label can be specified at the same time. For more information, see [Making an Amazon EBS Volume Available for Use on Linux](#) and [Booting from the Wrong Volume](#).

You can use the `Get-Disk` command to map Windows disk numbers to EBS volume IDs.

Windows Server 2016

To get the volume IDs, select `AdapterSerialNumber`. In this example, the ID of volume 0 is "vol-0651a78c608e09c6a".

```
PS C:\> Get-Disk | Select Number,AdapterSerialNumber | Sort-Object Number

Number AdapterSerialNumber
-----
0 vol0651a78c608e09c6a
1 vol03f93c68194556d14
2 vol0dbd294c35c6174de
```

Windows Server 2012 R2

To get the volume IDs, select `SerialNumber`. In this example, the ID of volume 0 is "vol-01257d42be427a58b".

```
PS C:\> Get-Disk | Select Number,SerialNumber | Sort-Object Number

Number SerialNumber
-----
0 vol01257d42be427a58b_00000001.
1 vol0da96b723afa69568_00000001.
2 vol0d577fdabd6001831_00000001.
```

Working with NVMe EBS Volumes

The latest AWS Windows AMIs contain the AWS NVMe driver that is required by instance types that expose EBS volumes as NVMe block devices. However, if you resize your root volume on a Windows system, you must rescan the volume in order for this change to be reflected in the instance. If you launched your instance from a different AMI, it might not contain the required AWS NVMe driver. If your instance does not have the latest AWS NVMe driver, you must install it. For more information, see [AWS NVMe Drivers for Windows Instances \(p. 424\)](#).

I/O Operation Timeout

Most operating systems specify a timeout for I/O operations submitted to NVMe devices. On Windows systems, the default timeout is 60 seconds and the maximum is 255 seconds. You can modify the `TimeoutValue` disk class registry setting using the procedure described in [Registry Entries for SCSI Miniport Drivers](#).

- Amazon Linux AMI 2017.09.01 or later
- Canonical 4.4.0-1041 or later
- SLES 12 SP2 (4.4 kernel) or later
- RHEL 7.5 (3.10.0-862 kernel) or later

You can verify the maximum value for your Linux distribution by writing a value higher than the suggested maximum to `/sys/module/nvme_core/parameters/io_timeout` and checking for the `Numerical result out of range` error when attempting to save the file.

Amazon EBS Volume Performance on Windows 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 \(p. 821\)](#)
- [Amazon EC2 Instance Configuration \(p. 822\)](#)
- [I/O Characteristics and Monitoring \(p. 823\)](#)
- [Initializing Amazon EBS Volumes \(p. 825\)](#)
- [RAID Configuration on Windows \(p. 827\)](#)
- [Benchmark EBS Volumes \(p. 832\)](#)

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 \(p. 822\)](#).

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 \(p. 823\)](#).

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.

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 \(p. 825\)](#).

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 \(p. 823\)](#)

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 Windows \(p. 827\)](#).

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 \(p. 757\)](#).

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 \(p. 807\)](#).

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 MiB/s and 250 MiB/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 1,000 MiB/s (at 64,000 IOPS). For more information, see [Amazon EBS Volume Types \(p. 736\)](#).

Note

These performance values for `io1` are guaranteed only for volumes attached to Nitro-based instances. For other instances, AWS guarantees performance up to 500 MiB/s and 32,000 IOPS per volume. For more information, see [Amazon EBS Volume Types](#).

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 64,000 for `io1` volumes and 16,000 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 \(p. 807\)](#).

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 250 MiB/s. If you are using a 256 KiB I/O size, your volume reaches its throughput limit at 1000 IOPS ($1000 \times 256 \text{ KiB} = 250 \text{ MiB}$). For smaller I/O sizes (such as 16 KiB), this same volume can sustain 3,000 IOPS because the throughput is well below 250 MiB/s. (These examples assume that your volume's I/O is not hitting the throughput limits of the instance.) For more information about the throughput limits for each EBS volume type, see [Amazon EBS Volume Types \(p. 736\)](#).

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 \(p. 822\)](#). 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 \(p. 757\)](#). 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 \(p. 762\)](#).

Initializing Amazon EBS Volumes on Windows

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 **dd** or **fio** for Windows to read from all of the blocks on a volume. All existing data on the volume will be preserved.

Before using either tool, gather information about the disks on your system as follows:

1. Use the **wmic** command to list the available disks on your system:

```
wmic diskdrive get size,deviceid
```

The following is example output:

DeviceID	Size
\.\PHYSICALDRIVE2	80517265920
\.\PHYSICALDRIVE1	80517265920
\.\PHYSICALDRIVE0	128849011200
\.\PHYSICALDRIVE3	107372805120

2. Identify the disk to initialize using **dd** or **fio**. The C: drive is on \.\PHYSICALDRIVE0. You can use the **diskmgmt.msc** utility to compare drive letters to disk drive numbers if you are not sure which drive number to use.

Using dd

Complete the following procedures to install and use **dd** to initialize a volume.

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.

Install dd for Windows

The **dd** for Windows program provides a similar experience to the **dd** program that is commonly available for Linux and Unix systems, and it allows you to initialize Amazon EBS volumes that have been restored from snapshots. At the time of this writing, the most recent beta version contains the /dev/null virtual device that is required to initialize volumes restored from snapshots. Full documentation for the program is available at <http://www.chrysocome.net/dd>.

1. Download the most recent binary version of **dd** for Windows from <http://www.chrysocome.net/dd>. You must use version 0.6 beta 3 or newer to initialize restored volumes.
2. (Optional) Create a folder for command line utilities that is easy to locate and remember, such as C:\bin. If you already have a designated folder for command line utilities, you can use that folder instead in the following step.
3. Unzip the binary package and copy the dd.exe file to your command line utilities folder (for example, C:\bin).
4. Add the command line utilities folder to your Path environment variable so you can execute the programs in that folder from anywhere.

Important

The following steps don't update the environment variables in your current command prompt windows. The command prompt windows that you open after you complete these steps will contain the updates. This is why it's necessary for you to open a new command prompt window to verify that your environment is set up properly.

- a. Choose **Start**, open the context (right-click) menu for **Computer**, and then choose **Properties**.
- b. Choose **Advanced system settings, Environment Variables**.
- c. For **System Variables**, select the variable **Path** and choose **Edit**.

- d. For **Variable value**, append a semicolon and the location of your command line utility folder (`;C:\bin\`) to the end of the existing value.
- e. Choose **OK** to close the **Edit System Variable** window.

Initialize a volume using dd for Windows

1. Execute the following command to read all blocks on the specified device (and send the output to the `/dev/null` virtual device). This command safely initializes your existing data.

Important

Incorrect use of **dd** can easily destroy a volume's data. Be sure to follow precisely the example command below. Only the `if=\\.\PHYSICALDRIVE\` parameter will vary depending on the name of the device you are reading.

```
dd if=\\.\PHYSICALDRIVE\ of=/dev/null bs=1M --progress --size
```

Note

You may see an error if **dd** attempts to read beyond the end of the volume. This can be safely ignored.

2. When the operation completes, you are ready to use your new volume. For more information, see [Making an Amazon EBS Volume Available for Use on Windows \(p. 753\)](#).

Using fio

Complete the following procedures to install and use **fio** to initialize a volume.

Install fio for Windows

The **fio** for Windows program provides a similar experience to the **fio** program that is commonly available for Linux and Unix systems, and it allows you to initialize Amazon EBS volumes that have been restored from snapshots. Full documentation for the program is available at <https://github.com/axboe/fio>.

1. Download the [fio MSI installer](#).
2. Install **fio**.

Initialize a volume using fio for Windows

1. Run a command similar to the following to initialize a volume:

```
fio --filename=\\.\PHYSICALDRIVE\ --rw=read --bs=128k --iodepth=32 --direct=1 --name=volume-initialize
```

2. When the operation completes, you are ready to use your new volume. For more information, see [Making an Amazon EBS Volume Available for Use on Windows \(p. 753\)](#).

RAID Configuration on Windows

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. If one of the devices fails, you may be unable to boot the operating system.

If you need to create a RAID array on a Linux instance, see [RAID Configuration on Linux](#) in the *Amazon EC2 User Guide for Linux Instances*.

Contents

- [RAID Configuration Options \(p. 828\)](#)
- [Creating a RAID Array on Windows \(p. 829\)](#)
- [Creating Snapshots of Volumes in a RAID Array \(p. 832\)](#)

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 Windows

Use the following procedure to create the RAID array. Note that you can get directions for Linux instances from [Creating a RAID Array on Linux](#) in the *Amazon EC2 User Guide for Linux Instances*.

To create a RAID array on Windows

1. Create the Amazon EBS volumes for your array. For more information, see [Creating an Amazon EBS Volume \(p. 749\)](#).

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 \(p. 822\)](#).

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 \(p. 752\)](#).
3. Connect to your Windows instance. For more information, see [Connecting to Your Windows Instance \(p. 359\)](#).

4. Open a command prompt and type the **diskpart** command.

```
diskpart

Microsoft DiskPart version 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
On computer: WIN-BM6QPPL51CO
```

5. At the DISKPART prompt, list the available disks with the following command.

```
DISKPART> list disk

Disk ###  Status       Size     Free      Dyn  Gpt
-----  -----
Disk 0   Online        30 GB    0 B
Disk 1   Online        8 GB     0 B
Disk 2   Online        8 GB     0 B
Disk 3   Online        8 GB     0 B
Disk 4   Online        8 GB     0 B
Disk 5   Online       419 GB    0 B
Disk 6   Online       419 GB    0 B
```

Identify the disks you want to use in your array and take note of their disk numbers.

6. Each disk you want to use in your array must be an online dynamic disk that does not contain any existing volumes. Use the following steps to convert basic disks to dynamic disks and to delete any existing volumes.
 - a. Select a disk you want to use in your array with the following command, substituting **n** with your disk number.

```
DISKPART> select disk n

Disk n is now the selected disk.
```

- b. If the selected disk is listed as **Offline**, bring it online by running the **online disk** command.
- c. If the selected disk does not have an asterisk in the **Dyn** column in the previous **list disk** command output, you need to convert it to a dynamic disk.

```
DISKPART> convert dynamic
```

Note

If you receive an error that the disk is write protected, you can clear the read-only flag with the **ATTRIBUTE DISK CLEAR READONLY** command and then try the dynamic disk conversion again.

- d. Use the **detail disk** command to check for existing volumes on the selected disk.

```
DISKPART> detail disk

XENSRV PVDISK SCSI Disk Device
Disk ID: 2D8BF659
Type : SCSI
Status : Online
Path : 0
Target : 1
LUN ID : 0
Location Path : PCIROOT(0)#PCI(0300)#SCSI(P00T01L00)
Current Read-only State : No
Read-only : No
Boot Disk : No
Pagefile Disk : No
Hibernation File Disk : No
Crashdump Disk : No
Clustered Disk : No

Volume ### Ltr Label Fs Type Size Status Info
----- -- -----
Volume 2 D NEW VOLUME FAT32 Simple 8189 MB Healthy
```

Note any volume numbers on the disk. In this example, the volume number is 2. If there are no volumes, you can skip the next step.

- e. (Only required if volumes were identified in the previous step) Select and delete any existing volumes on the disk that you identified in the previous step.

Warning

This destroys any existing data on the volume.

- i. Select the volume, substituting **n** with your volume number.

```
DISKPART> select volume n
Volume n is the selected volume.
```

- ii. Delete the volume.

```
DISKPART> delete volume
DiskPart successfully deleted the volume.
```

- iii. Repeat these substeps for each volume you need to delete on the selected disk.

- f. Repeat [Step 6 \(p. 829\)](#) for each disk you want to use in your array.

7. Verify that the disks you want to use are now dynamic.

```
DISKPART> list disk
```

Disk ###	Status	Size	Free	Dyn	Gpt
Disk 0	Online	30 GB	0 B		
Disk 1	Online	8 GB	0 B	*	
Disk 2	Online	8 GB	0 B	*	
Disk 3	Online	8 GB	0 B	*	
* Disk 4	Online	8 GB	0 B	*	
Disk 5	Online	419 GB	0 B		
Disk 6	Online	419 GB	0 B		

8. Create your raid array. On Windows, a RAID 0 volume is referred to as a striped volume and a RAID 1 volume is referred to as a mirrored volume.

(Striped volumes only) To create a striped volume array on disks 1 and 2, use the following command (note the `stripe` option to stripe the array):

```
DISKPART> create volume stripe disk=1,2
DiskPart successfully created the volume.
```

(Mirrored volumes only) To create a mirrored volume array on disks 3 and 4, use the following command (note the `mirror` option to mirror the array):

```
DISKPART> create volume mirror disk=3,4
DiskPart successfully created the volume.
```

9. Verify your new volume.

```
DISKPART> list volume

Volume ### Ltr Label Fs Type Size Status Info
----- -- -- --
Volume 0 C NTFS Partition 29 GB Healthy System
* Volume 1 RAW Mirror 8190 MB Healthy
Volume 2 RAW Stripe 15 GB Healthy
Volume 5 Z Temporary S NTFS Partition 419 GB Healthy
Volume 6 Y Temporary S NTFS Partition 419 GB Healthy
```

Note that for this example the `Type` column lists a `Mirror` volume and a `Stripe` volume.

10. Select and format your volume so that you can begin using it.

- a. Select the volume you want to format, substituting `n` with your volume number.

```
DISKPART> select volume n
Volume n is the selected volume.
```

- b. Format the volume.

Note

To perform a full format, omit the `quick` option.

```
DISKPART> format quick recommended label="My new volume"
100 percent completed
DiskPart successfully formatted the volume.
```

- c. Assign an available drive letter to your volume.

```
DISKPART> assign letter f
DiskPart successfully assigned the drive letter or mount point.
```

Your new volume is now ready to use.

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 the snapshot has been initiated or the snapshot API returns successfully, it is safe to resume all I/O activity.

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

You can test the performance of Amazon EBS volumes by simulating I/O workloads. The process is as follows:

1. Launch an EBS-optimized instance.
2. Create new EBS volumes.
3. Attach the volumes to your EBS-optimized instance.
4. Configure and mount the block device.
5. Install a tool to benchmark I/O performance.
6. Benchmark the I/O performance of your volumes.
7. Delete your volumes and terminate your instance so that you don't continue to incur charges.

Important

Some of the procedures result in the destruction of existing data on the EBS volumes you benchmark. The benchmarking procedures are intended for use on volumes specially created for testing purposes, not production volumes.

Set Up Your Instance

To get optimal performance from EBS volumes, we recommend that you use an EBS-optimized instance. EBS-optimized instances deliver dedicated throughput between Amazon EC2 and Amazon EBS, with instance. EBS-optimized instances deliver dedicated bandwidth between Amazon EC2 and Amazon EBS, with specifications depending on the instance type. For more information, see [Amazon EBS-Optimized Instances \(p. 807\)](#).

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 \(p. 807\)](#).

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 64,000. For more detailed EBS-volume specifications, see [Amazon EBS Volume Types \(p. 736\)](#). For information about creating an EBS volume, see [Creating an Amazon EBS Volume \(p. 749\)](#). For information about attaching a volume to an instance, see [Attaching an Amazon EBS Volume to an Instance \(p. 752\)](#).

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.

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 \(p. 749\)](#). For information about attaching these volumes to your instance, see [Attaching an Amazon EBS Volume to an Instance \(p. 752\)](#).

Install Benchmark Tools

The following table lists some of the possible tools you can use to benchmark the performance of EBS volumes.

Tool	Description
fio	<p>For benchmarking I/O performance. (Note that fio has a dependency on libaio-devel.)</p> <p>To install fio on Amazon Linux, run the following command:</p> <div style="background-color: #f0f0f0; padding: 5px;"><pre>[ec2-user ~]\$ sudo yum install -y fio</pre></div> <p>To install fio on Ubuntu, run the following command:</p> <div style="background-color: #f0f0f0; padding: 5px;"><pre>sudo apt-get install -y fio</pre></div>
Oracle Orion Calibration Tool	For calibrating the I/O performance of storage systems to be used with Oracle databases.

These benchmarking tools support a wide variety of test parameters. You should use commands that approximate the workloads your volumes will support. These commands provided below are intended as examples to help you get started.

Choosing the Volume Queue Length

Choosing the best volume queue length based on your workload and volume type.

Queue Length on SSD-backed Volumes

To determine the optimal queue length for your workload on SSD-backed volumes, we recommend that you target a queue length of 1 for every 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 value up or down to see what performs best for your application.

Disable C-States

Before you run benchmarking, you should disable processor C-states. Temporarily idle cores in a supported CPU can enter a C-state to save power. When the core is called on to resume processing, a certain amount of time passes until the core is again fully operational. This latency can interfere with processor benchmarking routines. For more information about C-states and which EC2 instance types support them, see [Processor State Control for Your EC2 Instance](#).

Disabling C-States on a Windows System

You can disable C-states on Windows as follows:

1. In PowerShell, get the current active power scheme.

```
C:\> $current_scheme = powercfg /getactivescheme
```

2. Get the power scheme GUID.

```
C:\> (Get-WmiObject -class Win32_PowerPlan -Namespace "root\cimv2\power" -Filter "ElementName='High performance'").InstanceID
```

3. Get the power setting GUID.

```
C:\> (Get-WmiObject -class Win32_PowerSetting -Namespace "root\cimv2\power" -Filter "ElementName='Processor idle disable'").InstanceID
```

4. Get the power setting subgroup GUID.

```
C:\> (Get-WmiObject -class Win32_PowerSettingSubgroup -Namespace "root\cimv2\power" -Filter "ElementName='Processor power management'").InstanceID
```

5. Disable C-states by setting the value of the index to 1. A value of 0 indicates that C-states are disabled.

```
C:\> powercfg /setacvalueindex <power_scheme_guid> <power_setting_subgroup_guid> <power_setting_guid> 1
```

- Set active scheme to ensure the settings are saved.

```
C:\> powercfg /setactive <power_scheme_guid>
```

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 \(p. 825\)](#).

When you are finished testing your volumes, see the following topics for help cleaning up: [Deleting an Amazon EBS Volume \(p. 773\)](#) and [Terminate Your Instance \(p. 368\)](#).

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.

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*. For full API reference, see the [EC2 API Reference](#).

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/01234567-0123-0123-0123-0123456789ab"  
    ],  
    "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": "",  
        "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 `copySnapshot` action started and ended.

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Snapshot Notification",  
    "source": "aws.ec2",  
    "account": "123456789012",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2::us-west-2:snapshot/snap-01234567"  
    ],  
    "detail": {  
        "event": "copySnapshot",  
        "result": "succeeded",  
        "cause": "",  
        "request-id": "",  
        "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",  
        "source": "arn:aws:ec2::eu-west-1:snapshot/snap-76543210",  
        "StartTime": "yyyy-mm-ddThh:mm:ssZ",  
        "EndTime": "yyyy-mm-ddThh:mm:ssZ",  
        "Incremental": "True"  
    }  
}
```

The listing below is an example of a JSON object emitted by EBS after a failed copySnapshot event. The cause for the failure was an invalid source snapshot ID. The value of snapshot_id is the ARN of the failed snapshot. In the detail section, the value of source is the ARN of the source snapshot. StartTime and EndTime represent when the copy-snapshot action started and ended.

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Snapshot Notification",  
    "source": "aws.ec2",  
    "account": "123456789012",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2::us-west-2:snapshot/snap-01234567"  
    ],  
    "detail": {  
        "event": "copySnapshot",  
        "result": "failed",  
        "cause": "Source snapshot ID is not valid",  
        "request-id": "",  
        "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",  
        "source": "arn:aws:ec2::eu-west-1:snapshot/snap-76543210",  
        "StartTime": "yyyy-mm-ddThh:mm:ssZ",  
        "EndTime": "yyyy-mm-ddThh:mm:ssZ"  
    }  
}
```

Share Snapshot (shareSnapshot)

The shareSnapshot event is sent to your AWS account when another account shares a snapshot with it. 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": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Snapshot Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2::us-west-2:snapshot/snap-01234567"  
    ],  
    "detail": {  
        "event": "shareSnapshot",  
        "result": "succeeded",  
        "cause": "",  
        "request-id": "",  
        "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",  
        "source": "012345678901",  
        "StartTime": "yyyy-mm-ddThh:mm:ssZ",  
        "EndTime": "yyyy-mm-ddThh:mm:ssZ"  
    }  
}
```

}

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**.

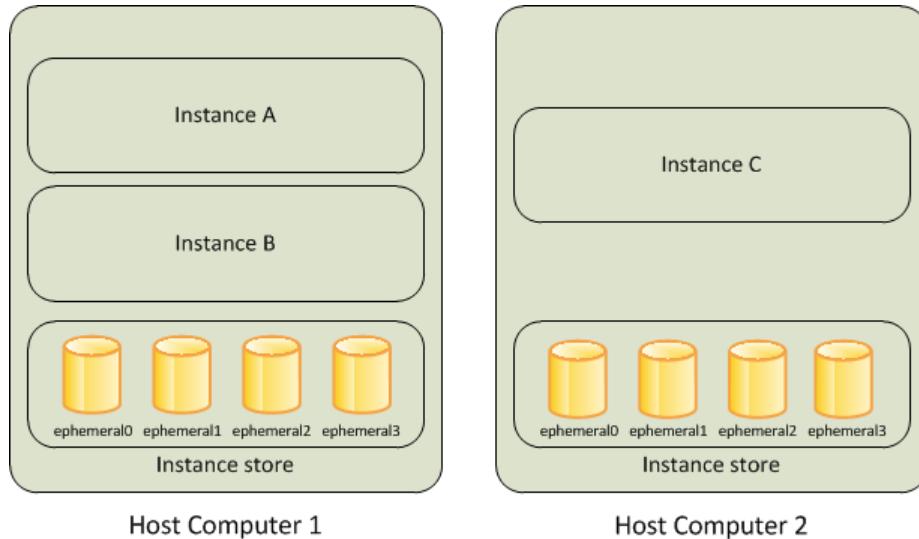
Your rule should now appear on the **Rules** tab. In the example shown, the event that you configured should be emitted by EBS the next time you copy a snapshot.

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.

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 \(p. 844\)](#)
- [Instance Store Volumes \(p. 845\)](#)
- [Add Instance Store Volumes to Your EC2 Instance \(p. 848\)](#)
- [SSD Instance Store Volumes \(p. 850\)](#)

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

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 \(p. 850\)](#).

The following table provides the quantity, size, type, and performance optimizations of instance store volumes available on each supported instance type. For a complete list of instance types, including EBS-only types, see [Amazon EC2 Instance Types](#).

Instance Type	Instance Store Volumes	Type	Needs Initialization*	TRIM Support**
c1.medium	1 x 350 GB	HDD	✓	
c1.xlarge	4 x 420 GB (1.6 TB)	HDD	✓	
c3.large	2 x 16 GB (32 GB)	SSD	✓	
c3.xlarge	2 x 40 GB (80 GB)	SSD	✓	
c3.2xlarge	2 x 80 GB (160 GB)	SSD	✓	
c3.4xlarge	2 x 160 GB (320 GB)	SSD	✓	
c3.8xlarge	2 x 320 GB (640 GB)	SSD	✓	
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 (3.36 TB)	HDD	✓	
cr1.8xlarge	2 x 120 GB (240 GB)	SSD	✓	
d2.xlarge	3 x 2,000 GB (6 TB)	HDD		
d2.2xlarge	6 x 2,000 GB (12 TB)	HDD		
d2.4xlarge	12 x 2,000 GB (24 TB)	HDD		
d2.8xlarge	24 x 2,000 GB (48 TB)	HDD		
f1.2xlarge	1 x 470 GB	NVMe SSD		✓

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Instance Store Volumes

Instance Type	Instance Store Volumes	Type	Needs Initialization*	TRIM Support**
f1.4xlarge	1 x 940 GB	NVMe SSD		✓
f1.16xlarge	4 x 940 GB (3.76 TB)	NVMe SSD		✓
g2.2xlarge	1 x 60 GB	SSD	✓	
g2.8xlarge	2 x 120 GB (240 GB)	SSD	✓	
h1.2xlarge	1 x 2000 GB (2 TB)	HDD		
h1.4xlarge	2 x 2000 GB (4 TB)	HDD		
h1.8xlarge	4 x 2000 GB (8 TB)	HDD		
h1.16xlarge	8 x 2000 GB (16 TB)	HDD		
hs1.8xlarge	24 x 2,000 GB (48 TB)	HDD	✓	
i2.xlarge	1 x 800 GB	SSD		✓
i2.2xlarge	2 x 800 GB (1.6 TB)	SSD		✓
i2.4xlarge	4 x 800 GB (3.2 TB)	SSD		✓
i2.8xlarge	8 x 800 GB (6.4 TB)	SSD		✓
i3.large	1 x 475 GB	NVMe SSD		✓
i3.xlarge	1 x 950 GB	NVMe SSD		✓
i3.2xlarge	1 x 1,900 GB	NVMe SSD		✓
i3.4xlarge	2 x 1,900 GB (3.8 TB)	NVMe SSD		✓
i3.8xlarge	4 x 1,900 GB (7.6 TB)	NVMe SSD		✓
i3.16xlarge	8 x 1,900 GB (15.2 TB)	NVMe SSD		✓
i3.metal	8 x 1,900 GB (15.2 TB)	NVMe SSD		✓
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 (1.6 TB)	HDD	✓	
m2.xlarge	1 x 420 GB	HDD	✓	
m2.2xlarge	1 x 850 GB	HDD	✓	
m2.4xlarge	2 x 840 GB (1.68 TB)	HDD	✓	
m3.medium	1 x 4 GB	SSD	✓	
m3.large	1 x 32 GB	SSD	✓	
m3.xlarge	2 x 40 GB (80 GB)	SSD	✓	

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Instance Store Volumes

Instance Type	Instance Store Volumes	Type	Needs Initialization*	TRIM Support**
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		✓
m5d.4xlarge	2 x 300 GB (600 GB)	NVMe SSD		✓
m5d.12xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓
m5d.24xlarge	4 x 900 GB (3.6 TB)	NVMe SSD		✓
p3dn.24xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓
r3.large	1 x 32 GB	SSD		✓
r3.xlarge	1 x 80 GB	SSD		✓
r3.2xlarge	1 x 160 GB	SSD		✓
r3.4xlarge	1 x 320 GB	SSD		✓
r3.8xlarge	2 x 320 GB (640 GB)	SSD		✓
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 (1.8 TB)	NVMe SSD		✓
r5d.24xlarge	4 x 900 GB (3.6 TB)	NVMe SSD		✓
x1.16xlarge	1 x 1,920 GB	SSD		
x1.32xlarge	2 x 1,920 GB (3.84 TB)	SSD		
x1e.xlarge	1 x 120 GB	SSD		
x1e.2xlarge	1 x 240 GB	SSD		
x1e.4xlarge	1 x 480 GB	SSD		
x1e.8xlarge	1 x 960 GB	SSD		
x1e.16xlarge	1 x 1,920 GB	SSD		
x1e.32xlarge	2 x 1,920 GB (3.84 TB)	SSD		
z1d.large	1 x 75 GB	NVMe SSD		✓
z1d.xlarge	1 x 150 GB	NVMe SSD		✓
z1d.2xlarge	1 x 300 GB	NVMe SSD		✓

Instance Type	Instance Store Volumes	Type	Needs Initialization*	TRIM Support**
z1d.3xlarge	1 x 450 GB	NVMe SSD		✓
z1d.6xlarge	1 x 900 GB	NVMe SSD		✓
z1d.12xlarge	2 x 900 GB (1.8 TB)	NVMe SSD		✓

* Volumes attached to certain instances suffer a first-write penalty unless initialized.

** For more information, see [Instance Store Volume TRIM Support \(p. 851\)](#).

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 \(p. 855\)](#).

A block device mapping always specifies the root volume for the instance. The root volume is mounted automatically.

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 \(p. 734\)](#).

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 \(p. 845\)](#). 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 \(p. 848\)](#)
- [Adding Instance Store Volumes to an Instance \(p. 849\)](#)
- [Making Instance Store Volumes Available on Your Instance \(p. 850\)](#)

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 Windows Instances \(p. 854\)](#).) 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 \(p. 851\)](#). 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 \(p. 863\)](#).

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 \(p. 865\)](#).

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, G2, I2, M3, R3, and X1. For more information about the instance store volumes support by each instance type, see [Instance Store Volumes \(p. 845\)](#).

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

NVMe SSD Volumes

The following instances offer non-volatile memory express (NVMe) SSD instance store volumes: C5d, I3, F1, M5d, p3dn.24xlarge, R5d, and z1d. To access the NVMe volumes, you must use an operating system that supports NVMe. The following are the recommended operating systems:

- Windows Server 2016
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2

After you connect to your instance, you can verify that you see the NVMe volumes in Disk Manager. On the taskbar, open the context (right-click) menu for the Windows logo and choose **Disk Management**. On Windows Server 2008 R2, choose **Start, Administrative Tools, Computer Management, Disk Management**.

If you are using a supported version of Windows Server but you do not see the NVMe devices, verify that the NVMe storage controllers are operational using Device Manager. Expand **Storage controllers** and look for **Standard NVM Express Controller**. If you are using a custom Windows Server 2008 R2 AMI and can't see the instance storage volume, consider installing the Microsoft hotfix for [Native driver support in NVM Express in Windows 7 and Windows Server 2008 R2](#).

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, p3dn.24xlarge, R3, R5d, and z1d.

Instances running Windows Server 2012 R2 support TRIM as of AWS PV Driver version 7.3.0. Instances running earlier versions of Windows Server do not support TRIM.

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 Windows, use the `fsutil behavior set DisableDeleteNotify 1` command..

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 [Amazon Elastic File System \(Amazon EFS\)](#) in the *Amazon EC2 User Guide for Linux Instances* and the [Amazon Elastic File System product page](#).

Important

Amazon EFS is not supported on Windows instances.

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 \(p. 732\)](#).

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.

AWS Tools for Windows PowerShell

Windows instances have the benefit of a graphical browser that you can use to access the Amazon S3 console directly; however, for scripting purposes, Windows users can also use the [AWS Tools for Windows PowerShell](#) to move objects to and from Amazon S3.

Use the following command to copy an Amazon S3 object to your Windows instance.

```
PS C:\> Copy-S3Object -BucketName my_bucket -Key path-to-file -LocalFile my_copied_file.ext
```

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.

```
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.

```
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.

```
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 \(p. 853\)](#)
- [Windows-Specific Volume Limits \(p. 853\)](#)
- [Instance Type Limits \(p. 854\)](#)
- [Bandwidth versus Capacity \(p. 854\)](#)

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 Instances \(p. 414\)](#).

For more information about how device names related to volumes, see [Mapping Disks to Volumes on Your Windows Instance \(p. 864\)](#).

Instance Type Limits

C5, C5d, C5n, M5, M5a, M5d, p3dn.24xlarge, R5, R5a, R5d, T3, and z1d instances support a maximum of 28 attachments, including network interfaces, EBS volumes, and NVMe instance store volumes. Every instance has at least one network interface attachment. NVMe instance store volumes are automatically attached. For example, if you have no additional network interface attachments on an EBS-only instance, you can attach up to 27 EBS volumes to it. If you have one additional network interface on an instance with 2 NVMe instance store volumes, you can attach 24 EBS volumes to it. For more information, see [Elastic Network Interfaces \(p. 665\)](#) and [Instance Store Volumes \(p. 845\)](#).

i3.metal instances support a maximum of 31 EBS volumes.

u-6tb1.metal, u-9tb1.metal, and u-12tb1.metal instances support a maximum of 13 EBS volumes.

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 \(p. 822\)](#) 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 Windows 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 \(p. 854\)](#)
- [Device Name Considerations \(p. 855\)](#)

For information about device names on Linux instances, see [Device Naming on Linux Instances](#) in the [Amazon EC2 User Guide for Linux Instances](#).

Available Device Names

The following table lists the available device names that you can specify in a block device mapping or when attaching an EBS volume. The number of volumes that your instance can support is determined by the operating system. For more information, see [Instance Volume Limits \(p. 853\)](#).

Driver Type	Available	Reserved for Root	Recommended for EBS Volumes	Instance Store Volumes
AWS PV, Citrix PV	xvd[b-z] xvd[b-c][a-z] /dev/sda1	/dev/sda1	xvd[f-z] *	xvd[a-e] xvdc[a-x] (hs1.8xlarge)

Driver Type	Available	Reserved for Root	Recommended for EBS Volumes	Instance Store Volumes
	/dev/sd[b-e]			**
Red Hat PV	xvd[a-z] xvd[b-c][a-z] /dev/sda1 /dev/sd[b-e]	/dev/sda1	xvd[f-p]	xvd[a-e] xvdc[a-x] (hs1.8xlarge)

* If you map an EBS volume with the name xvda, Windows does not recognize the volume.

** NVMe instance store volumes are automatically enumerated and assigned a Windows drive letter.

For more information about instance store volumes, see [Amazon EC2 Instance Store \(p. 843\)](#). For more information about NVMe EBS volumes, see [Amazon EBS and NVMe \(p. 819\)](#).

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.
- AWS Windows AMIs come with additional software that prepares an instance when it first boots up. This is either the Ec2Config service (Windows AMIs prior to Windows Server 2016) or EC2Launch (Windows Server 2016). After the devices have been mapped to drives, they are initialized and mounted. The root drive is initialized and mounted as C:\. The instance store volumes attached to the instance are initialized and mounted as Z:\, Y:\, and so on. By default, when an EBS volume is attached to a Windows instance, it can show up as any drive letter on the instance. You can change the settings to set the drive letters of the volumes per your specifications. For more information, see [Configuring a Windows Instance Using the EC2Config Service \(p. 383\)](#), [Configuring a Windows Instance Using EC2Launch \(p. 374\)](#), and [Mapping Disks to Volumes on Your Windows Instance \(p. 864\)](#).
- The number of NVMe instance store volumes for an instance depends on the size of the instance. NVMe instance store volumes are automatically enumerated and assigned a Windows drive letter.

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 \(p. 752\)](#). 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 [Root Device Volume \(p. 7\)](#).

Contents

- [Block Device Mapping Concepts \(p. 856\)](#)
- [AMI Block Device Mapping \(p. 858\)](#)
- [Instance Block Device Mapping \(p. 860\)](#)

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 \(p. 856\)](#)
- [Block Device Mapping Instance Store Caveats \(p. 857\)](#)
- [Example Block Device Mapping \(p. 857\)](#)
- [How Devices Are Made Available in the Operating System \(p. 858\)](#)

Block Device Mapping Entries

When you create a block device mapping, you specify the following information for each block device that you need to attach to the instance:

- The device name used within Amazon EC2. The block device driver for the instance assigns the actual volume name when mounting the volume. The name assigned can be different from the name that Amazon EC2 recommends. For more information, see [Device Naming on Windows Instances \(p. 854\)](#).
- [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.

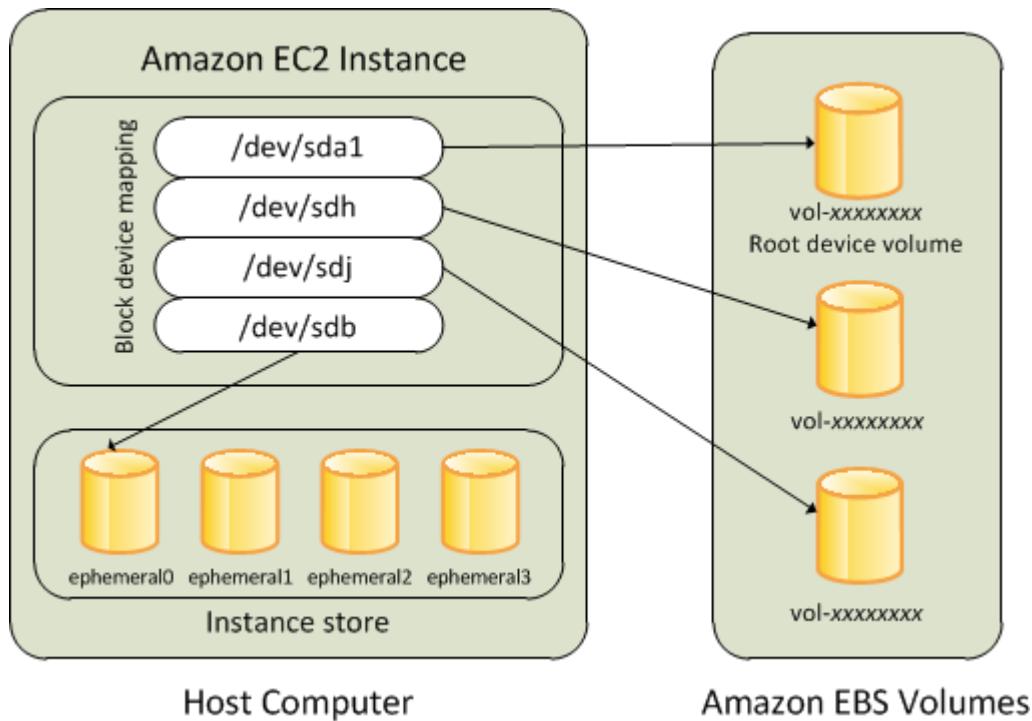
Note

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 \(p. 859\)](#) and [Updating the Block Device Mapping when Launching an Instance \(p. 861\)](#).

How Devices Are Made Available in the Operating System

Device names like /dev/sdh and xvhd 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 Windows instance, the device names specified in the block device mapping are mapped to their corresponding block devices when the instance first boots, and then the Ec2Config service initializes and mounts the drives. The root device volume is mounted as C:\. The instance store volumes are mounted as Z:\, Y:\, and so on. When an EBS volume is mounted, it can be mounted using any available drive letter. However, you can configure how the Ec2Config Service assigns drive letters to EBS volumes; for more information, see [Configuring a Windows Instance Using the EC2Config Service \(p. 383\)](#).

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 \(p. 859\)](#)
- [Viewing the EBS Volumes in an AMI Block Device Mapping \(p. 860\)](#)

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": "xvdb",  
    "VirtualName": "ephemeral0"  
}
```

To add an empty 100 GiB Magnetic volume, use the following mapping:

```
{  
    "DeviceName": "xvdg",  
    "Ebs": {  
        "VolumeSize": 100  
    }  
}
```

To add an EBS volume based on a snapshot, use the following mapping:

```
{  
    "DeviceName": "xvdh",  
    "Ebs": {  
        "SnapshotId": "snap-xxxxxxxx"  
    }  
}
```

To omit a mapping for a device, use the following mapping:

```
{  
    "DeviceName": "xvdj",  
    "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 \(p. 861\)](#)
- [Updating the Block Device Mapping of a Running Instance \(p. 862\)](#)
- [Viewing the EBS Volumes in an Instance Block Device Mapping \(p. 863\)](#)
- [Viewing the Instance Block Device Mapping for Instance Store Volumes \(p. 863\)](#)

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:

- xvdb=ephemeral0
- xvdh=snap-1234567890abcdef0
- xvdj=:100

To prevent xvdj from attaching to an instance launched from this AMI, use the following mapping:

```
{  
    "DeviceName": "xvdj",  
    "NoDevice": ""  
}
```

To increase the size of xvdh to 300 GiB, specify the following mapping. Notice that you don't need to specify the snapshot ID for xvdh, because specifying the device name is enough to identify the volume.

```
{  
    "DeviceName": "xvdh",  
    "Ebs": {  
        "VolumeSize": 300  
    }  
}
```

To attach an additional instance store volume, xvdc, specify the following mapping. If the instance type doesn't support multiple instance store volumes, this mapping has no effect.

```
{  
    "DeviceName": "xvdc",  
    "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`, `xvdh`, and `xvdj`)

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	<code>/dev/sda1</code>
Block devices	<code>/dev/sda1</code> <code>/dev/sdf</code>

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/>.

Important

NVMe instance store volumes are not included in the block device mapping.

First, connect to your running instance. From the instance, use this query to get its block device mapping.

```
PS C:\> Invoke-RestMethod -uri 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.

```
PS C:\> Invoke-RestMethod -uri http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

For more information, see [Instance Metadata and User Data \(p. 452\)](#).

Mapping Disks to Volumes on Your Windows Instance

Your Windows instance comes with an EBS volume that serves as the root volume. If your Windows instance uses AWS PV or Citrix PV drivers, you can optionally add up to 25 volumes, making a total of 26 volumes. For more information, see [Instance Volume Limits \(p. 853\)](#)

Depending on the instance type of your instance, you'll have from 0 to 24 possible instance store volumes available to the instance. To use any of the instance store volumes that are available to your instance, you must specify them when you create your AMI or launch your instance. You can also add EBS volumes when you create your AMI or launch your instance, or attach them while your instance is running. For more information, see [Making an Amazon EBS Volume Available for Use on Windows \(p. 753\)](#).

When you add a volume to your instance, you specify the device name that Amazon EC2 uses. For more information, see [Device Naming on Windows Instances \(p. 854\)](#). AWS Windows Amazon Machine Images (AMIs) contain a set of drivers that are used by Amazon EC2 to map instance store and EBS volumes to Windows disks and drive letters. If you launch an instance from a Windows AMI that uses AWS PV or Citrix PV drivers, you can use the relationships described on this page to map your Windows disks to your instance store and EBS volumes. If your Windows AMI uses Red Hat PV drivers, you can update your instance to use the Citrix drivers. For more information, see [Upgrading PV Drivers on Your Windows Instances \(p. 414\)](#).

Contents

- [Listing the Disks Using Windows Disk Management \(p. 865\)](#)
- [Listing the Disks Using Windows PowerShell \(p. 866\)](#)
- [Disk Device to Device Name Mapping \(p. 868\)](#)

Listing the Disks Using Windows Disk Management

You can find the disks on your Windows instance using Windows Disk Management.

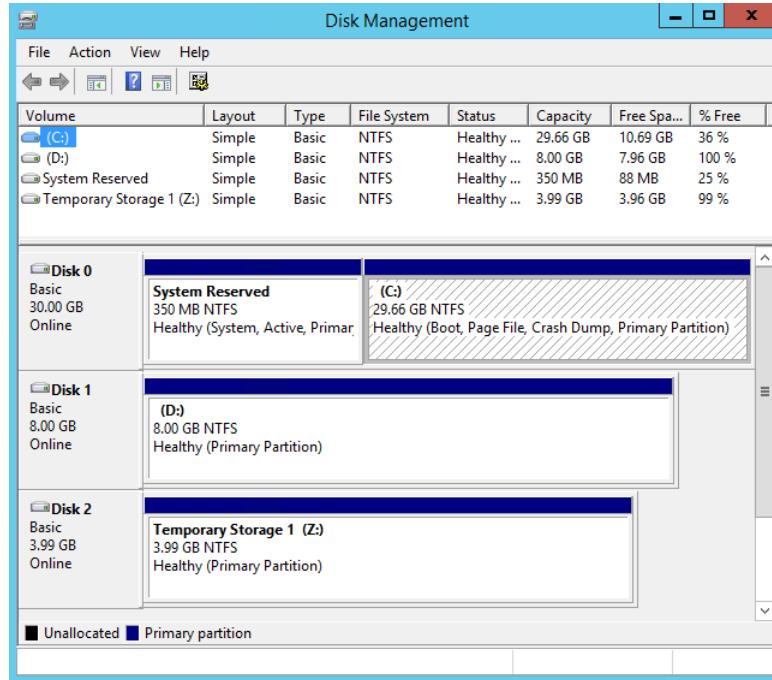
To find the disks on your Windows instance

1. Log in to your Windows instance using Remote Desktop. For more information, see, [Connecting to Your Windows Instance \(p. 359\)](#).
2. Start the Disk Management utility.

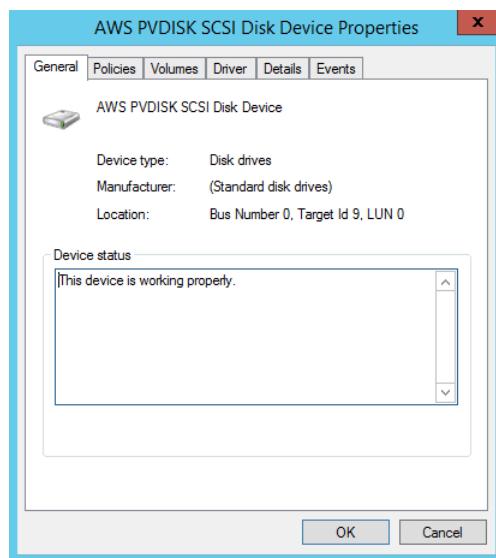
On Windows Server 2012 or Windows Server 2016, on the taskbar, right-click the Windows logo, and then choose **Disk Management**. On Windows Server 2008, choose **Start, Administrative Tools, Computer Management, Disk Management**.

3. Review the disks. The root volume is an EBS volume mounted as C:\. If there are no other disks shown, then you didn't specify additional volumes when you created the AMI or launched the instance.

The following is an example that shows the disks that are available if you launch an m3.medium instance with an instance store volume (Disk 2) and an additional EBS volume (Disk 1).



4. Right-click the gray pane labeled Disk 1, and then select **Properties**. Note the value of **Location** and look it up in the tables in [Disk Device to Device Name Mapping \(p. 868\)](#). For example, the following disk has the location Bus Number 0, Target Id 9, LUN 0. According to the table for EBS volumes, the device name for this location is xvdfj.



- To map the device name of an EBS volume to its volume ID, open the Amazon EC2 console on your computer. In the navigation pane, select **Instances**, and then select your instance. Under **Block devices** on the **Description** tab, click the device name, and locate **EBS ID**. For this example, the volume ID is `vol-0a07f3e37b14708b9`.

Block Device xvdf	
EBS ID	<u>vol-0a07f3e37b14708b9</u>
Root device type	EBS
Attachment time	2016-04-26T09:48:49.000Z
Block device status	attached
Delete on termination	False

Note that the Amazon EC2 console shows only the EBS volumes.

Another way to access volume IDs from the console is to select **Volumes** under **Elastic Block Store** in the navigation pane. Volumes will be listed by instance name and volume size. Review or select the **Attachment Information** to verify the instance to which the volume is attached.

Listing the Disks Using Windows PowerShell

The following PowerShell script lists each disk and its corresponding device name and volume.

```
# List the Windows disks

function Get-EC2InstanceMetadata {
    param([string]$Path)
    (Invoke-WebRequest -Uri "http://169.254.169.254/latest/$Path").Content
}

function Convert-SCSITargetIdToDeviceName {
```

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Listing the Disks Using Windows PowerShell

```
param([int]$SCSITargetId)
If ($SCSITargetId -eq 0) {
    return "sda1"
}
$deviceName = "xvd"
If ($SCSITargetId -gt 25) {
    $deviceName += [char](0x60 + [int]($SCSITargetId / 26))
}
$deviceName += [char](0x61 + $SCSITargetId % 26)
return $deviceName
}

Try {
    $InstanceId = Get-EC2InstanceMetadata "meta-data/instance-id"
    $AZ = Get-EC2InstanceMetadata "meta-data/placement/availability-zone"
    $Region = $AZ.Remove($AZ.Length - 1)
    $BlockDeviceMappings = (Get-EC2Instance -Region $Region -Instance
    $InstanceId).Instances.BlockDeviceMappings
    $VirtualDeviceMap = @{}
    (Get-EC2InstanceMetadata "meta-data/block-device-mapping").Split("`n") | ForEach-Object {
        $VirtualDevice = $_
        $BlockDeviceName = Get-EC2InstanceMetadata "meta-data/block-device-mapping/
$VirtualDevice"
        $VirtualDeviceMap[$BlockDeviceName] = $VirtualDevice
        $VirtualDeviceMap[$VirtualDevice] = $BlockDeviceName
    }
}
Catch {
    Write-Host "Could not access the AWS API, therefore, VolumeId is not available.
Verify that you provided your access keys." -ForegroundColor Yellow
}

Get-disk | ForEach-Object {
    $DiskDrive = $_
    $Disk = $_.Number
    $Partitions = $_.NumberOfPartitions
    $EbsVolumeID = $_.SerialNumber -replace "[^ ]*vol" -replace "vol", "vol-"
    Get-Partition -DiskId $_.Path | ForEach-Object {
        if ($_.DriveLetter -ne "") {
            $DriveLetter = $_.DriveLetter
            $VolumeName = (Get-PSDrive | Where-Object {$_ .Name -eq $DriveLetter}).Description
        }
    }

    If ($DiskDrive.path -like "*PROD_PVDISK*") {
        $BlockDeviceName = Convert-SCSITargetIdToDeviceName((Get-WmiObject -
        Class Win32_Diskdrive | Where-Object {$_ .DeviceID -eq ("\\.\PHYSICALDRIVE" +
        $DiskDrive.Number) }).SCSITargetId)
        $BlockDeviceName = "/dev/" + $BlockDeviceName
        $BlockDevice = $BlockDeviceMappings | Where-Object { $BlockDeviceName -like "*"+
        $_ .DeviceName+"*" }
        $EbsVolumeID = $BlockDevice.Ebs.VolumeId
        $VirtualDevice = If ($VirtualDeviceMap.ContainsKey($BlockDeviceName))
        { $VirtualDeviceMap[$BlockDeviceName] } Else { $null }
    }
    ElseIf ($DiskDrive.path -like "*PROD_AMAZON_EC2_NVME*") {
        $BlockDeviceName = Get-EC2InstanceMetadata "meta-data/block-device-mapping/
ephemeral$((Get-WmiObject -Class Win32_Diskdrive | Where-Object {$_ .DeviceID -eq ("\\.
\PHYSICALDRIVE"+$DiskDrive.Number) }).SCSIPort - 2)"
        $BlockDevice = $null
        $VirtualDevice = If ($VirtualDeviceMap.ContainsKey($BlockDeviceName))
        { $VirtualDeviceMap[$BlockDeviceName] } Else { $null }
    }
    ElseIf ($DiskDrive.path -like "*PROD_AMAZON*") {
        $BlockDevice = ""
    }
}
```

```
$BlockDeviceName = ($BlockDeviceMappings | Where-Object {$_._ebs.VolumeId -eq $EbsVolumeID}).DeviceName
$VirtualDevice = $null
}
Else {
    $BlockDeviceName = $null
    $BlockDevice = $null
    $VirtualDevice = $null
}
New-Object PSObject -Property @{
    Disk          = $Disk;
    Partitions    = $Partitions;
    DriveLetter   = If ($DriveLetter -eq $null) { "N/A" } Else { $DriveLetter };
    EbsVolumeId   = If ($EbsVolumeID -eq $null) { "N/A" } Else { $EbsVolumeID };
    Device        = If ($BlockDeviceName -eq $null) { "N/A" } Else { $BlockDeviceName };
    VirtualDevice = If ($VirtualDevice -eq $null) { "N/A" } Else { $VirtualDevice };
    VolumeName    = If ($VolumeName -eq $null) { "N/A" } Else { $VolumeName };
}
} | Sort-Object Disk | Format-Table -AutoSize -Property Disk, Partitions, DriveLetter,
EbsVolumeId, Device, VirtualDevice, VolumeName
```

Note

This script requires a profile configured in the AWS Tools for PS, or an IAM role attached to the instance.

Before you run this script, be sure to run the following command to enable PowerShell script execution.

```
Set-ExecutionPolicy RemoteSigned
```

Copy the script and save it as a .ps1 file on the Windows instance. If you run the script without setting your access keys, you'll see output similar to the following.

Disk	Partitions	DriveLetter	EbsVolumeId	Device	VirtualDevice	VolumeName
0	0	N/A	N/A	xvdca	ephemeral0	N/A
1	0	N/A	N/A	xvdcb	ephemeral1	N/A
2	1	C:	vol-0064aexamplec838a	/dev/sdal	root	Windows
3	0	N/A	vol-02256example8a4a3	xvdf	ebs2	N/A

If you specified an IAM role with a policy that allows access to Amazon EC2 when you launched the instance, or if you set up your credentials on the Windows instance as described in [Using AWS Credentials in the AWS Tools for Windows PowerShell User Guide](#), you'll get the volume ID for the EBS volumes in the VolumeId column instead of NA.

Disk Device to Device Name Mapping

The block device driver for the instance assigns the actual volume names when mounting volumes.

Mappings

- [Instance Store Volumes \(p. 868\)](#)
- [EBS Volumes \(p. 869\)](#)
- [NVMe EBS Volumes \(p. 870\)](#)

Instance Store Volumes

The following table describes how the Citrix PV and AWS PV drivers map non-NVMe instance store volumes to Windows volumes. The number of available instance store volumes is determined by the instance type. For more information, see [Instance Store Volumes \(p. 845\)](#).

Location	Device Name
Bus Number 0, Target ID 78, LUN 0	xvdca
Bus Number 0, Target ID 79, LUN 0	xvdcb
Bus Number 0, Target ID 80, LUN 0	xvdcc
Bus Number 0, Target ID 81, LUN 0	xvdcd
Bus Number 0, Target ID 82, LUN 0	xvdce
Bus Number 0, Target ID 83, LUN 0	xvdcf
Bus Number 0, Target ID 84, LUN 0	xvdcg
Bus Number 0, Target ID 85, LUN 0	xvdch
Bus Number 0, Target ID 86, LUN 0	xvdci
Bus Number 0, Target ID 87, LUN 0	xvdcj
Bus Number 0, Target ID 88, LUN 0	xvdck
Bus Number 0, Target ID 89, LUN 0	xvdcl

EBS Volumes

The following table describes how the Citrix PV and AWS PV drivers map non-NVME EBS volumes to Windows volumes.

Location	Device Name
Bus Number 0, Target ID 0, LUN 0	/dev/sda1
Bus Number 0, Target ID 1, LUN 0	xvdb
Bus Number 0, Target ID 2, LUN 0	xvdc
Bus Number 0, Target ID 3, LUN 0	xvdd
Bus Number 0, Target ID 4, LUN 0	xvde
Bus Number 0, Target ID 5, LUN 0	xvdf
Bus Number 0, Target ID 6, LUN 0	xvdg
Bus Number 0, Target ID 7, LUN 0	xvdh
Bus Number 0, Target ID 8, LUN 0	xvdi
Bus Number 0, Target ID 9, LUN 0	xvdj
Bus Number 0, Target ID 10, LUN 0	xvdk
Bus Number 0, Target ID 11, LUN 0	xndl
Bus Number 0, Target ID 12, LUN 0	xvdm
Bus Number 0, Target ID 13, LUN 0	xvdn

Location	Device Name
Bus Number 0, Target ID 14, LUN 0	xvdo
Bus Number 0, Target ID 15, LUN 0	xvdp
Bus Number 0, Target ID 16, LUN 0	xvdq
Bus Number 0, Target ID 17, LUN 0	xvdr
Bus Number 0, Target ID 18, LUN 0	xvds
Bus Number 0, Target ID 19, LUN 0	xvdt
Bus Number 0, Target ID 20, LUN 0	xvdu
Bus Number 0, Target ID 21, LUN 0	xvdv
Bus Number 0, Target ID 22, LUN 0	xvdw
Bus Number 0, Target ID 23, LUN 0	xvdx
Bus Number 0, Target ID 24, LUN 0	xvdy
Bus Number 0, Target ID 25, LUN 0	xvdz

NVMe EBS Volumes

With C5 and M5 instances, EBS volumes are exposed as NVMe devices. You can use the [Get-Disk](#) command to map Windows disk numbers to EBS volume IDs. For more information, see [Identifying the EBS Device \(p. 819\)](#).

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 \(p. 870\)](#)
- [Finding Public Data Sets \(p. 871\)](#)
- [Creating a Public Data Set Volume from a Snapshot \(p. 871\)](#)
- [Attaching and Mounting the Public Data Set Volume \(p. 872\)](#)

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 \(p. 871\)](#).

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

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 \(p. 793\)](#).

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 \(p. 736\)](#).
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 \(p. 871\)](#) 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 the 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 \(p. 752\)](#).

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 Windows \(p. 753\)](#).

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 Windows \(p. 773\)](#).

Tutorial: Deploy Storage Spaces Direct (S2D) on Amazon EC2

Storage Spaces Direct (S2D) is a highly-scalable, software-defined storage architecture that enables users to cluster local storage with features in Windows Server 2016. S2D is an alternative to traditional SAN or NAS arrays. It uses built-in Windows features and tools to configure highly-available storage that crosses multiple nodes in a cluster. For more information, see [Storage Spaces Direct](#) in the Microsoft documentation.

The following diagram shows the architecture of S2D on Amazon EC2 Windows.



Skill Level

A basic understanding of Windows Server computing as well as how to create and manage domain-joined Amazon EC2 Windows instances in a VPC is required. Knowledge of the AWS Tools for Windows PowerShell and Windows Failover Clustering is helpful, but not required.

What you will accomplish in this tutorial

- Provision a highly-available storage cluster using [Storage Spaces Direct \(S2D\)](#).
- Provision a fault-tolerant, cluster-shared volume (CSV) on your cluster.

Before you begin

- If you haven't done so already, open <https://aws.amazon.com/> and create an AWS account.
- Create a virtual private cloud (VPC) with a public subnet and two private subnets for your instances. A third, private, subnet should be configured for AWS Directory Service.
- Select one of the latest Amazon Machine Images (AMIs) for Windows Server 2016. You can use this AMI as is, or use it as the basis for your own custom AMI. AWS recommends using the latest public EC2 Windows Server 2016 AMI.
- Create an AWS Directory Service directory. This is no longer a requirement for enabling the Failover Clustering feature in Windows Server 2016. However, this tutorial assumes that your instances will be joined to an Active Directory domain, either on EC2 or AWS Managed Active Directory. For more information, see [Getting Started with AWS Directory Service](#) in the *AWS Directory Service Administration Guide*.
- Install and configure the AWS Tools for Windows PowerShell on your computer. For more information, see the [AWS Tools for Windows PowerShell User Guide](#).

Important considerations

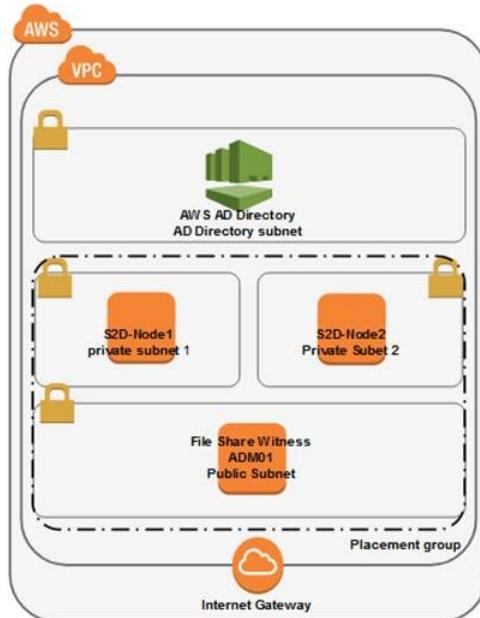
- Stopping instances with [Instance Store Volumes \(p. 843\)](#) can cause data loss if the data is not backed up or replicated. The data in an instance store persists only during the lifetime of its associated instance. If an instance reboots (intentionally or unintentionally), data in the instance store persists. However, data in the instance store is lost under the following circumstances:
 - The underlying disk drive fails.
 - The instance stops.
 - The instance terminates.
- Stopping too many instances in a cluster can cause data loss if the data is not backed up or replicated. When you use S2D on AWS, as with any cluster, losing more nodes than your fault tolerance allows will result in loss of data. One of the biggest risks to any cluster is losing all nodes. Cluster redundancy protects against failures on a single instance (or more, if your fault tolerance supports it). However, you can lose data if the number of instances with failed disk drives in a cluster exceeds the fault tolerance. You can also lose data if the number of stopped or terminated instances exceeds the fault tolerance. To reduce risk, limit the number of people or systems that can stop or terminate instances in the cluster. To mitigate the risk of terminating cluster node instances, [enable termination protection \(p. 369\)](#) on these instances. You can also configure [IAM policies](#) to allow users to only restart nodes from the AWS console but not stop them.
- S2D does not protect against networking or data center failures that affect the entire cluster. To reduce risk, consider using Dedicated Hosts to ensure that instances are not placed in the same rack.

Tasks

- [Step 1: Launch and Domain Join Instances \(p. 874\)](#)

- [Step 2: Install and Configure Instance Prerequisites \(p. 876\)](#)
- [Step 3: Create Failover Cluster \(p. 877\)](#)
- [Step 4: Enable S2D \(p. 878\)](#)
- [Step 5: Provision Storage \(p. 878\)](#)
- [Step 6: Review the S2D Resources \(p. 879\)](#)
- [Step 7: Clean Up \(p. 880\)](#)
- [Additional Resources \(p. 880\)](#)

The following diagram shows the architecture of a two node EC2 Windows S2D Cluster using a file share witness hosted on an existing bastion machine on AWS.



Step 1: Launch and Domain Join Instances

All Nitro instances support Storage Spaces Direct using EBS and/or NVMe. The best performance for storage can be achieved using I3 instances because they provide local instance store with NVMe and high network performance. Configuring S2D on Amazon EC2 requires a cluster of at least two, but no more than 16 instances. These instances must each have at least two NVMe devices with high performance network connections between nodes, and run Windows Server 2016. For more information, see [Storage Spaces Direct hardware requirements](#) in the Microsoft documentation.

We recommend the I3 instance size because it satisfies the [S2D hardware requirements](#) and includes the largest and fastest instance store devices available. It also includes enhanced networking, which maximizes the available resources for S2D per instance. You can use M5D and R5D instance types, which have at least 2 NVMe disks, but local instance store disks will be used as cache disks for the storage spaces direct cluster and at least 2 EBS volumes will have to be added to each instance to provide capacity storage.

We recommend that you launch three instances to take advantage of three-way mirroring [S2D fault tolerance](#), which enables you to conduct maintenance on a single node while maintaining fault tolerance in your cluster if a witness such as a file share witness is configured. You can also use two-way mirroring with two instances as a less expensive solution, but a witness will be necessary and high availability will not be maintained during maintenance on a cluster node.

We will deploy a two node cluster architecture using a file share witness hosted on an existing bastion machine that acts as our administration workstation. Each cluster node must be deployed in a different subnet. This architecture will be deployed into a single availability zone because Microsoft does not currently support stretch cluster with Storage Spaces Direct. However, the performance of a single availability zone and multi-availability zones are exactly the same as a result of our very low-latency and high-bandwidth design for availability zones.

To launch instances for your cluster

1. Using the Amazon EC2 console or the [New-EC2Instance](#) cmdlet, launch two `i3.8xlarge` instances to create the cluster and a `t2.medium` instance as an administration workstation and to host the file share witness. Use a different subnet for each instance. If you wish to follow a logic for IP assignment, then define the primary private IP address at creation time. In this case, you will need to define a secondary private IP address for each cluster node because the secondary IP will be assigned to the cluster VIP later.

To create each instance with PowerShell, use the [New-EC2Instance](#) command.

```
New-EC2Instance -ImageId ami-c49c0dac -MinCount 1 -MaxCount 1 -KeyName myPSKeyPair -  
SecurityGroupId mySGID -InstanceType i3.8xlarge -SubnetId mysubnetID
```

To create an AWS AD directory with PowerShell, use the [New-DSMicrosoftAD](#) command (or, refer to [Create Your AWS Managed Microsoft AD Directory in AWS](#)).

```
New-DSMicrosoftAD -Name corp.example.com -ShortName corp -Password P@ssw0rd -  
Description "AWS DS Managed" - VpcSettings_VpcId vpc-xxxxxxxx -VpcSettings_SubnetId  
subnet-xxxxxxx, subnet-xxxxxxx
```

We use the following S2D-node1 network interface configuration:

▼ Network interfaces ⓘ		Device	Network Interface	Subnet	Primary IP	Secondary IP addresses	IPv6 IPs
eth0	New network interface ▾			subnet-9850a3fe ▾	172.16.1.199	172.16.1.200 Delete	Add IP
						172.16.1.201 Delete	Add IP

Note

Each role deployed on this cluster, such as a SQL Failover Cluster instance or file server, will require additional secondary IP addresses on each node. The exception is the Scale-Out File Server role, which does not require an access point.

We use the following configuration:

Server NetBIOS Name	IP Address	Subnets
S2D-Node1	172.16.1.199 (Primary) 172.16.1.200 (secondary which will be used for the cluster VIP) 172.16.1.201 (secondary which will be used later for a role such as SQL FCI)	AZ1 (e.g., eu-west-1a) – private subnet 1

Server NetBIOS Name	IP Address	Subnets
S2D-Node2	172.16.3.199 (Primary) 172.16.3.200 (secondary which will be used for the cluster VIP) 172.16.3.201 (secondary which will be used later for a role such as SQL FCI)	AZ1 (e.g., eu-west-1a) – private subnet 2
ADM01	Not specified	AZ1 (e.g., eu-west-1a) – public subnet

2. You can use seamless domain join at creation time to join instances to the domain. If you want to join them to the domain after they are launched, use the [Add-Computer](#) command. We recommend using SSM and [AWS Directory Service](#) to [seamlessly join EC2 instances to a domain](#).

The steps in the remainder of this tutorial require execution with a domain account with local administrative privileges on each instance. Rename the instances as you want them before moving to the configuration. Ensure that your security groups and Windows firewalls are properly configured to allow remote PowerShell connection and cluster communications on these nodes.

Step 2: Install and Configure Instance Prerequisites

S2D requires File Services and Failover Clustering Window features, and at least one ten Gbps network interface. We recommend that you configure SMB to use [SMB Multichannel](#), with RSS client connection counts that match the RSS queue count of the enhanced network adapter.

The following steps will be accomplished from the bastion instance ADM01.

To install required Windows features

- Install the File Services and Failover-Clustering Windows features with the management tools on cluster nodes. Install only failover management tools on ADM01.

Note

Change "S2D-Node1" and "S2D-Node2" to reflect the computer names that you set for the two instances; otherwise, the values will not change.

```
$nodes = "S2D-Node1", "S2D-Node2"
foreach ($node in $nodes) {
    Install-WindowsFeature -ComputerName $node -Name File-Services, Failover-Clustering
    -IncludeManagementTools
}
Install-WindowsFeature -Name RSAT-Clustering
```

To configure networking

- Enable multichannel and set the RSS Connection Count.

```
foreach ($node in $nodes) {
    Invoke-Command -ComputerName $node -ScriptBlock {
```

```
[int]$RssQCount = (Get-NetAdapterAdvancedProperty | Where DisplayName -like "Maximum Number of RSS Queues").RegistryValue | Select -First 1
$Params = @{
    EnableMultiChannel           = $true;
    ConnectionCountPerRssNetworkInterface = $RssQCount;
    Confirm                      = $false;
}
Set-SmbClientConfiguration @Params
}
```

2. Configure RSS.

```
foreach ($node in $nodes) {
    Invoke-Command -ComputerName $node -ScriptBlock {
        Get-WmiObject -class Win32_processor | ft systemname, Name, DeviceID,
        NumberOfCores, NumberOfLogicalProcessors
        $maxvcpu = (Get-WmiObject -class Win32_processor).NumberOfLogicalProcessors
        Get-NetAdapter | Set-NetAdapterRss -BaseProcessorNumber 2 -MaxProcessors
        $maxvcpu
    }
}
```

Note

You will see a disconnection message when executing this command because the network adapter restarts after setting the RSS configuration.

Receive Side Scaling (RSS) is a very important technology in networking on Windows. RSS ensures that incoming network traffic is spread among the available processors in the server for processing. If RSS is not used, network processing is bound to one processor, which is limited to approximately 4GBps. Currently, every NIC, by default, enables RSS, but the configuration is not optimized. Every NIC is configured, by default, with "Base Processor" 0, which means it will start processing on processor 0 together with the others NICs. To optimally configure RSS, start at processor 1 so we don't interfere with processes landing default on processor 0.

3. Increase storage space I/O timeout value to 30 seconds (recommended when configured into a guest cluster).

```
foreach ($node in $nodes) {
    Invoke-Command -ComputerName $node -ScriptBlock {
        Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\spaceport
        \Parameters -Name HwTimeout -Value 0x00007530 -Verbose
    }
}
```

4. Reboot all nodes to apply all of the changes.

```
Restart-Computer -ComputerName $nodes -Wait -For Wmi -Force
```

Step 3: Create Failover Cluster

S2D is a feature that is enabled on an existing failover cluster. After you enable S2D on a failover cluster, it takes control of the local storage of each node in the cluster. For this reason, we recommend that you install a cluster with no storage at creation time, and then enable S2D.

When you create a cluster on AWS, you must assign static IP addresses from each subnet from which a node is deployed. From the console, they must be set as secondary private IP addresses on each node. For this tutorial, we configured 172.16.1.200 and 172.16.3.200 upon deployment of each node.

You can verify and review the cluster configuration with the built-in [Test-Cluster](#) command.

Test and verify your cluster configuration

1. Run the [Test-Cluster](#) command with the Storage Spaces Direct, Inventory, Network, and System Configuration tests.

```
$report = Test-Cluster -Node $nodes -Include 'Storage Spaces Direct', 'Inventory', 'Network', 'System Configuration'
```

2. Review the test results.

```
$reportFilePath = $report.FullName  
Start-Process $reportFilePath
```

3. Create the cluster using [New-Cluster](#). Virtual IPs must be assigned a secondary private IP address from the AWS Console to each respective node.

```
$vips = "172.16.1.200", "172.16.3.200"  
New-Cluster -Name S2D -Node $nodes -StaticAddress $vips -NoStorage
```

4. Configure a file share witness.

```
New-Item -ItemType Directory -Path c:\Share\Witness  
[string]$DomainName = (Get-WmiObject win32_computersystem).domain  
New-SmbShare -Name fsw -Path c:\Share\Witness -FullAccess ($DomainName + "\Domain Computers")  
Set-ClusterQuorum -Cluster S2D -FileShareWitness \\$env:COMPUTERNAME\fsw
```

Step 4: Enable S2D

When the cluster is ready, enable S2D on one of the nodes using [Enable-ClusterS2D](#) as follows. Because we have only one type of disk in our setup (local NVMe), we won't use any disks as a cache disk.

1. Enable S2D on i3 instance types using the [Enable-ClusterS2D](#) command.

```
Enable-ClusterS2D -PoolFriendlyName S2DPool -Confirm:$false -SkipEligibilityChecks:$true -CimSession $nodes[0]
```

2. If you are using m5d or r5d instance types with NVMe and EBS, use NVMe disks as cache disks. The command would look like this:

```
Enable-ClusterS2D -PoolFriendlyName S2DPool -CacheDeviceModel "Amazon EC2 NVMe" -Confirm:$false -SkipEligibilityChecks:$true -CimSession $nodes[0]
```

Step 5: Provision Storage

To provision storage, create a storage pool and then create volumes in that pool. To keep things simple, by default, the [Enable-ClusterS2D](#) command creates a pool using all of the disks available in the cluster. With this command we configured the storage pool name as "S2D Pool."

Once volumes are created, they become accessible to every node in the cluster. The volumes can then be assigned to a specific role in the cluster, such as a file server role; or, they can be assigned as [cluster shared volumes](#) (CSV). A CSV is accessible to the entire cluster, which means that every node in this cluster can write-read to this volume.

To improve performance, we recommend you use fixed provisioning and a ReFS file system for CSV. Sector size depends on what type of workloads will be deployed on the cluster. For more information on sector size, see [Cluster Size Recommendations for ReFS and NTFS](#). For improved local read performance, we recommend that you align the CSV with the node hosting your application or workload. You can have multiple CSV and multiple applications spread across nodes.

Create a cluster shared volume (CSV)

- Use the [New-Volume](#) command to create a new 1TB CSV.

```
$Params = @{
    FriendlyName      = 'CSV1';
    FileSystem        = 'CSVFS_ReFS';
    StoragePoolFriendlyName = 'S2DPool';
    Size              = 1TB;
    AllocationUnitSize = 65536;
    ProvisioningType   = 'Fixed';
    CimSession         = $nodes[0];
}
New-Volume @Params
```

Step 6: Review the S2D Resources

The S2D resources that you configured are displayed in the Failover Cluster Manager.

To view your CSV

1. Open Server Manager.
2. Choose **Tools, Failover Cluster Manager**.
3. Expand the name of the cluster, expand **Storage**, and choose **Disks**.

The friendly name, capacity, node hosting the CSV, and other data will be listed. For more information on managing CSVs, see [Use Cluster Shared Volumes in a Failover Cluster](#).

To synthesize a load on your CSV

Use a tool such as [Diskspd Utility](#). Connect to one of the cluster nodes with RDP and run the following with the Diskspd tool.

```
$mycsv = (gci C:\ClusterStorage\ | select -First 1).fullname
.\diskspd.exe -d60 -b4k -o1024 -t32 -L -Sh -r -w50 -W60 -c100G $mycsv\test.dat
```

To view the S2D storage performance of the cluster

Use the [Get-StorageHealthReport](#) command to view the cluster performance on one of the cluster nodes.

1. Open a new PowerShell windows and start your synthesized workload.
2. In your original PowerShell windows, run [Get-StorageSubSystem *cluster* | Get-StorageHealthReport](#) to see the performance results of the storage subsystem while the workload is running.

```
PS C:\> Get-StorageSubSystem *cluster* | Get-StorageHealthReport
```

CPUUsageAverage	:	60.44 %
CapacityPhysicalPooledAvailable	:	9.82 GB
CapacityPhysicalPooledTotal	:	13.82 TB
CapacityPhysicalTotal	:	13.82 TB
CapacityPhysicalUnpooled	:	0 B
CapacityVolumesAvailable	:	1.89 TB
CapacityVolumesTotal	:	2 TB
IOLatencyAverage	:	257.56 ms
IOLatencyRead	:	255.87 ms
IOLatencyWrite	:	259.25 ms
IOPSRead	:	64327.37 /S
IOPSTotal	:	128582.85 /S
IOPSSWrite	:	64255.49 /S
IOThroughputRead	:	251.28 MB/S
IOThroughputTotal	:	502.28 MB/S
IOThroughputWrite	:	251 MB/S
MemoryAvailable	:	477.77 GB
MemoryTotal	:	488 GB

Step 7: Clean Up

If you followed the tutorial to create a highly available storage cluster using S2D in EC2 Windows, you created a Storage Spaces Direct cluster of two instances from a bastion server, which also serves as a file share witness to the cluster. You are charged for each hour or partial hour that you keep your instances running. When you no longer need your cluster, use the EC2 Console or the [AWS Tools for Windows](#) to delete the resources you created for this project. Do this by deleting the cluster from the failover cluster management mmc, terminating the instances, and deleting the computer objects for the cluster and its respective nodes from your Active Directory.

Additional Resources

[Storage Spaces Direct Calculator \(Preview\)](#)

[Planning Storage Spaces Direct](#)

[Storage Spaces Direct Overview](#)

[Fault Tolerance and Storage Efficiency in Storage Spaces Direct](#)

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 \(p. 881\)](#)
- [Resource IDs \(p. 882\)](#)
- [Listing and Filtering Your Resources \(p. 887\)](#)
- [Tagging Your Amazon EC2 Resources \(p. 890\)](#)
- [Amazon EC2 Service Limits \(p. 899\)](#)
- [Amazon EC2 Usage Reports \(p. 901\)](#)

Resource Locations

Some resources can be used in all regions (global), and some resources are specific to the region or Availability Zone in which they reside.

Resource	Type	Description
AWS account	Global	You can use the same AWS account in all regions.
Key pairs	Global or Regional	<p>The key pairs that you create using Amazon EC2 are tied to the region where you created them. You can create your own RSA key pair and upload it to the region in which you want to use it; therefore, you can make your key pair globally available by uploading it to each region.</p> <p>For more information, see Amazon EC2 Key Pairs and Windows Instances (p. 543).</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. 71) .

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. 793) .
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
<pre>elastic-ip-association export-task flow-log image import-task internet-gateway network-acl network-acl- association network-interface network-interface- attachment prefix-list route-table route-table-association security- group subnet subnet-cidr-block-association vpc vpc-cidr- block-association vpc-endpoint vpc-peering-connection vpn-connection vpn- gateway</pre>		

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 \(p. 883\)](#)
- [Modifying Longer ID Settings \(p. 884\)](#)

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 \(p. 887\)](#)
- [Listing Resources Using the Console \(p. 888\)](#)
- [Filtering Resources Using the Console \(p. 889\)](#)
- [Listing and Filtering Using the CLI and API \(p. 890\)](#)

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

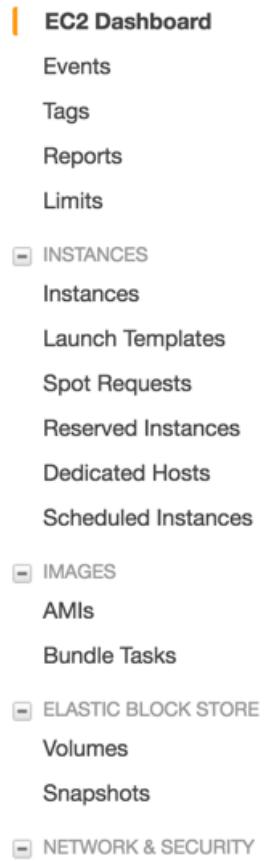
In general, multiple filters with the same key field (for example, tag:Name, search, Instance State) are automatically joined with OR. This is intentional, as the vast majority of filters would not be logical if they were joined with AND. For example, you would get zero results for a search on Instance 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 Windows AMIs backed by Amazon EBS

1. In the navigation pane, choose **AMIs**.
2. In the **Filter** pane, select **Public images**, **EBS images**, and then **Windows** 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

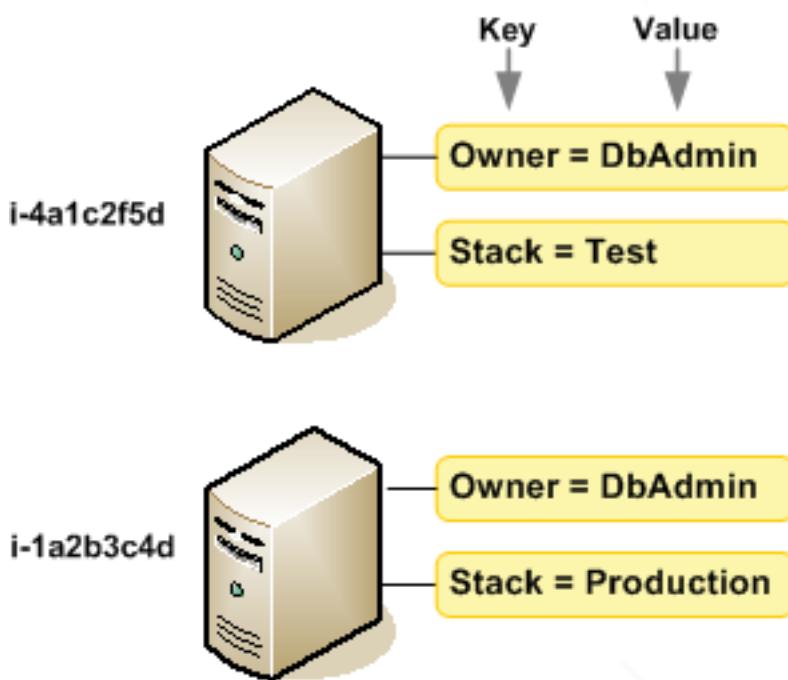
- [Tag Basics \(p. 891\)](#)
- [Tagging Your Resources \(p. 892\)](#)
- [Tag Restrictions \(p. 894\)](#)
- [Tagging Your Resources for Billing \(p. 894\)](#)
- [Working with Tags Using the Console \(p. 895\)](#)
- [Working with Tags Using the CLI or API \(p. 898\)](#)

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 \(p. 562\)](#).

Tagging Your Resources

You can tag most Amazon EC2 resources that already exist in your account. The [table \(p. 892\)](#) below lists the resources that support tagging.

If you're using the Amazon EC2 console, you can apply tags to resources by using the **Tags** tab on the relevant resource screen, or you can use the **Tags** screen. Some resource screens enable you to specify tags for a resource when you create the resource; for example, a tag with a key of `Name` and a value that you specify. In most cases, the console applies the tags immediately after the resource is created (rather than during resource creation). The console may organize resources according to the `Name` tag, but this tag doesn't have any semantic meaning to the Amazon EC2 service.

If you're using the Amazon EC2 API, the AWS CLI, or an AWS SDK, you can use the `CreateTags` EC2 API action to apply tags to existing resources. Additionally, some resource-creating actions enable you to specify tags for a resource when the resource is created. If tags cannot be applied during resource creation, we roll back the resource creation process. This ensures that resources are either created with tags or not created at all, and that no resources are left untagged at any time. By tagging resources at the time of creation, you can eliminate the need to run custom tagging scripts after resource creation.

The following table describes the Amazon EC2 resources that can be tagged, and the resources that can be tagged on creation using the Amazon EC2 API, the AWS CLI, or an AWS SDK.

Tagging Support for Amazon EC2 Resources

Resource	Supports tags	Supports tagging on creation
AFI	Yes	No
AMI	Yes	No
Bundle task	No	No
Capacity Reservation	Yes	Yes
Customer gateway	Yes	No
Dedicated Host	Yes	Yes
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

Resource	Supports tags	Supports tagging on creation
Instance	Yes	Yes
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	Yes	No
Subnet	Yes	No
Transit gateway	Yes	Yes
Transit gateway route table	Yes	Yes
Transit gateway VPC attachment	Yes	Yes
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 \(p. 573\)](#) and [Example Policies for Working with the AWS CLI or an AWS SDK \(p. 601\)](#).

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 \(p. 892\)](#).

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 \(p. 887\)](#).

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 \(p. 895\)](#)
- [Adding and Deleting Tags on an Individual Resource \(p. 896\)](#)
- [Adding and Deleting Tags to a Group of Resources \(p. 896\)](#)
- [Adding a Tag When You Launch an Instance \(p. 897\)](#)
- [Filtering a List of Resources by Tag \(p. 897\)](#)

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.

Manage Tags						
Filter: <input type="text"/> Search Keys <input type="button" value="X"/>		Search Values <input type="button" value="X"/>		1 to 7 of 7 Tags		
	Tag Key	Tag Value	Total	Instances	AMIs	Volumes
Manage Tag	Name	DNS Server	1	1	0	0
Manage Tag	Owner	TeamB	2	0	0	2
Manage Tag	Owner	TeamA	2	0	0	2
Manage Tag	Purpose	Project2	1	0	0	1
Manage Tag	Purpose	Logs	1	0	0	1
Manage Tag	Purpose	Network Management	1	1	0	0
Manage Tag	Purpose	Project1	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 \(p. 881\)](#).
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 \(p. 881\)](#).
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 \(p. 881\)](#).
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 \(p. 881\)](#).
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 \(p. 881\)](#).
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 Windows AMI \(p. 53\)](#).
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 \(p. 887\)](#).

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.

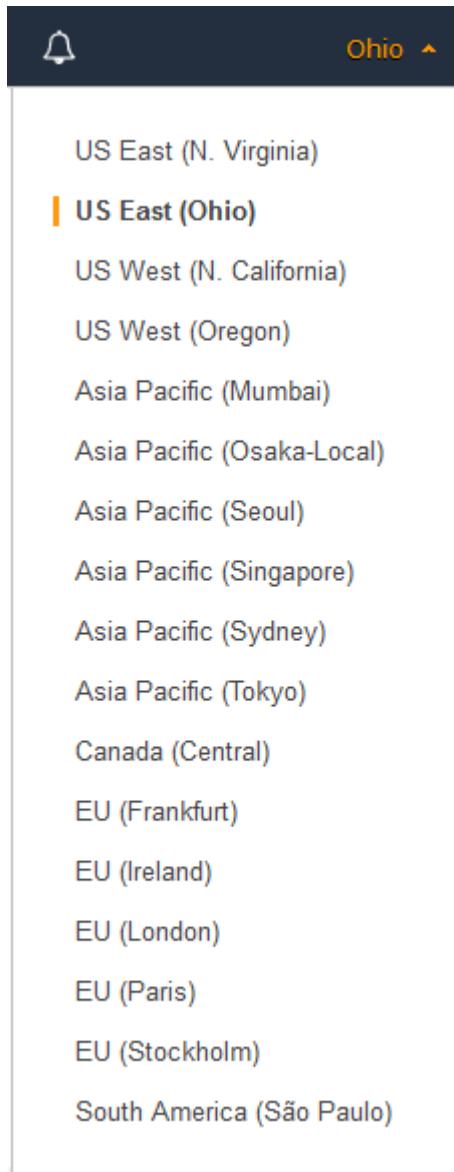
For more information about the limits for other services, see [AWS Service Limits](#) in the *Amazon Web Services General Reference*.

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.

Limits on Email Sent Using Port 25

Amazon EC2 throttles traffic on port 25 of all instances by default. You can request that this throttle be removed. For more information, see [How do I remove the throttle on port 25 from my EC2 instance?](#) in the AWS Knowledge Center.

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

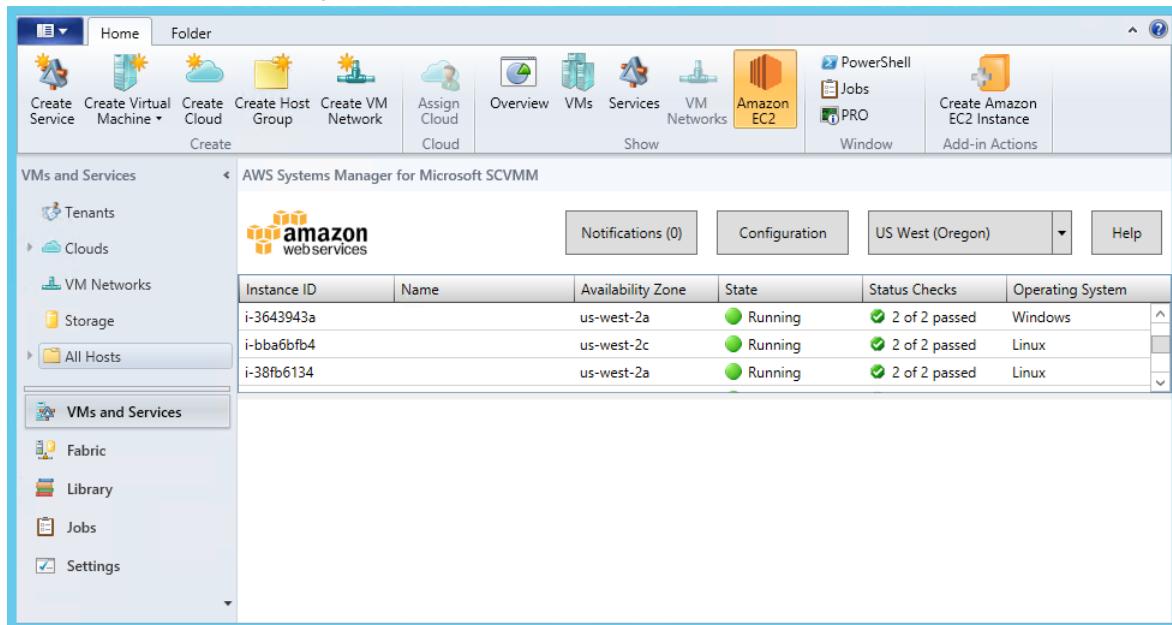
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Reports** and select the report to view.

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](#).

AWS Systems Manager for Microsoft System Center VMM

Amazon Web Services (AWS) Systems Manager for Microsoft System Center Virtual Machine Manager (SCVMM) provides a simple, easy-to-use interface for managing AWS resources, such as EC2 instances, from Microsoft SCVMM. It is implemented as an add-in for the VMM console. For more information, see [AWS Add-ins for Microsoft System Center](#).



Features

- Administrators can grant permissions to users so that they can manage EC2 instances from SCVMM.
- Users can launch, view, reboot, stop, start, and terminate instances, if they have the required permissions.
- Users can get the passwords for their Windows instances and connect to them using RDP.
- Users can get the public DNS names for their Linux instances and connect to them using SSH.
- Users can import their Hyper-V Windows virtual machines from SCVMM to Amazon EC2.

Limitations

- Users must have an account that they can use to log in to SCVMM.
- You can't import Linux virtual machines from SCVMM to Amazon EC2.
- This is not a comprehensive tool for creating and managing AWS resources. The add-in enables SCVMM users to get started quickly with the basic tasks for managing their EC2 instances. Future releases might support managing additional AWS resources.

Requirements

- An AWS account
- Microsoft System Center VMM 2012 R2 or System Center VMM 2012 SP1 with the latest update roll-up

Getting Started

To get started, see the following documentation:

- [Setting Up \(p. 903\)](#)
- [Managing EC2 Instances \(p. 907\)](#)
- [Troubleshooting \(p. 913\)](#)

Setting Up AWS Systems Manager for Microsoft SCVMM

When you set up AWS Systems Manager, users in your organization can access your AWS resources. The process involves creating accounts, deploying the add-in, and providing your credentials.

Tasks

- [Sign Up for AWS \(p. 903\)](#)
- [Set Up Access for Users \(p. 904\)](#)
- [Deploy the Add-In \(p. 906\)](#)
- [Provide Your AWS Credentials \(p. 906\)](#)

Sign Up for AWS

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

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 sign up for an AWS account

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

Note

If you previously signed in to the AWS Management Console using AWS account root user credentials, choose **Sign in to a different account**. If you previously signed in to the console using IAM credentials, choose **Sign-in using root account credentials**. 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 verification code using the phone keypad.

Set Up Access for Users

The first time that you use AWS Systems Manager, you must provide AWS credentials. To enable multiple users to access the same AWS account using unique credentials and permissions, create an IAM user for each user. You can create one or more groups with policies that grant permissions to perform limited tasks. Then you can create one or more IAM users, and add each user to the appropriate group.

To create an Administrators 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. In the **Group Name** box, specify **Administrators** and then choose **Next Step**.
4. On the **Attach Policy** page, select the **AdministratorAccess** AWS managed policy.
5. Choose **Next Step** and then choose **Create Group**.

To create a group with limited access to Amazon EC2

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. In the **Group Name** box, specify a meaningful name for the group and then choose **Next Step**.
4. On the **Attach Policy** page, do not select an AWS managed policy — choose **Next Step**, and then choose **Create Group**.
5. Choose the name of the group you've just created. On the **Permissions** tab, choose **Inline Policies**, and then click **here**.
6. Select the **Custom Policy** radio button and then choose **Select**.
7. Enter a name for the policy and a policy document that grants limited access to Amazon EC2, and then choose **Apply Policy**. For example, you can specify one of the following custom policies.

Grant users in this group permission to view information about EC2 instances only

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:Describe*",  
                "iam>ListInstanceProfiles"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Grant users in this group permission to perform all operations on EC2 instances that are supported by the add-in

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iam>ListInstanceProfiles", "iam:PassRole",  
                "ec2:Describe*", "ec2>CreateKeyPair",  
            ]  
        }  
    ]  
}
```

```
        "ec2:CreateTags", "ec2>DeleteTags",
        "ec2:RunInstances", "ec2:GetPasswordData",
        "ec2:RebootInstances", "ec2:StartInstances",
        "ec2:StopInstances", "ec2:TerminateInstances"
    ],
    "Resource": "*"
}
]
```

Grant users in this group permission to import a VM to Amazon EC2

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3>ListAllMyBuckets", "s3>CreateBucket",
                "s3>DeleteBucket", "s3>DeleteObject",
                "s3>GetBucketLocation", "s3GetObject",
                "s3>ListBucket", "s3PutObject",
                "ec2>DescribeTags", "ec2CancelConversionTask",
                "ec2DescribeConversionTasks", "ec2DescribeInstanceAttribute",
                "ec2CreateImage", "ec2AttachVolume",
                "ec2ImportInstance", "ec2ImportVolume",
                "dynamodbDescribeTable", "dynamodbCreateTable",
                "dynamodbScan", "dynamodbPutItem", "dynamodbUpdateItem"
            ],
            "Resource": "*"
        }
    ]
}
```

To create an IAM user, get the user's AWS credentials, and grant the user permissions

1. In the navigation pane, choose **Users** and then choose **Add user**.
2. Enter a user name.
3. Select the type of access this set of users will have. Select **Programmatic access** and **AWS Management Console access** if this user must also access the AWS Management Console.
4. For **Console password type**, 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 appropriate group.
7. Choose **Next: Review**, then **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 that you want to see. To save the access keys, choose **Download .csv** and then save the file to a safe location.

Note

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**.

Deploy the Add-In

Add-ins for System Center VMM are distributed as .zip files. To deploy the add-in, use the following procedure.

To deploy the add-in

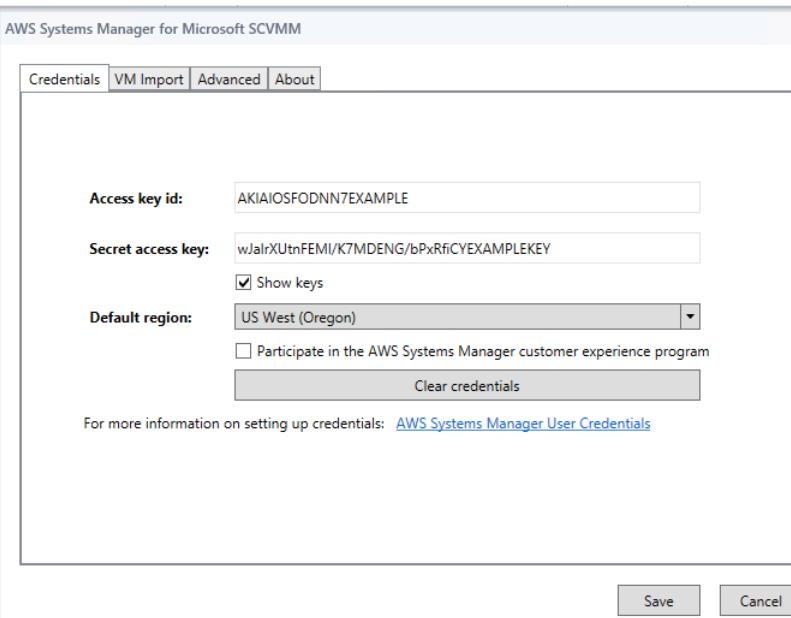
1. From your instance, go to [AWS Systems Manager for Microsoft System Center Virtual Machine Manager](#) and click **SCVMM**. Save the aws-systems-manager-1.5.zip file to your instance.
2. Open the VMM console.
3. In the navigation pane, click **Settings** and then click **Console Add-Ins**.
4. On the ribbon, click **Import Console Add-in**.
5. On the **Select an Add-in** page, click **Browse** and select the aws-systems-manager-1.5.zip file for the add-in that you downloaded.
6. Ignore any warnings that there are assemblies in the add-in that are not signed by a trusted authority. Select **Continue installing this add-in anyway** and then click **Next**.
7. On the **Summary** page, click **Finish**.
8. When the add-in is imported, the status of the job is **Completed**. You can close the **Jobs** window.

Provide Your AWS Credentials

When you use the AWS Systems Manager for the first time, you must provide your AWS credentials. Your access keys identify you to AWS. There are two types of access keys: access key IDs (for example, AKIAIOSFODNN7EXAMPLE) and secret access keys (for example, wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY). You should have stored your access keys in a safe place when you received them.

To provide your AWS credentials

1. Open the VMM console.
2. In the navigation pane, click **VMs and Services**.
3. On the ribbon, click **Amazon EC2**.
4. On the **Credentials** tab, specify your AWS credentials, select a default region, and then click **Save**.



To change these credentials at any time, click **Configuration**.

Managing EC2 Instances Using AWS Systems Manager for Microsoft SCVMM

After you log in to the AWS Systems Manager using your AWS credentials, you can manage your EC2 instances.

Tasks

- [Creating an EC2 Instance \(p. 907\)](#)
- [Viewing Your Instances \(p. 909\)](#)
- [Connecting to Your Instance \(p. 909\)](#)
- [Rebooting Your Instance \(p. 910\)](#)
- [Stopping Your Instance \(p. 910\)](#)
- [Starting Your Instance \(p. 910\)](#)
- [Terminating Your Instance \(p. 910\)](#)

Creating an EC2 Instance

The permissions that you've been granted by your administrator determine whether you can create instances.

Prerequisites

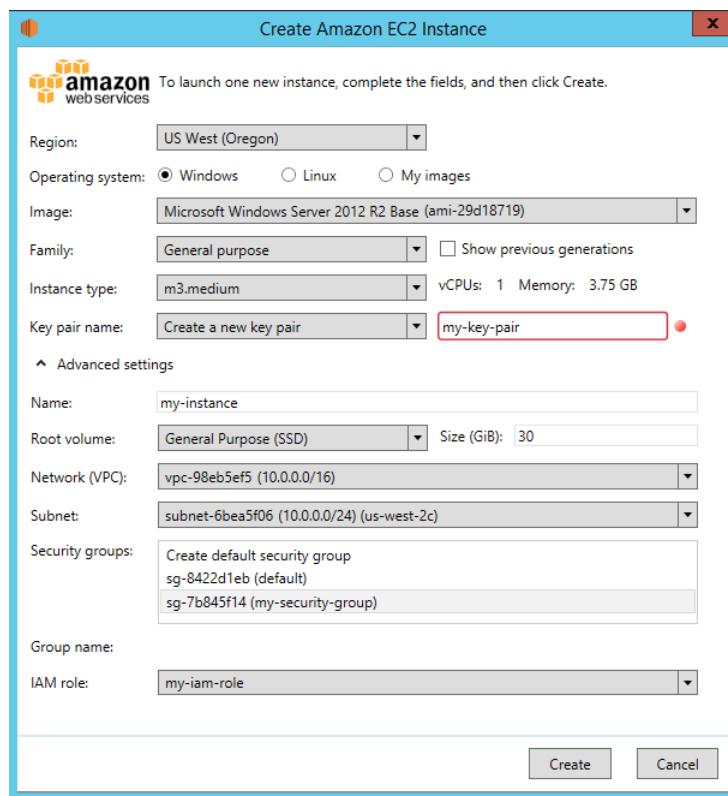
- A virtual private cloud (VPC) with a subnet in the Availability Zone where you'll launch the instance. For more information about creating a VPC, see the [Amazon VPC Getting Started Guide](#).

To create an EC2 instance

1. Open SCVMM.
2. On the ribbon, click **Create Amazon EC2 Instance**.
3. Complete the **Create Amazon EC2 Instance** dialog box as follows:
 - a. Select a region for your instance. By default, we select the region that you configured as your default region.
 - b. Select a template (known as an AMI) for your instance. To use an AMI provided by Amazon, select **Windows** or **Linux** and then select an AMI from **Image**. To use an AMI that you created, select **My images** and then select the AMI from **Image**.
 - c. Select an instance type for the instance. First, select one of the latest instance families from **Family**, and then select an instance type from **Instance type**. To include previous generation instance families in the list, select **Show previous generations**. For more information, see [Amazon EC2 Instances](#) and [Previous Generation Instances](#).
 - d. Create or select a key pair. To create a key pair, select **Create a new key pair** from **Key pair name** and enter a name for the key pair in the highlighted field (for example, `my-key-pair`).
 - e. (Optional) Under **Advanced settings**, specify a display name for the instance.
 - f. (Optional) Under **Advanced settings**, select a VPC from **Network (VPC)**. Note that this list includes all VPCs for the region, including VPCs created using the Amazon VPC console and the default VPC (if it exists). If you have a default VPC in this region, we select it by default. If the

text is "There is no VPC available for launch or import operations in this region", then you must create a VPC in this region using the Amazon VPC console.

- g. (Optional) Under **Advanced settings**, select a subnet from **Subnet**. Note that this list includes all subnets for the selected VPC, including any default subnets. If this list is empty, you must add a subnet to the VPC using the Amazon VPC console, or select a different VPC. Otherwise, we select a subnet for you.
- h. (Optional) Under **Advanced settings**, create a security group or select one or more security groups. If you select **Create default security group**, we create a security group that grants RDP and SSH access to everyone, which you can modify using the Amazon EC2 or Amazon VPC console. You can enter a name for this security group in the **Group name** box.
- i. (Optional) Under **Advanced settings**, select an IAM role. If this list is empty, you can create a role using the IAM console.



4. Click **Create**. If you are creating a key pair, you are prompted to save the .pem file. Save this file in a secure place; you'll need it to log in to your instance. You'll receive confirmation that the instance has launched. Click **Close**.

After you've created your instance, it appears in the list of instances for the region in which you launched it. Initially, the status of the instance is **pending**. After the status changes to **running**, your instance is ready for use.

You can manage the lifecycle of your instance using AWS Systems Manager, as described on this page. To perform other tasks, such as the following, you must use the AWS Management Console:

- Attach an Amazon EBS volume to your instance ([p. 752](#))
- Associate an Elastic IP address with your instance ([p. 663](#))
- Enable termination protection ([p. 369](#))

Viewing Your Instances

The permissions that your administrator grants you determine whether you can view instances and get detailed information about them.

To view your instances and get detailed information

1. Open AWS Systems Manager.
2. From the region list, select a region.
3. From the list of instances, select one or more instances.
4. In the lower pane, click the down arrow next to each instance to view detailed information about the instance.

Virtual machine information		Networking	
Instance ID:	i-343e9f3a	Public DNS name:	
Name:	my-instance	Public IP address:	
State:	Running	Private DNS name:	ip-10-0-0-147.us-west-2.compute.internal
Launch time:	1/20/2015 12:26:48 PM -08:00 (1 minute ago)	Private IP address:	10.0.0.147
Instance type:	m3.medium	Vpc ID:	vpc-f1663d98
Tenancy:	default	Subnet ID:	subnet-c9663da0
Image ID:	ami-29d18719	Network interfaces:	eni-89b0bed0
Operating system:	Windows		

Connecting to Your Instance

You can log in to an EC2 instance if you have the private key (.pem file) for the key pair that was specified when launching the instance. The tool that you'll use to connect to your instance depends on whether the instance is a Windows instance or a Linux instance.

To connect to a Windows EC2 instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance, right-click, and then click **Retrieve Windows Password**.
3. In the **Retrieve Default Windows Administrator Password** dialog box, click **Browse**. Select the private key file for the key pair and then click **Open**.
4. Click **Decrypt Password**. Save the password or copy it to the clipboard.
5. Select the instance, right-click, and then click **Connect via RDP**. When prompted for credentials, use the name of the administrator account and the password that you saved in the previous step.
6. Because the certificate is self-signed, you might get a warning that the security certificate is not from a trusted certifying authority. Click **Yes** to continue.

If the connection fails, see [Troubleshooting Windows Instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

To connect to a Linux EC2 instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. In the lower pane, click the down arrow next to the instance ID to view detailed information about the instance.
4. Locate the public DNS name. You'll need this information to connect to your instance.

5. Connect to the instance using PuTTY. For step-by-step instructions, see [Connect to Your Linux Instance from Windows Using PuTTY](#) in the *Amazon EC2 User Guide for Linux Instances*.

Rebooting Your Instance

The permissions that you've been granted by your administrator determine whether you can reboot instances.

To reboot your instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Reset (Reboot)**.
4. When prompted for confirmation, click **Yes**.

Stopping Your Instance

The permissions that you've been granted by your administrator determine whether you can stop instances.

To stop your instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Shut Down (Stop)**.
4. When prompted for confirmation, click **Yes**.

Starting Your Instance

The permissions that you've been granted by your administrator determine whether you can start instances.

To start your instance

1. Open AWS Systems Manager.
2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Power On (Start)**.
4. When prompted for confirmation, click **Yes**.

If you get a quota error when you try to start an instance, you have reached your concurrent running instance limit. The default limit for your AWS account is 20. If you need additional running instances, complete the form at [Request to Increase Amazon EC2 Instance Limit](#).

Terminating Your Instance

The permissions that you've been granted by your administrator determine whether you can terminate instances.

To terminate your instance

1. Open AWS Systems Manager.

2. From the list of instances, select the instance.
3. Right-click the instance, and then click **Delete (Terminate)**.
4. When prompted for confirmation, click **Yes**.

Importing Your Virtual Machine Using AWS Systems Manager for Microsoft SCVMM

You can launch an EC2 instance from a virtual machine that you import from SCVMM to Amazon EC2.

Important

You can't import Linux virtual machines from SCVMM to Amazon EC2.

Contents

- [Prerequisites \(p. 911\)](#)
- [Importing Your Virtual Machine \(p. 911\)](#)
- [Checking the Import Task Status \(p. 912\)](#)
- [Backing Up Your Imported Instance \(p. 913\)](#)

Prerequisites

- Ensure that your VM is ready. For more information, see [Prepare Your VM](#) in the *VM Import/Export User Guide*.
- In AWS Systems Manager, click **Configuration**, select the **VM Import** tab, and review the following settings:
 - **S3 bucket prefix:** We create a bucket for disk images to be uploaded before they are imported. The name of the bucket starts with the prefix listed here and includes the region (for example, `us-east-2`). To delete the disk images after they are imported, select **Clean up S3 bucket after import**.
 - **VM image export path:** A location for the disk images exported from the VM. To delete the disk images after they are imported, select **Clean up export path after import**.
 - **Alternate Hyper-V PowerShell module path:** The location of the Hyper-V PowerShell module, if it's not installed in the standard location. For more information, see [Installing the Hyper-V Management Tools](#) in the Microsoft TechNet Library.

Importing Your Virtual Machine

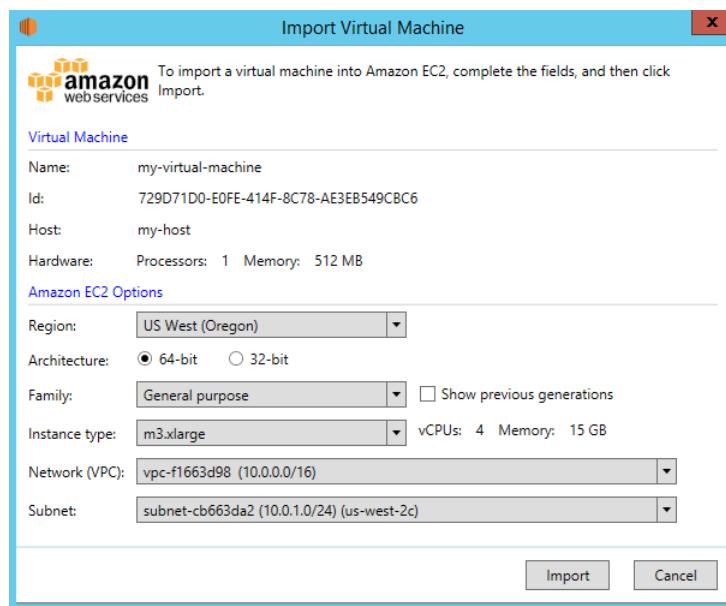
The permissions that you've been granted by your administrator determine whether you can import HyperV Windows virtual machines from SCVMM to AWS.

To import your virtual machine

1. Open SCVMM.
2. On the ribbon, click **VMs**. Select your virtual machine from the list.
3. On the ribbon, click **Import VM to Amazon EC2**.
4. Complete the **Import Virtual Machine** dialog box as follows:
 - a. Select a region for the instance. By default, we select the region that you configured as your default region.
 - b. Select an instance type for the instance. First, select one of the latest instance families from **Family**, and then select an instance type from **Instance type**. To include previous generation

instance families in the list, select **Show previous generations**. For more information, see [Amazon EC2 Instances](#) and [Previous Generation Instances](#).

- c. Select a VPC from **Network (VPC)**. Note that this list includes all VPCs for the region, including VPCs created using the Amazon VPC console and the default VPC (if it exists). If you have a default VPC in this region, we select it by default. If the text is "There is no VPC available for launch or import operations in this region", then you must create a VPC in this region using the Amazon VPC console.
- d. Select a subnet from **Subnet**. Note that this list includes all subnets for the selected VPC, including any default subnets. If this list is empty, you must add a subnet to the VPC using the Amazon VPC console, or select a different VPC. Otherwise, we select a subnet for you.



5. Click **Import**. If you haven't specified the required information in the **VM Import** tab, you'll receive an error asking you to provide the required information. Otherwise, you'll receive confirmation that the import task has started. Click **Close**.

Checking the Import Task Status

The import task can take several hours to complete. To view the current status, open AWS System Manager and click **Notifications**.

You'll receive the following notifications as the import task progresses:

- Import VM: Created Import VM Task
- Import VM: Export VM Disk Image Done
- Import VM: Upload to S3
- Import VM: Image Conversion Starting
- Import VM: Image Conversion Done
- Import VM: Import Complete

Note that you'll receive the **Import VM: Upload to S3**, **Import VM: Image Conversion Starting**, and **Import VM: Image Conversion Done** notifications for each disk image converted.

If the import task fails, you'll receive the notification `Import VM: Import Failed`. For more information about troubleshooting issues with import tasks, see [Errors Importing a VM \(p. 914\)](#).

Backing Up Your Imported Instance

After the import operation completes, the instance runs until it is terminated. If your instance is terminated, you can't connect to or recover the instance. To ensure that you can start a new instance with the same software as an imported instance if needed, create an Amazon Machine Image (AMI) from the imported instance. For more information, see [Creating a Custom Windows AMI \(p. 65\)](#).

Troubleshooting AWS Systems Manager for Microsoft SCVMM

The following are common errors and troubleshooting steps.

Contents

- [Error: Add-in cannot be installed \(p. 913\)](#)
- [Installation Errors \(p. 913\)](#)
- [Checking the Log File \(p. 914\)](#)
- [Errors Importing a VM \(p. 914\)](#)
- [Uninstalling the Add-In \(p. 914\)](#)

Error: Add-in cannot be installed

If you receive the following error, try installing [KB2918659](#) on the computer running the VMM console. For more information, see [Description of System Center 2012 SP1 Update Rollup 5](#). Note that you don't need to install all the updates listed in this article to address this issue, just KB2918659.

```
Add-in cannot be installed
The assembly "Amazon.Scvmm.Addin" referenced to by add-in component "AWS Systems Manager
for
Microsoft SCVMM" could not be found in the add-in package. This could be due to the
following
reasons:
1. The assembly was not included with the add-in package.
2. The AssemblyName attribute for the add-in does not match the name of the add-in
assembly.
3. The assembly file is corrupt and cannot be loaded.
```

Installation Errors

If you receive one of the following errors during installation, it is likely due to an issue with SCVMM:

```
Could not update managed code add-in pipeline due to the following error:
Access to the path 'C:\Program Files\Microsoft System Center 2012\Virtual Machine Manager
\Bin\AddInPipeline\PipelineSegments.store' is denied.
```

```
Could not update managed code add-in pipeline due to the following error:
The required folder 'C:\Program Files\Microsoft System Center 2012\Virtual Machine Manager
\Bin\AddInPipeline\HostSideAdapters' does not exist.
```

Add-in cannot be installed
The assembly "Microsoft.SystemCenter.VirtualMachineManager.UIAddIns.dll" referenced by the add-in assembly "Amazon.Scvmm.AddIn" could not be found in the add-in package. Make sure that this assembly was included with the add-in package.

Try one of the following steps to work around this issue:

- Grant authenticated users permission to read and execute the C:\Program Files\Microsoft System Center 2012\Virtual Machine Manager\Bin\AddInPipeline folder. In Windows Explorer, right-click the folder, select **Properties**, and then select the **Security** tab.
- Close the SCVMM console and start it one time as an administrator. From the **Start** menu, locate SCVMM, right-click, and then select **Run as administrator**.

Checking the Log File

If you have a problem using the add-in, check the generated log file, %APPDATA%\Amazon\SCVMM\ec2addin.log, for useful information.

Errors Importing a VM

The log file, %APPDATA%\Amazon\SCVMM\ec2addin.log, contains detailed information about the status of an import task. The following are common errors that you might see in the log file when you import your VM from SCVMM to Amazon EC2.

Error: Unable to extract Hyper-V VirtualMachine object

Solution: Configure the path to the Hyper-V PowerShell module.

Error: You do not have permission to perform the operation

This error usually occurs when Hyper-V can't save the VM image into the configured path. To resolve this issue, do the following.

1. Create a directory on the Hyper-V server. For example: C:\vmimages.
2. Share the directory you just created in Hyper-V. Any user running SCVMM should be given access to the directory.
3. In the plugin, set the export path to \\hyperv\vmimages.
4. Perform the export.

The image will be exported to a local directory on the Hyper-V server. The SCVMM plugin will pull it from Hyper-V, and upload into Amazon S3.

Uninstalling the Add-In

If you need to uninstall the add-in, use the following procedure.

To uninstall the add-in

1. Open the VMM console.
2. Select the **Settings** workspace, and then click **Console Add-Ins**.
3. Select **AWS Systems Manager for Microsoft SCVMM**.
4. On the ribbon, click **Remove**.
5. When prompted for confirmation, click **Yes**.

If you reinstall the add-in after uninstalling it and receive the following error, delete the path as suggested by the error message.

```
Error (27301)
There was an error while installing the add-in. Please ensure that the following path does
not
exist and then try the installation again.

C:\Program Files\Microsoft System Center 2012\Virtual Machine Manager\Bin\AddInPipeline\
AddIns\EC2WINDOWS...
```

AWS Management Pack for Microsoft System Center

Amazon Web Services (AWS) offers a complete set of infrastructure and application services for running almost anything in the cloud—from enterprise applications and big data projects to social games and mobile apps. The AWS Management Pack for Microsoft System Center provides availability and performance monitoring capabilities for your applications running in AWS.

The AWS Management Pack allows Microsoft System Center Operations Manager to access your AWS resources (such as instances and volumes), so that it can collect performance data and monitor your AWS resources. The AWS Management Pack is an extension to System Center Operations Manager. There are two versions of the AWS Management Pack: one for System Center 2012 — Operations Manager and another for System Center Operations Manager 2007 R2.

The AWS Management Pack uses Amazon CloudWatch metrics and alarms to monitor your AWS resources. Amazon CloudWatch metrics appear in Microsoft System Center as performance counters and Amazon CloudWatch alarms appear as alerts.

You can monitor the following resources:

- EC2 instances
- EBS volumes
- ELB load balancers
- Amazon EC2 Auto Scaling groups and Availability Zones
- Elastic Beanstalk applications
- CloudFormation stacks
- CloudWatch Alarms
- CloudWatch Custom Metrics

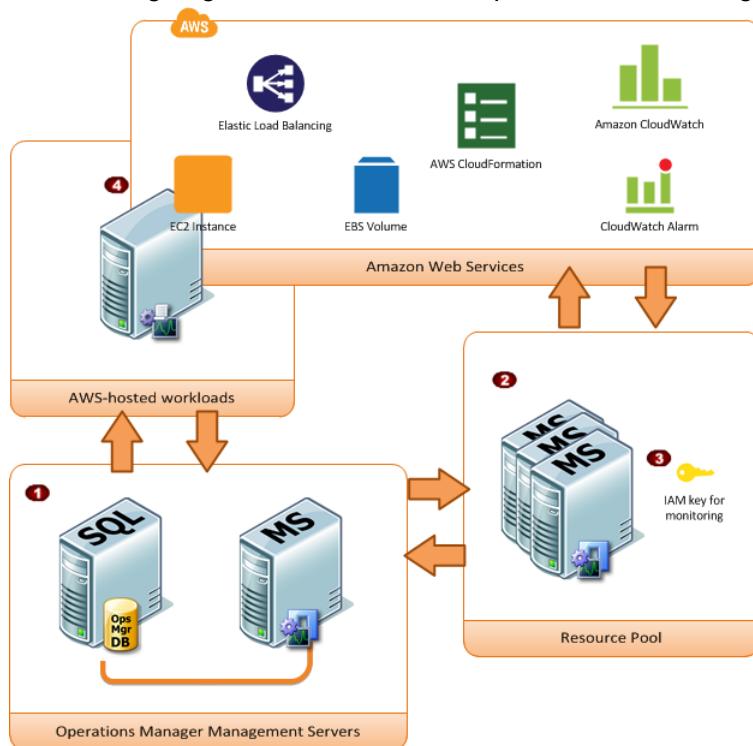
Contents

- [Overview of AWS Management Pack for System Center 2012 \(p. 916\)](#)
- [Overview of AWS Management Pack for System Center 2007 R2 \(p. 918\)](#)
- [Downloading the AWS Management Pack \(p. 919\)](#)
- [Deploying the AWS Management Pack \(p. 920\)](#)
- [Using the AWS Management Pack \(p. 929\)](#)
- [Upgrading the AWS Management Pack \(p. 949\)](#)
- [Uninstalling the AWS Management Pack \(p. 950\)](#)
- [Troubleshooting the AWS Management Pack \(p. 951\)](#)

Overview of AWS Management Pack for System Center 2012

The AWS Management Pack for System Center 2012 — Operations Manager uses a resource pool that contains one or more management servers to discover and monitor your AWS resources. You can add management servers to the pool as you increase the number of AWS resources that you use.

The following diagram shows the main components of AWS Management Pack.

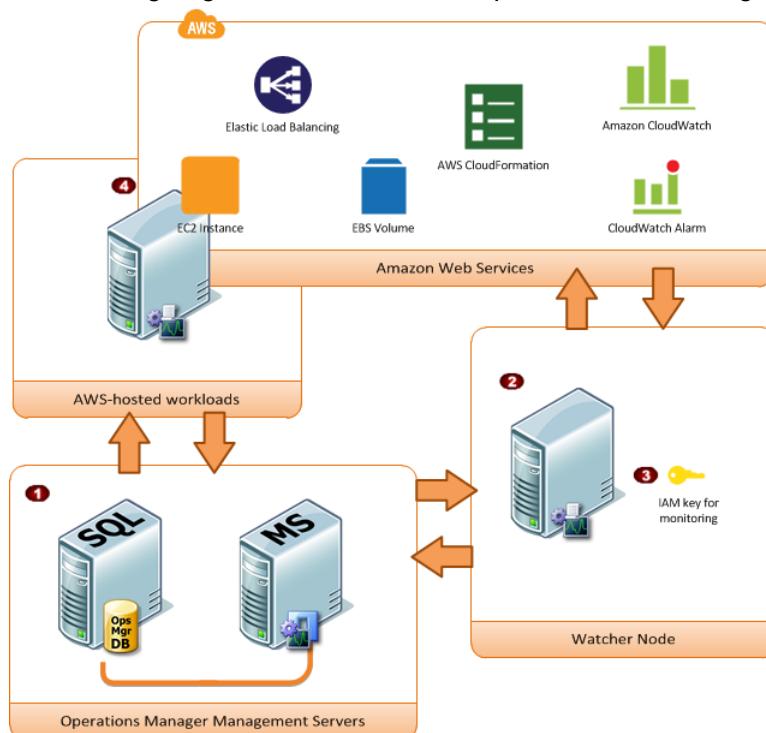


Item	Component	Description
1	Operations Manager infrastructure	One or more management servers and their dependencies, such as Microsoft SQL Server and a Microsoft Active Directory domain. These servers can either be deployed on-premises or in the AWS cloud; both scenarios are supported.
2	Resource pool	One or more management servers used for communicating with AWS using the AWS SDK for .NET. These servers must have Internet connectivity.
3	AWS credentials	An access key ID and a secret access key used by the management servers to make AWS API calls. You must specify these credentials while you configure the AWS Management Pack. We recommend that you create an IAM user with read-only privileges and use those credentials. For more information about creating an IAM user, see Adding a New User to Your AWS Account in the <i>IAM User Guide</i> .
4	EC2 instances	Virtual computers running in the AWS cloud. Some instances might have the Operations Manager Agent installed, others might not. When you install Operations Manager Agent you can see the operating system and application health apart from the instance health.

Overview of AWS Management Pack for System Center 2007 R2

The AWS Management Pack for System Center Operations Manager 2007 R2 uses a designated computer that connects to your System Center environment and has Internet access, called a *watcher node*, to call AWS APIs to remotely discover and collect information about your AWS resources.

The following diagram shows the main components of AWS Management Pack.



Item	Component	Description
①	Operations Manager infrastructure	One or more management servers and their dependencies, such as Microsoft SQL Server and a Microsoft Active Directory domain. These servers can either be deployed on-premises or in the AWS cloud; both scenarios are supported.
②	Watcher node	A designated agent-managed computer used for communicating with AWS using the AWS SDK for .NET. It can either be deployed on-premises or in the AWS cloud, but it must be an agent-managed computer, and it must have Internet connectivity. You can use exactly one watcher node to monitor an AWS account. However, one watcher node can monitor multiple AWS accounts. For more information about setting up a watcher node, see Deploying Windows Agents in the Microsoft System Center documentation.
③	AWS credentials	An access key ID and a secret access key used by the watcher node to make AWS API calls. You must specify these credentials while you configure the AWS Management Pack. We recommend that you create an IAM user with read-only

Item	Component	Description
		privileges and use those credentials. For more information about creating an IAM user, see Adding a New User to Your AWS Account in the <i>IAM User Guide</i> .
④	EC2 instances	Virtual computers running in the AWS cloud. Some instances might have the Operations Manager Agent installed, others might not. When you install the Operations Manager Agent you can see the operating system and application health apart from the instance health.

Downloading the AWS Management Pack

To get started, download the AWS Management Pack. The AWS Management Pack is free. You might incur charges for Amazon CloudWatch, depending on how you configure monitoring or how many AWS resources you monitor.

System Center 2012

Before you download the AWS Management Pack, ensure that your systems meet the following system requirements and prerequisites.

System Requirements

- System Center Operations Manager 2012 R2 or System Center Operations Manager 2012 SP1
- Cumulative Update 1 or later. You must deploy the update to the management servers monitoring AWS resources, as well as agents running the watcher nodes and agents to be monitored by the AWS Management Pack. We recommend that you deploy the latest available Operations Manager updates on all computers monitoring AWS resources.
- Microsoft.Unix.Library MP version 7.3.2026.0 or later

Prerequisites

- Your data center must have at least one management server configured with Internet connectivity. The management servers must have the Microsoft .NET Framework version 4.5 or later and PowerShell 2.0 or later installed.
- The action account for the management server must have local administrator privileges on the management server.

To download the AWS Management Pack

1. On the [AWS Add-Ins for Microsoft System Center](#) website, click **SCOM 2012**.
2. Save **AWS-SCOM-MP-2.5.zip** to your computer and unzip it.

Continue with [Deploying the AWS Management Pack \(p. 920\)](#).

System Center 2007 R2

Before you download the AWS Management Pack, ensure that your systems meet the following system requirements and prerequisites.

System Requirements

- System Center Operations Manager 2007 R2
- Microsoft.Unix.Library MP version 6.1.7000.256 or later

Prerequisites

- Your data center must have an agent-managed computer with Internet connectivity that you designate as the watcher node. The watcher node must have the following Agent Proxy option enabled: **Allow this agent to act as a proxy and discover managed objects on other computers**. The watcher node must have the Microsoft .NET Framework version 3.5.1 or later and PowerShell 2.0 or later installed.
- The action account for the watcher node must have local administrator privileges on the watcher node.
- You must ensure that your watcher node has the agent installed, has Internet access, and can communicate with the management servers in your data center. For more information, see [Deploying Windows Agents](#) in the Microsoft System Center documentation.

To download the AWS Management Pack

1. On the [AWS Add-Ins for Microsoft System Center](#) website, click **SCOM 2007**.
2. Save **AWS-MP-Setup-2.5.msi** to your computer.

Continue with [Deploying the AWS Management Pack \(p. 920\)](#).

Deploying the AWS Management Pack

Before you can deploy the AWS Management Pack, you must download it. For more information, see [Downloading the AWS Management Pack \(p. 919\)](#).

Tasks

- [Step 1: Installing the AWS Management Pack \(p. 920\)](#)
- [Step 2: Configuring the Watcher Node \(p. 922\)](#)
- [Step 3: Create an AWS Run As Account \(p. 922\)](#)
- [Step 4: Run the Add Monitoring Wizard \(p. 925\)](#)
- [Step 5: Configure Ports and Endpoints \(p. 929\)](#)

Step 1: Installing the AWS Management Pack

After you download the AWS Management Pack, you must configure it to monitor one or more AWS accounts.

System Center 2012

To install the AWS Management Pack

1. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.
2. In the **Actions** pane, click **Import Management Packs**.

3. On the **Select Management Packs** page, click **Add**, and then click **Add from disk**.
4. In the **Select Management Packs to import** dialog box, select the `Amazon.AmazonWebServices.mpb` file from the location where you downloaded it, and then click **Open**.
5. On the **Select Management Packs** page, under **Import list**, select the **Amazon Web Services** management pack, and then click **Install**.

Note
System Center Operations Manager doesn't import any management packs in the **Import** list that display an **Error** icon.
6. The **Import Management Packs** page shows the progress for the import process. If a problem occurs, select the management pack in the list to view the status details. Click **Close**.

System Center 2007 R2

To install the AWS Management Pack

The management pack is distributed as a Microsoft System Installer file, `AWS-MP-Setup.msi`. It contains the required DLLs for the watcher node, root management server, and Operations console, as well as the `Amazon.AmazonWebServices.mp` file.

1. Run `AWS-MP-Setup.msi`.

Note

If your root management server, Operations console, and watcher node are on different computers, you must run the installer on each computer.

2. On the **Welcome to the Amazon Web Services Management Pack Setup Wizard** screen, click **Next**.
3. On the **End-User License Agreement** screen, read the license agreement, and, if you accept the terms, select the **I accept the terms in the License Agreement** check box, and then click **Next**.
4. On the **Custom Setup** screen, select the features you want to install, and then click **Next**.

Operations Console

Installs `Amazon.AmazonWebServices.UI.Pages.dll` and registers it in the Global Assembly Cache (GAC), and then installs `Amazon.AmazonWebServices.mp`.

Root Management Server

Installs `Amazon.AmazonWebServices.Modules.dll`, `Amazon.AmazonWebServices.SCOM.SDK.dll` and the AWS SDK for .NET (`AWSSDK.dll`), and then registers them in the GAC.

AWS Watcher Node

Installs `Amazon.AmazonWebServices.Modules.dll` and `Amazon.AmazonWebServices.SCOM.SDK.dll`, and then installs the AWS SDK for .NET (`AWSSDK.dll`) and registers it in the GAC.

5. On the **Ready to install Amazon Web Services Management Pack** screen, click **Install**.
6. On the **Completed the Amazon Web Services Management Pack Setup Wizard** screen, click **Finish**.

Note

The required DLLs are copied and registered in the GAC, and the management pack file (*.mp) is copied to the `Program Files (x86)/Amazon Web Services Management Pack` folder on the computer running the Operations console. Next, you must import the management pack into System Center.

7. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.

8. In the **Actions** pane, click **Import Management Packs**.
9. On the **Select Management Packs** page, click **Add**, and then click **Add from disk**.
10. In the **Select Management Packs to import** dialog box, change the directory to C:\\Program Files (x86)\\Amazon Web Services Management Pack, select the Amazon.AmazonWebServices.mp file, and then click **Open**.
11. On the **Select Management Packs** page, under **Import list**, select the **Amazon Web Services** management pack, and then click **Install**.

Note

System Center Operations Manager doesn't import any management packs in the **Import** list that display an **Error** icon.

12. The **Import Management Packs** page shows the progress for the import process. If a problem occurs, select the management pack in the list to view the status details. Click **Close**.

Step 2: Configuring the Watcher Node

On System Center Operations Manager 2007 R2, the watcher node runs discoveries that go beyond the watcher node computer, so you must enable the proxy agent option on the watcher node. The proxy agent allows those discoveries to access the objects on other computers.

Note

If your system is configured with a large number of resources, we recommend that you configure one management server as a Watcher Node. Having a separate Watcher Node management server can improve performance.

If you're using System Center 2012 — Operations Manager, you can skip this step.

To enable the proxy agent on System Center Operations Manager 2007 R2

1. In the Operations console, on the **Go** menu, click **Administration**.
2. In the **Administration** workspace, under **Device Management**, click **Agent Managed**.
3. In the **Agent Managed** list, right-click the watcher node, and then click **Properties**.
4. In the **Agent Properties** dialog box, click the **Security** tab, select **Allow this agent to act as proxy and discover managed objects on other computers**, and then click **OK**.

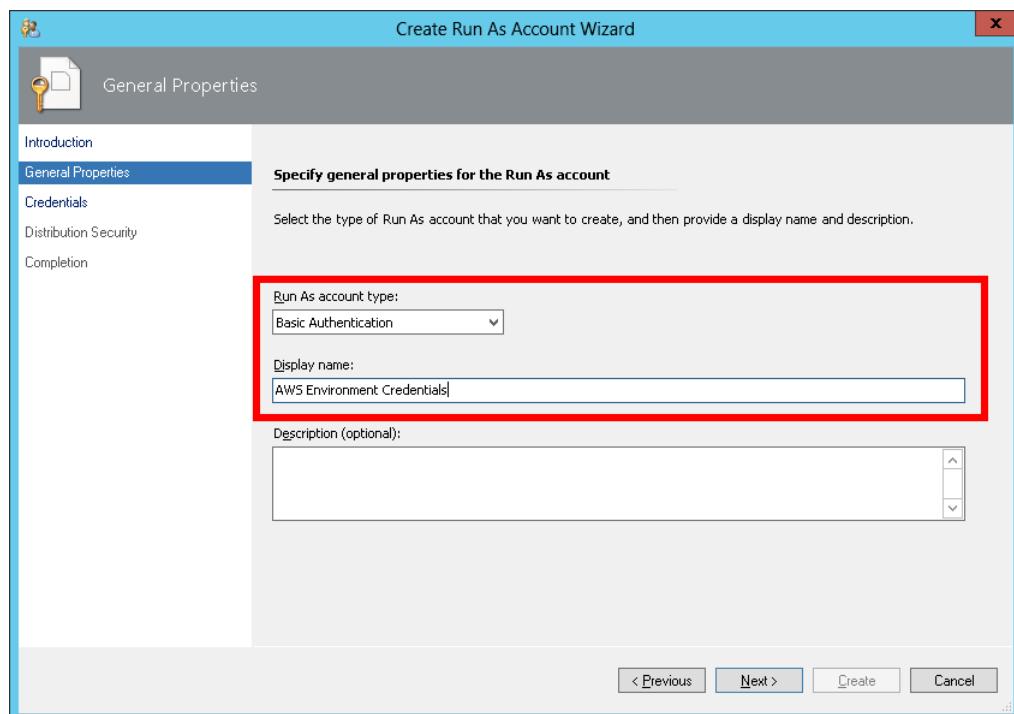
Step 3: Create an AWS Run As Account

You must set up credentials that grant AWS Management Pack access to your AWS resources.

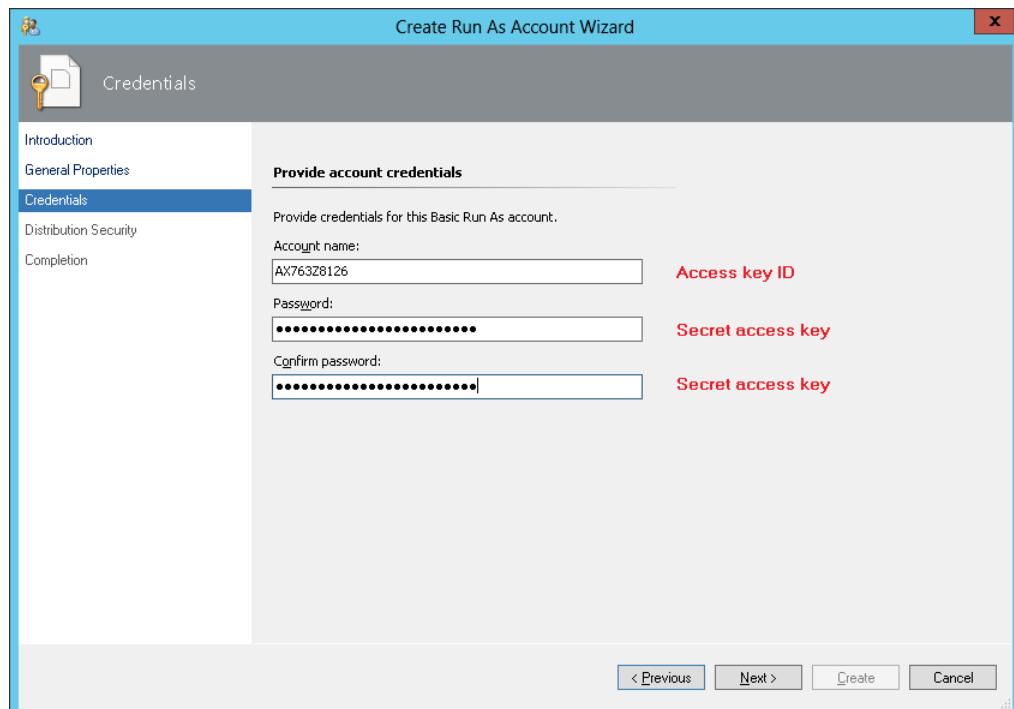
To create an AWS Run As account

1. We recommend that you create an IAM user with the minimum access rights required (for example, the **ReadOnlyAccess** AWS managed policy works in most cases). You'll need the access keys (access key ID and secret access key) for this user to complete this procedure. For more information, see [Administering Access Keys for IAM Users](#) in the *IAM User Guide*.
2. In the Operations console, on the **Go** menu, click **Administration**.
3. In the **Administration** workspace, expand the **Run As Configuration** node, and then select **Accounts**.
4. Right-click the **Accounts** pane, and then click **Create Run As Account**.
5. In the **Create Run As Account Wizard**, on the **General Properties** page, in the **Run As account type** list, select **Basic Authentication**.
6. Enter a display name (for example, "My IAM Account") and a description, and then click **Next**.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Step 3: Create an AWS Run As Account

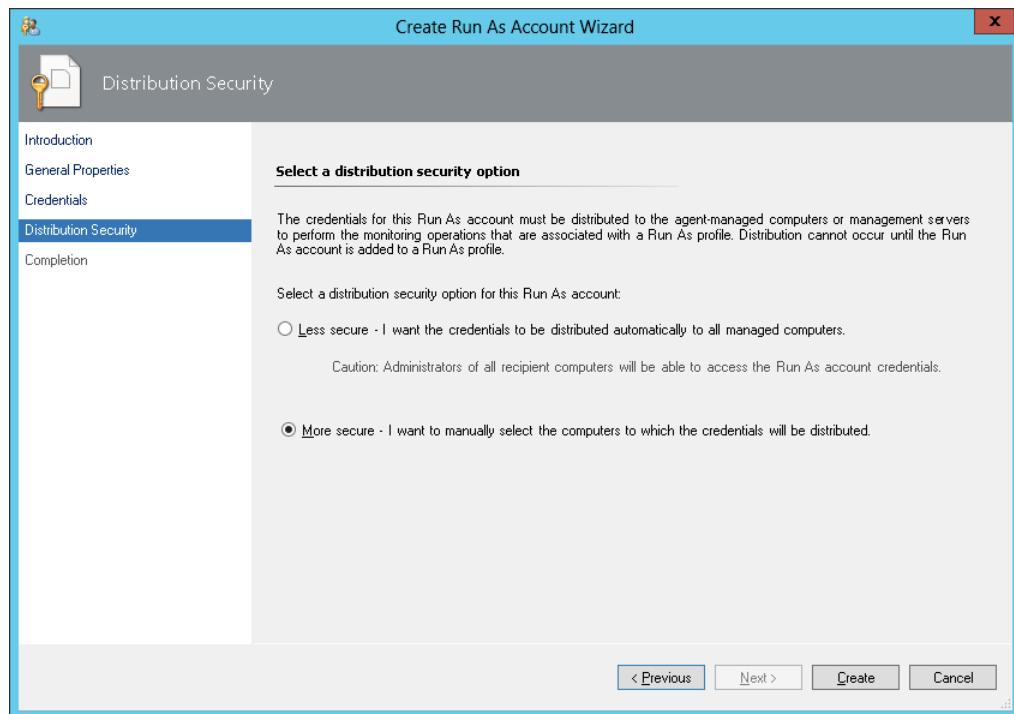


7. On the **Credentials** page, enter the access key ID in the **Account name** box and the secret access key in the **Password** box, and then click **Next**.

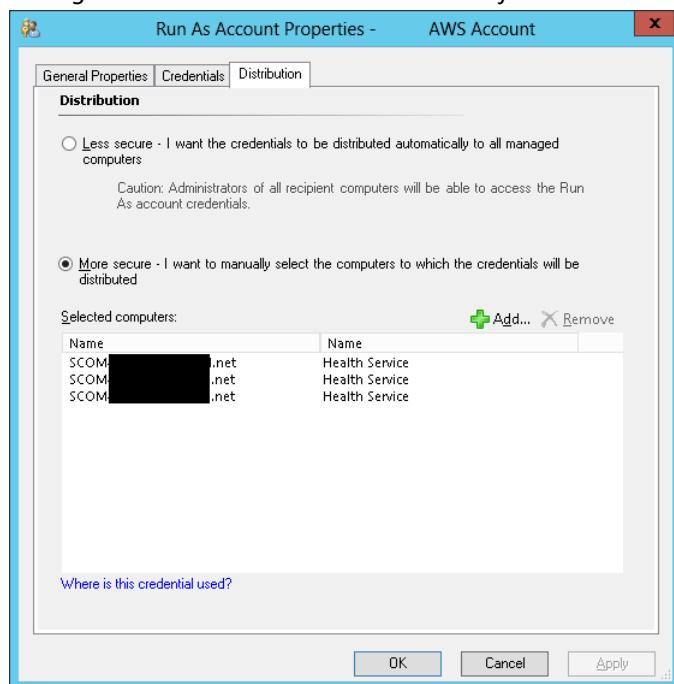


8. On the **Distribution Security** page, select **More secure - I want to manually select the computers to which the credentials will be distributed**, and then click **Create**.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Step 3: Create an AWS Run As Account



9. Click **Close**.
10. In the list of accounts, select the account that you just created.
11. In the **Actions** pane, click **Properties**.
12. In the **Properties** dialog box, verify that the **More Secure** option is selected and that all management servers to be used to monitor your AWS resources are listed.



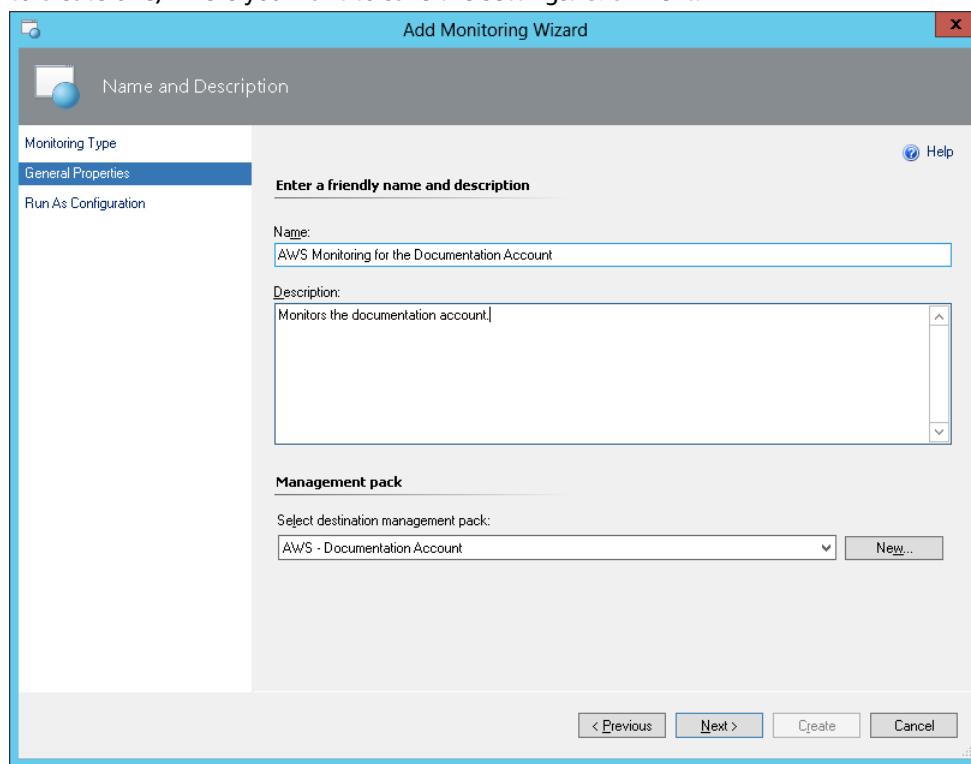
Step 4: Run the Add Monitoring Wizard

You can configure the AWS Management Pack to monitor a particular AWS account by using the Add Monitoring Wizard, which is available in the **Authoring** workspace of the Operations console. This wizard creates a management pack that contains the settings for the AWS account to monitor. You must run this wizard to monitor each AWS account. For example, if you want to monitor two AWS accounts, you must run the wizard twice.

System Center 2012

To run the Add Monitoring Wizard on System Center 2012 — Operations Manager

1. In the Operations console, on the **Go** menu, click **Authoring**.
2. In the **Authoring** workspace, expand the **Management Pack Templates** node, right-click **Amazon Web Services**, and then click **Add Monitoring Wizard**.
3. In the **Add Monitoring Wizard**, in the **Select the monitoring type** list, select **Amazon Web Services**, and then click **Next**.
4. On the **General Properties** page, in the **Name** box, enter a name (for example, "My AWS Resources"). In the **Description** box, enter a description.
5. In the **Select destination management pack** list, select an existing management pack (or click **New** to create one) where you want to save the settings. Click **Next**.

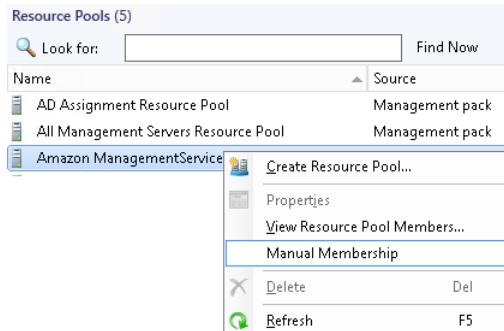


By default, when you create a management pack object, disable a rule or monitor, or create an override, Operations Manager saves the setting to the default management pack. As a best practice, you should create a separate management pack for each sealed management pack that you want to customize, instead of saving your customized settings to the default management pack.

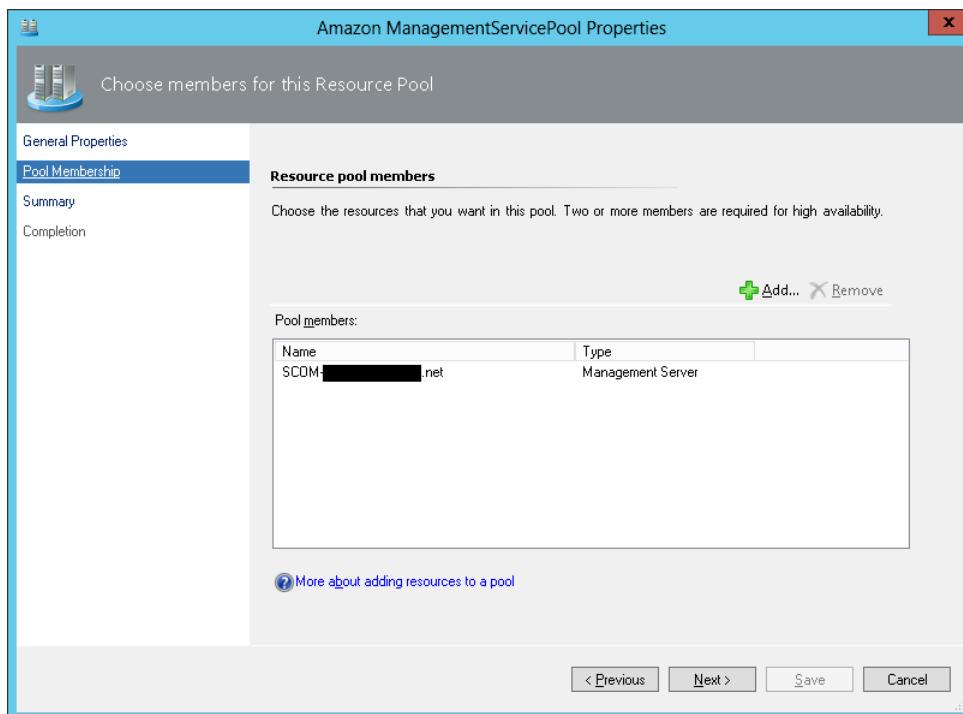
6. The AWS Management Pack automatically creates a resource pool and adds the management servers to it. To control server membership, make the following changes:

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Step 4: Run the Add Monitoring Wizard

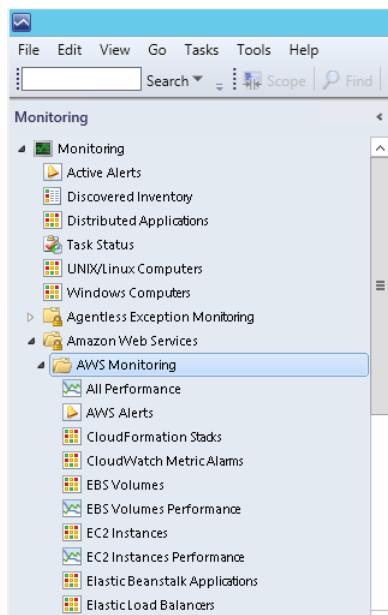
- a. Click **Administration** on the **Go** menu.
- b. Click the **Resource Pools** node.
- c. Right-click the **AWS Resource Pool** in the **Resource Pools** pane and select **Manual Membership**.



- d. Right-click the **AWS Resource Pool** in the **Resource Pools** pane and select **Properties**.
- e. On the **Pool Membership** page, remove the management servers that should not monitor AWS resources.



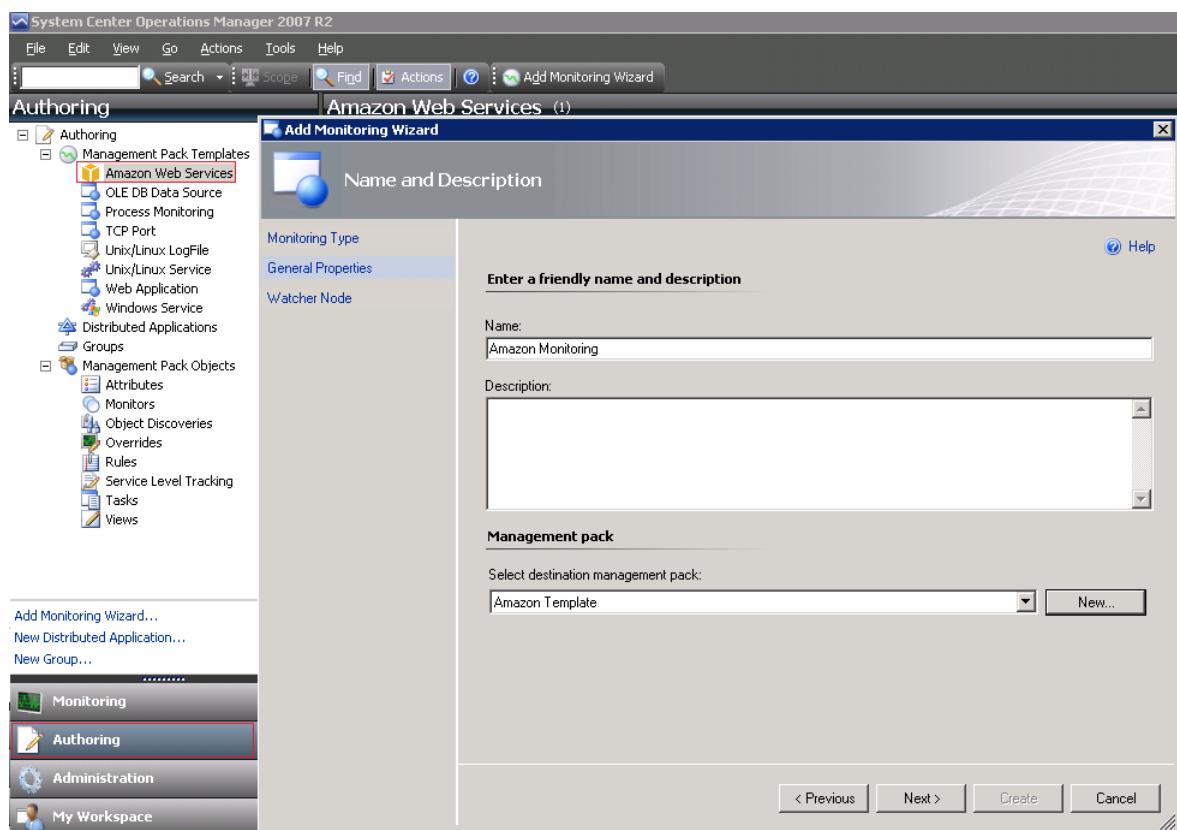
7. After the AWS Management Pack is configured, it shows up as a sub-folder of the **Amazon Web Services** folder in the **Monitoring** workspace of the Operations console.



System Center 2007 R2

To run the Add Monitoring Wizard on System Center Operations Manager 2007

1. In the Operations console, on the **Go** menu, click **Authoring**.
2. In the **Authoring** workspace, expand the **Management Pack Templates** node, right-click **Amazon Web Services**, and then click **Add Monitoring Wizard**.
3. In the **Add Monitoring Wizard**, in the **Select the monitoring type list**, select **Amazon Web Services**, and then click **Next**.
4. On the **General Properties** page, in the **Name** box, enter a name (for example, "My AWS Resources"). In the **Description** box, enter a description.
5. In the **Select destination management pack** drop-down list, select an existing management pack (or click **New** to create a new one) where you want to save the settings. Click **Next**.

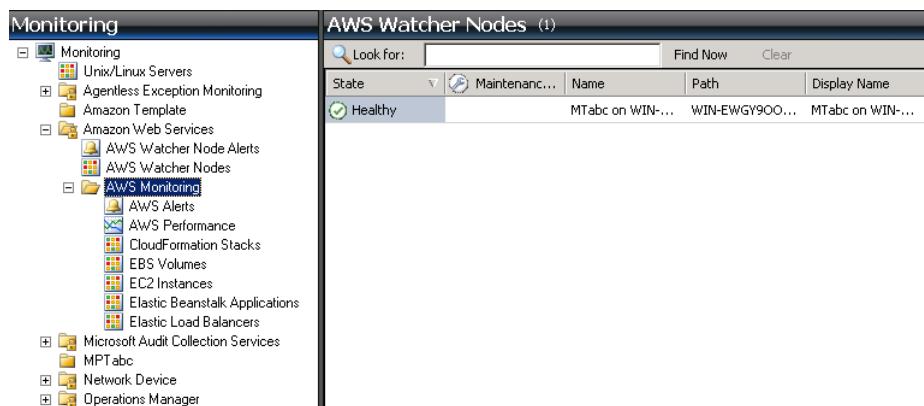


By default, when you create a management pack object, disable a rule or monitor, or create an override, Operations Manager saves the setting to the default management pack. As a best practice, you should create a separate management pack for each sealed management pack that you want to customize, instead of saving your customized settings to the default management pack.

6. On the **Watcher Node Configuration** page, in the **Watcher Node** list, select an agent-managed computer to act as the watcher node.
7. In the **Select AWS Run As account** drop-down list, select the Run As account that you created earlier, and then click **Create**.
8. After the AWS Management Pack is configured, it first discovers the watcher node. To verify that the watcher node was discovered successfully, navigate to the **Monitoring** workspace in the Operations console. You should see a new **Amazon Web Services** folder and an **Amazon Watcher Nodes** subfolder under it. This subfolder displays the watcher nodes. The AWS Management Pack automatically checks and monitors the watcher node connectivity to AWS. When the watcher node is discovered, it shows up in this list. When the watcher node is ready, its state changes to **Healthy**.

Note

To establish connectivity with AWS, the AWS Management Pack requires that you deploy the AWS SDK for .NET, modules, and scripts to the watcher node. This can take about ten minutes. If the watcher node doesn't appear, or if you see the state as **Not Monitored**, verify your Internet connectivity and IAM permissions. For more information, see [Troubleshooting the AWS Management Pack \(p. 951\)](#).



- After the watcher node is discovered, dependent discoveries are triggered, and the AWS resources are added to the **Monitoring** workspace of the Operations console.

The discovery of AWS resources should finish within twenty minutes. This process can take more time, based on your Operations Manager environment, your AWS environment, the load on the management server, and the load on the watcher node. For more information, see [Troubleshooting the AWS Management Pack \(p. 951\)](#).

Step 5: Configure Ports and Endpoints

The AWS Management Pack for Microsoft System Center must be able to communicate with AWS services to monitor the performance of those services and provide alerts in System Center. For monitoring to succeed, you must configure the firewall on the Management Pack servers to allow outbound HTTP calls on ports 80 and 443 to the AWS endpoints for the following services.

This enables monitoring for the following AWS services:

- Amazon Elastic Compute Cloud (EC2)
- Elastic Load Balancing
- Amazon EC2 Auto Scaling
- AWS Elastic Beanstalk
- Amazon CloudWatch
- AWS CloudFormation

The AWS Management Pack uses the public APIs in the AWS SDK for .NET to retrieve information from these services over ports 80 and 443. Log on to each server and enable outbound firewall rules for ports 80 and 443.

If your firewall application supports more detailed settings you can configure specific endpoints for each service. An endpoint is a URL that is the entry point for a web service. For example, ec2.us-west-2.amazonaws.com is an entry point for the Amazon EC2 service. To configure endpoints on your firewall, [locate the specific endpoint URLs](#) for the AWS services you are running and specify those endpoints in your firewall application.

Using the AWS Management Pack

You can use the AWS Management Pack to monitor the health of your AWS resources.

Contents

- [Views \(p. 930\)](#)
- [Discoveries \(p. 944\)](#)
- [Monitors \(p. 945\)](#)
- [Rules \(p. 946\)](#)
- [Events \(p. 946\)](#)
- [Health Model \(p. 947\)](#)
- [Customizing the AWS Management Pack \(p. 948\)](#)

Views

The AWS Management Pack provides the following views, which are displayed in the **Monitoring** workspace of the Operations console.

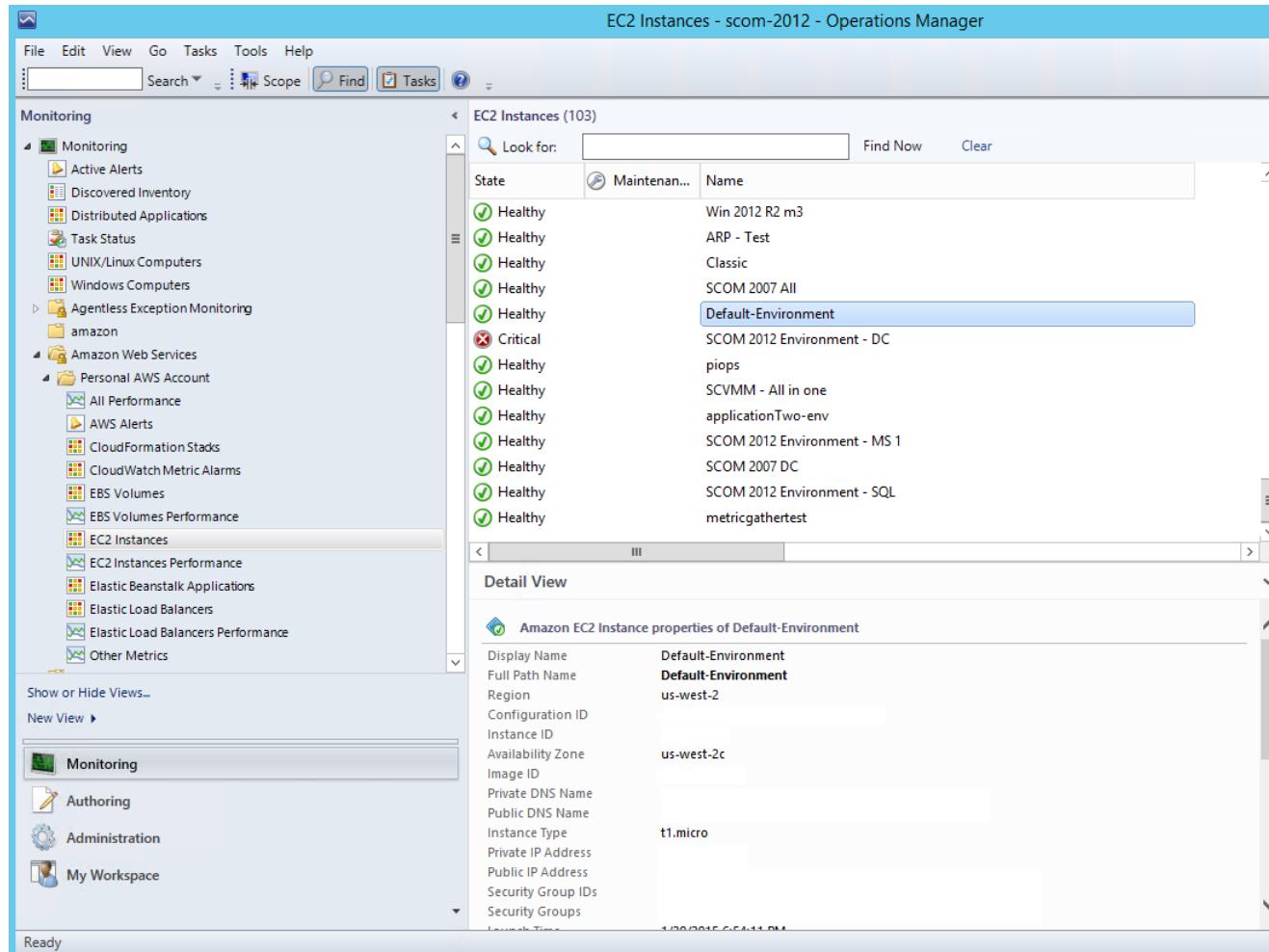
Views

- [EC2 Instances \(p. 930\)](#)
- [Amazon EBS Volumes \(p. 932\)](#)
- [Elastic Load Balancers \(p. 934\)](#)
- [AWS Elastic Beanstalk Applications \(p. 936\)](#)
- [AWS CloudFormation Stacks \(p. 938\)](#)
- [Amazon Performance Views \(p. 940\)](#)
- [Amazon CloudWatch Metric Alarms \(p. 941\)](#)
- [AWS Alerts \(p. 942\)](#)
- [Watcher Nodes \(System Center Operations Manager 2007 R2\) \(p. 943\)](#)

EC2 Instances

View the health state of the EC2 instances for a particular AWS account, from all Availability Zones and regions. The view also includes EC2 instances running in a virtual private cloud (VPC). The AWS Management Pack retrieves tags, so you can search and filter the list using those tags.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Views



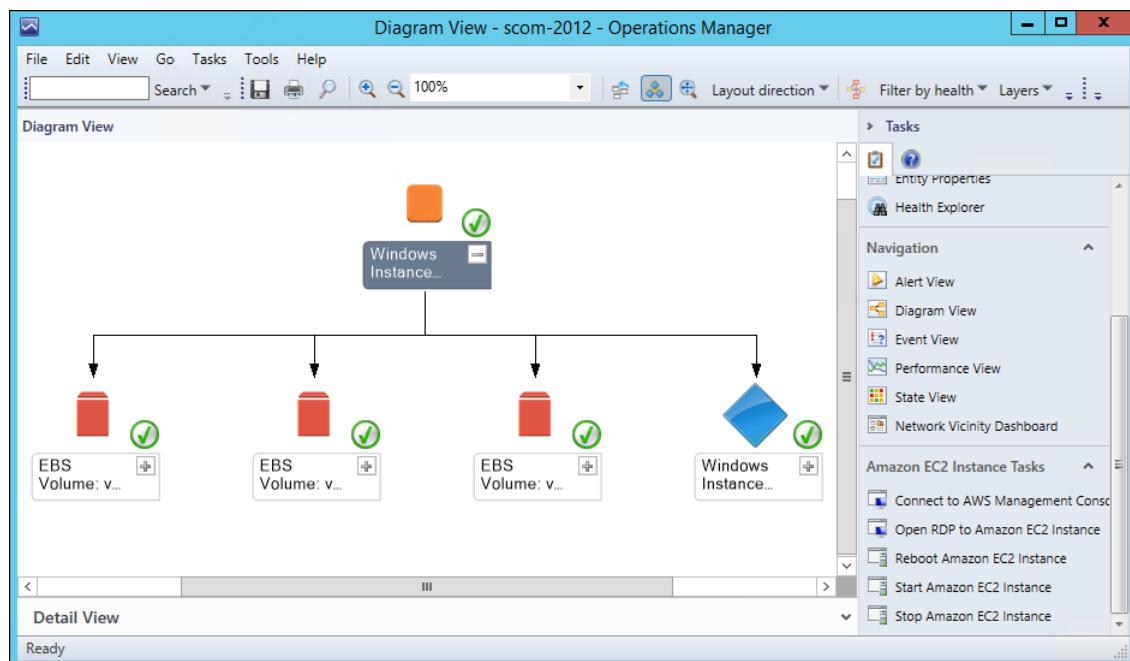
When you select an EC2 instance, you can perform instance health tasks:

- **Open Amazon Console:** Launches the AWS Management Console in a web browser.
- **Open RDP to Amazon EC2 Instance:** Opens an RDP connection to the selected Windows instance.
- **Reboot Amazon EC2 Instance:** Reboots the selected EC2 instance.
- **Start Amazon EC2 Instance:** Starts the selected EC2 instance.
- **Stop Amazon EC2 Instance:** Stops the selected EC2 instance.

EC2 Instances Diagram View

Shows the relationship of an instance with other components.

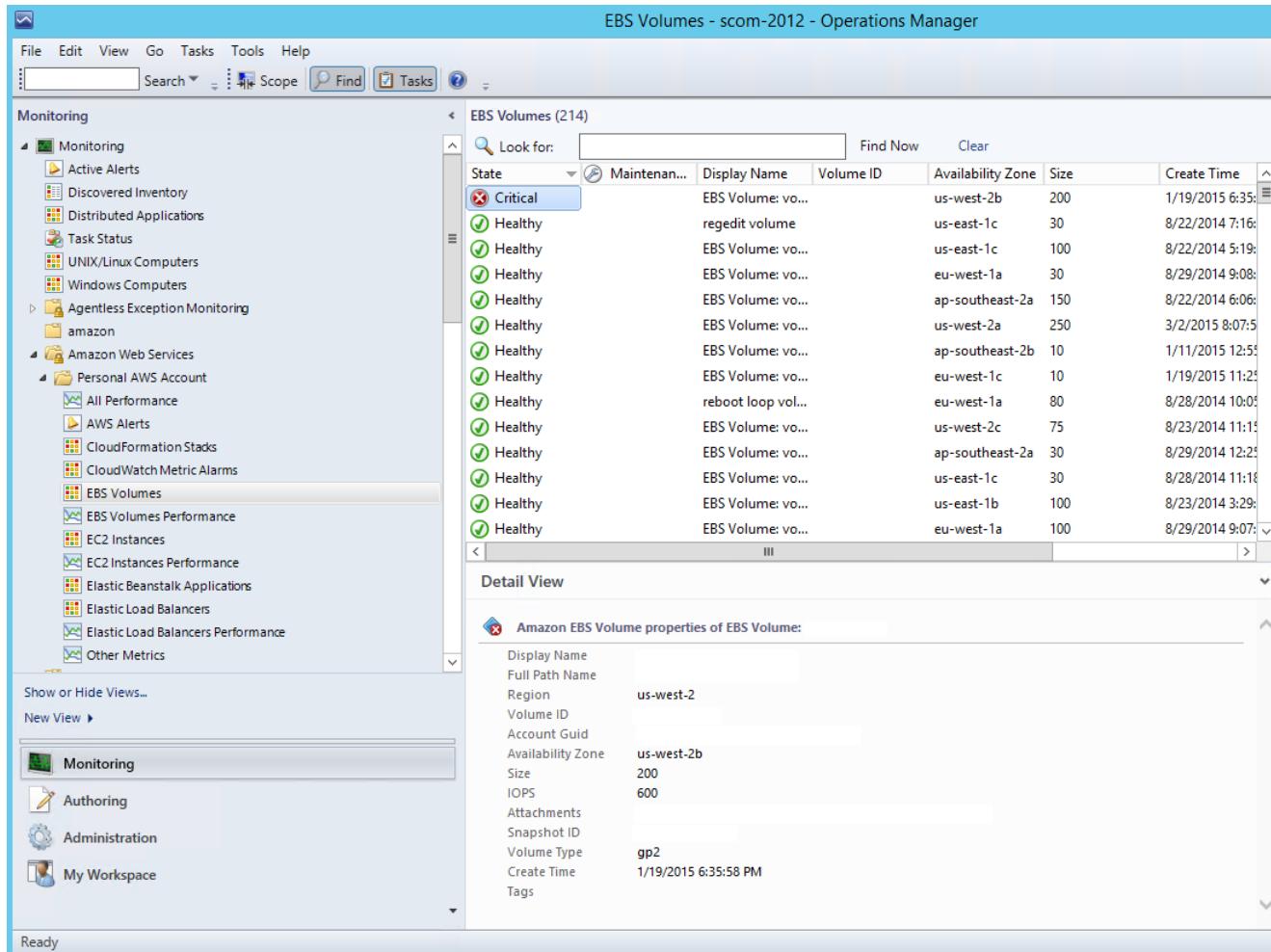
Amazon Elastic Compute Cloud
User Guide for Windows Instances
Views



Amazon EBS Volumes

Shows the health state of all the Amazon EBS volumes for a particular AWS account from all Availability Zones and regions.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Views

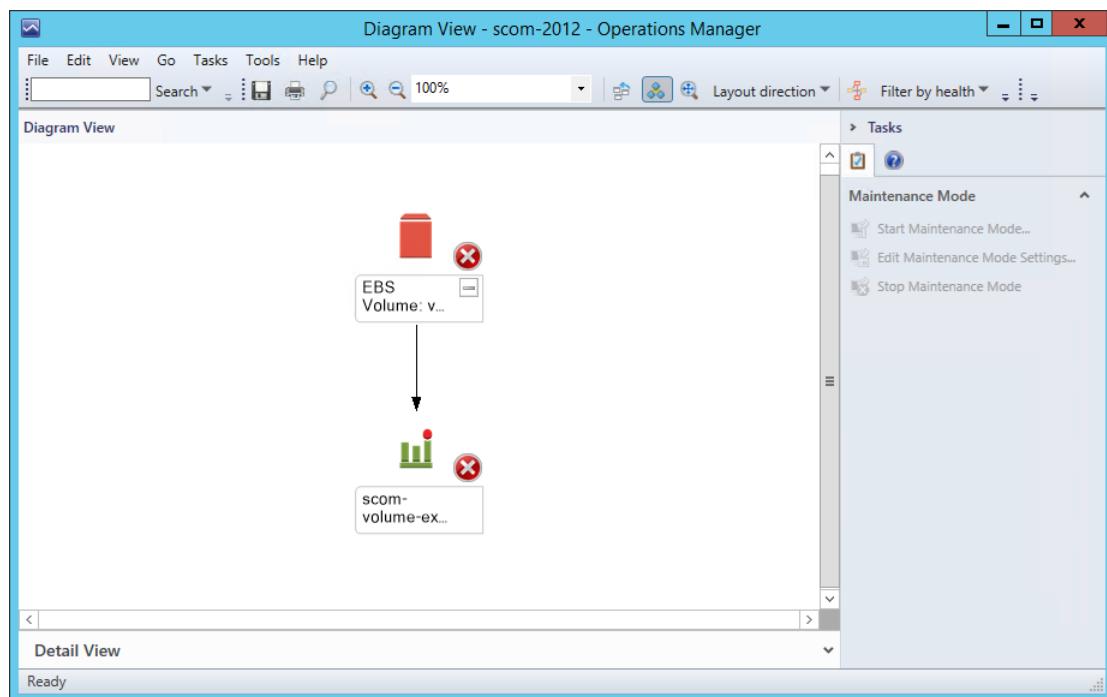


The screenshot shows the 'EBS Volumes - scom-2012 - Operations Manager' window. The left pane contains a navigation tree with categories like Monitoring, Active Alerts, Discovered Inventory, Distributed Applications, Task Status, UNIX/Linux Computers, Windows Computers, Agentless Exception Monitoring, Amazon Web Services, Personal AWS Account, and EBS Volumes. The right pane displays a table titled 'EBS Volumes (214)' with columns: State, Maintenance, Display Name, Volume ID, Availability Zone, Size, and Create Time. A search bar at the top of the table allows filtering by 'Look for'. Below the table is a 'Detail View' section for a selected 'Critical' volume, showing its properties: Display Name (EBS Volume: vo...), Region (us-west-2), Volume ID, Account Guid, Availability Zone (us-west-2b), Size (200), IOPS (600), Attachments, Snapshot ID, Volume Type (gp2), Create Time (1/19/2015 6:35:58 PM), and Tags.

State	Maintain...	Display Name	Volume ID	Availability Zone	Size	Create Time
X Critical		EBS Volume: vo...		us-west-2b	200	1/19/2015 6:35:58 PM
✓ Healthy		regedit volume		us-east-1c	30	8/22/2014 7:16:40 AM
✓ Healthy		EBS Volume: vo...		us-east-1c	100	8/22/2014 5:16:40 AM
✓ Healthy		EBS Volume: vo...		eu-west-1a	30	8/29/2014 9:08:40 AM
✓ Healthy		EBS Volume: vo...		ap-southeast-2a	150	8/22/2014 6:06:40 AM
✓ Healthy		EBS Volume: vo...		us-west-2a	250	3/2/2015 8:07:55 AM
✓ Healthy		EBS Volume: vo...		ap-southeast-2b	10	1/11/2015 12:55:40 AM
✓ Healthy		EBS Volume: vo...		eu-west-1c	10	1/19/2015 11:25:40 AM
✓ Healthy		reboot loop vol...		eu-west-1a	80	8/28/2014 10:05:40 AM
✓ Healthy		EBS Volume: vo...		us-west-2c	75	8/23/2014 11:15:40 AM
✓ Healthy		EBS Volume: vo...		ap-southeast-2a	30	8/29/2014 12:25:40 AM
✓ Healthy		EBS Volume: vo...		us-east-1c	30	8/28/2014 11:18:40 AM
✓ Healthy		EBS Volume: vo...		us-east-1b	100	8/23/2014 3:29:40 AM
✓ Healthy		EBS Volume: vo...		eu-west-1a	100	8/29/2014 9:07:40 AM

Amazon EBS Volumes Diagram View

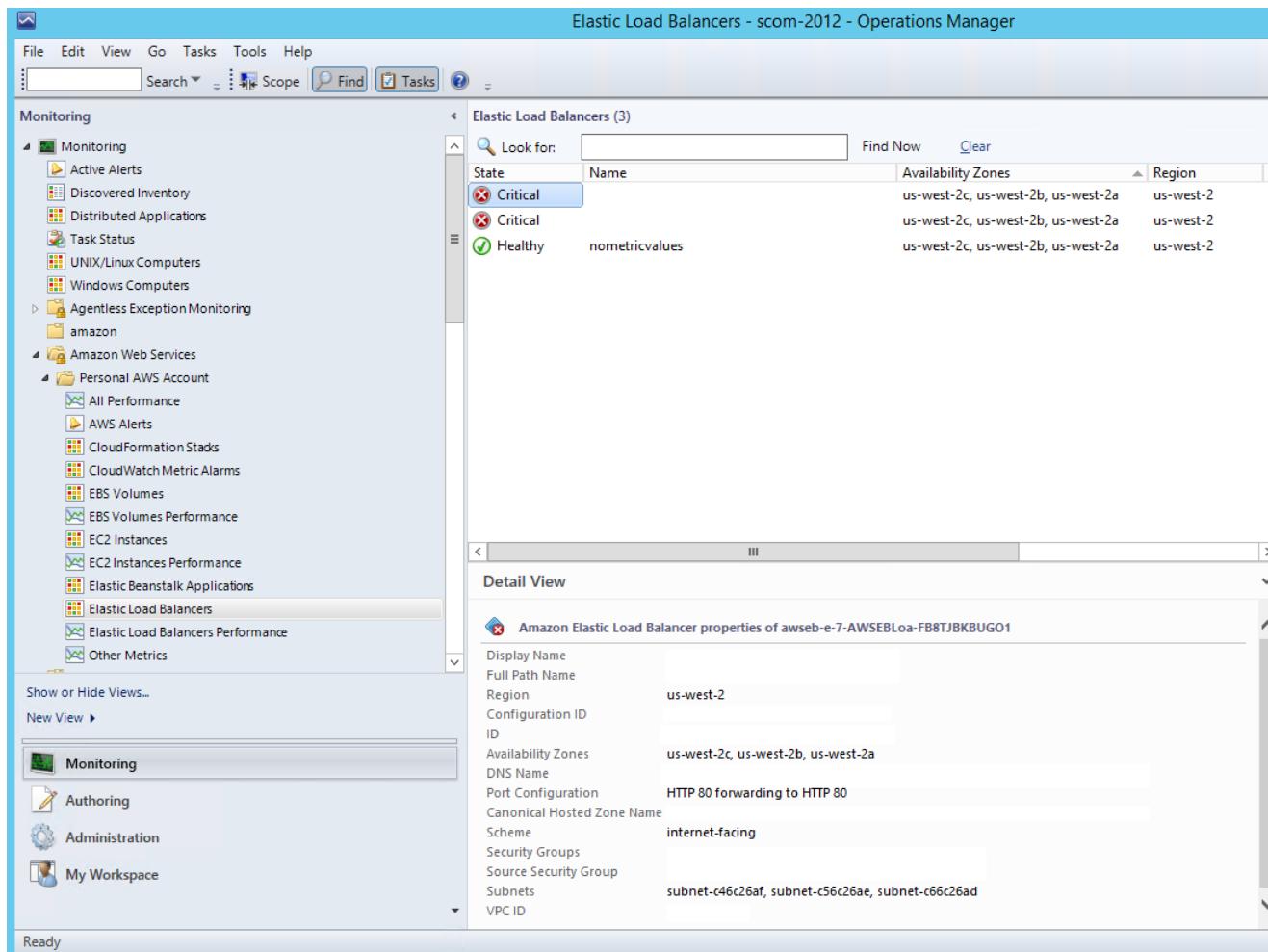
Shows an Amazon EBS volume and any associated alarms. The following illustration shows an example:



Elastic Load Balancers

Shows the health state of all the load balancers for a particular AWS account from all regions.

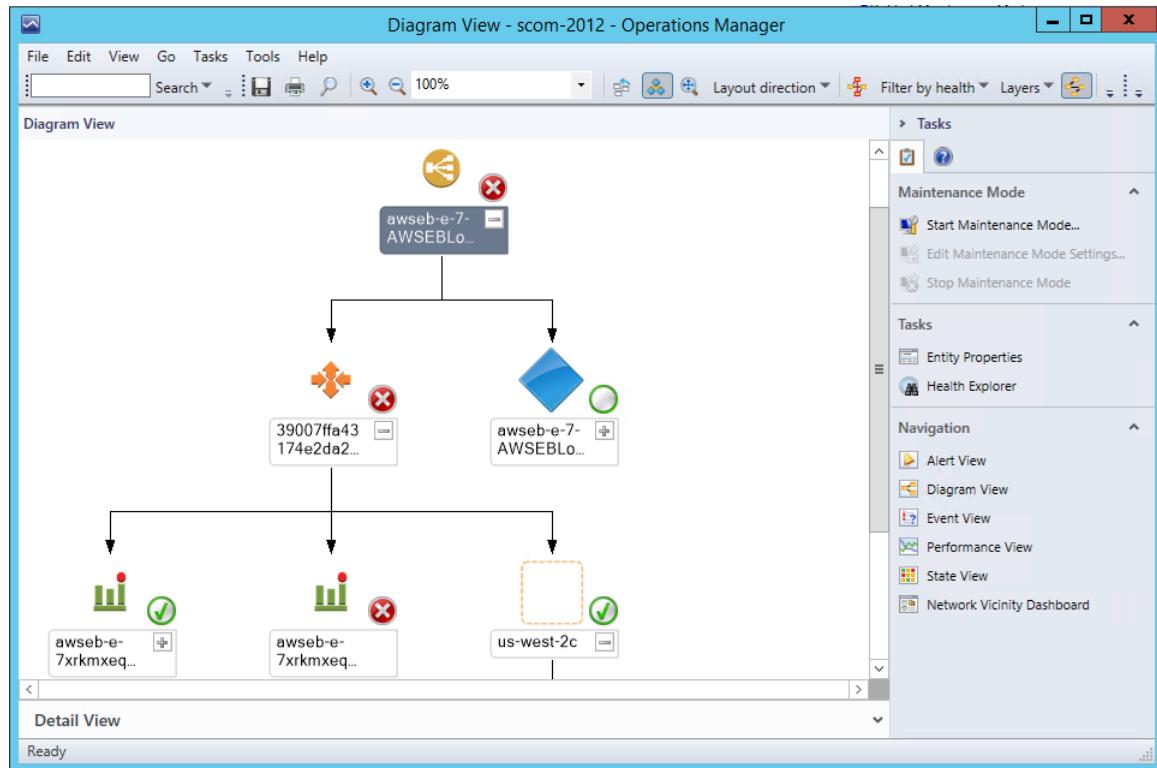
**Amazon Elastic Compute Cloud
User Guide for Windows Instances
Views**



Elastic Load Balancing Diagram View

Shows the Elastic Load Balancing relationship with other components. The following illustration shows an example:

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Views



AWS Elastic Beanstalk Applications

Shows the state of all discovered AWS Elastic Beanstalk applications.

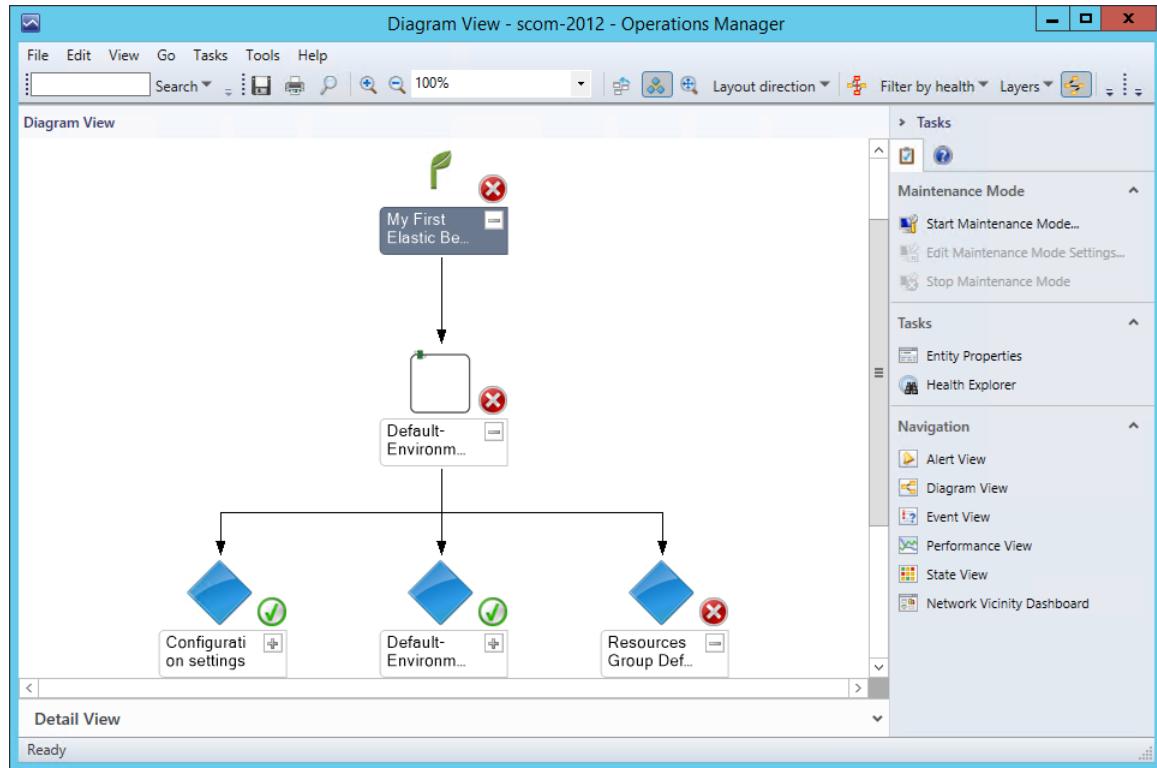
Amazon Elastic Compute Cloud
User Guide for Windows Instances
Views

The screenshot shows the 'Elastic Beanstalk Applications - scom-2012 - Operations Manager' window. The left sidebar contains a tree view with categories like Monitoring, Active Alerts, Discovered Inventory, Distributed Applications, Task Status, UNIX/Linux Computers, Windows Computers, Agentless Exception Monitoring, amazon, and Amazon Web Services. Under Amazon Web Services, Personal AWS Account is expanded, showing All Performance, AWS Alerts, CloudFormation Stacks, CloudWatch Metric Alarms, EBS Volumes, EBS Volumes Performance, EC2 Instances, EC2 Instances Performance, Elastic Beanstalk Applications (selected), Elastic Load Balancers, Elastic Load Balancers Performance, and Other Metrics. The main pane displays a table titled 'Elastic Beanstalk Applications (2)'. The table has columns for State (Critical for both entries), Application Name (application two and My First Elastic Beanstalk Application), Application Environment (Amazon Elastic Beanstalk Application Environment), Date Created (2/19/2015 4:52:... and 4/9/2014 7:52:1...), and Date Updated (2/19/2015 4:52:... and 4/9/2014 7:52:...). Below the table is a 'Detail View' section for 'Amazon Elastic Beanstalk Application properties of application two'. It lists the following details:

Display Name	application two
Full Path Name	application two
Region	us-west-2
Application Region	us-west-2
Application Name	application two
Application Description	
Date Created	2/19/2015 4:52:28 AM
Date Updated	2/19/2015 4:52:28 AM

AWS Elastic Beanstalk Applications Diagram View

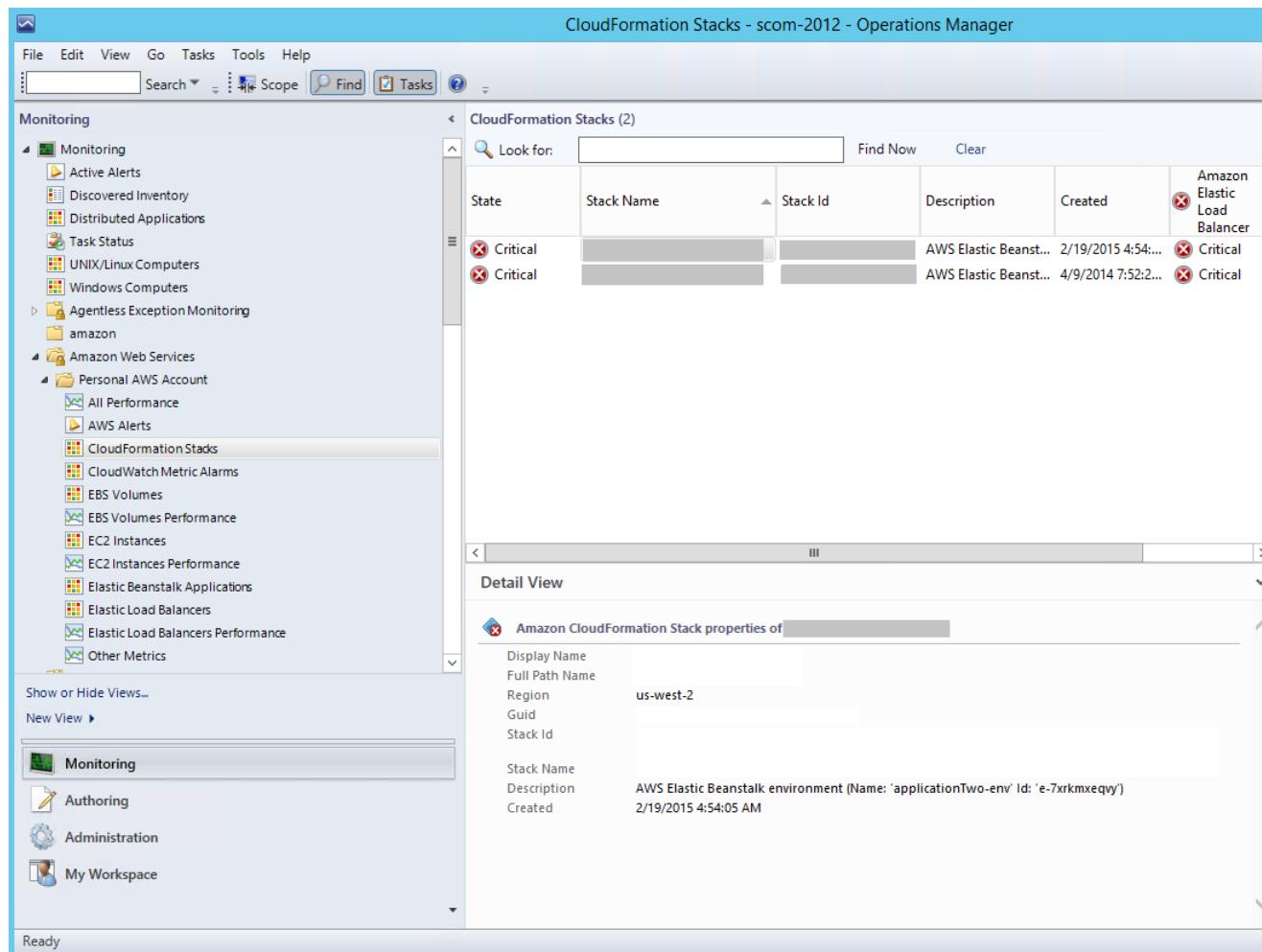
Shows the AWS Elastic Beanstalk application, application environment, application configuration, and application resources objects.



AWS CloudFormation Stacks

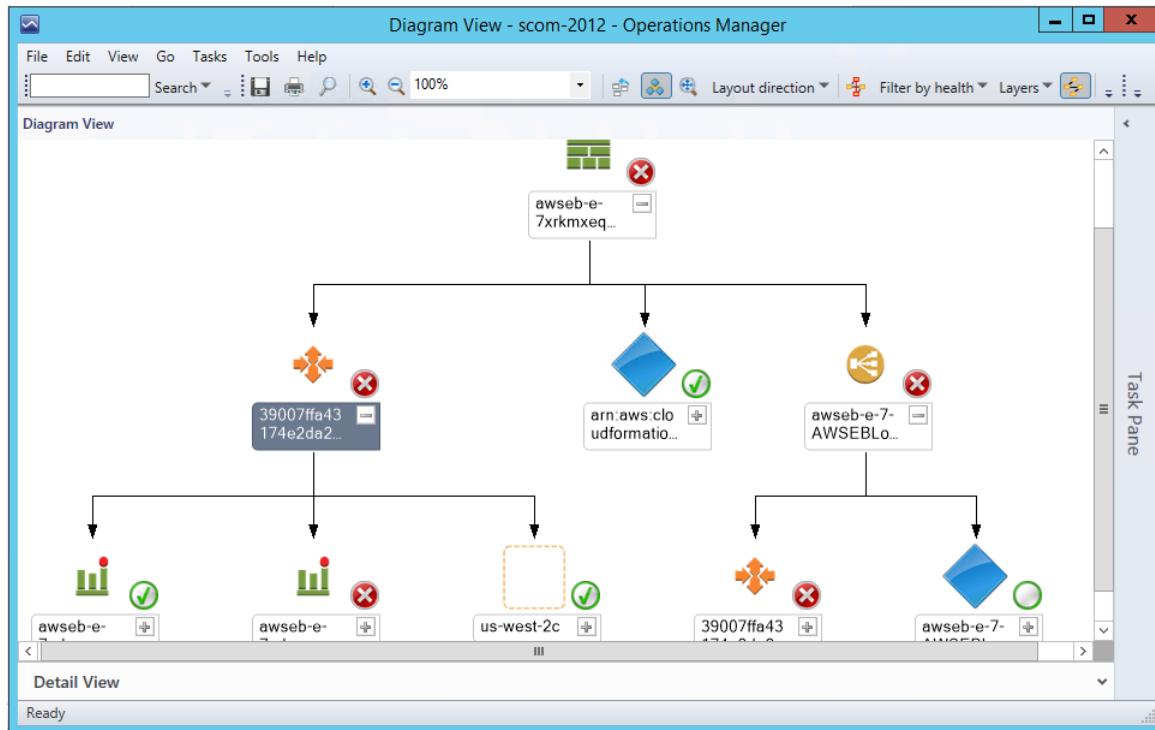
Shows the health state of all the AWS CloudFormation stacks for a particular AWS account from all regions.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Views



AWS CloudFormation Stacks Diagram View

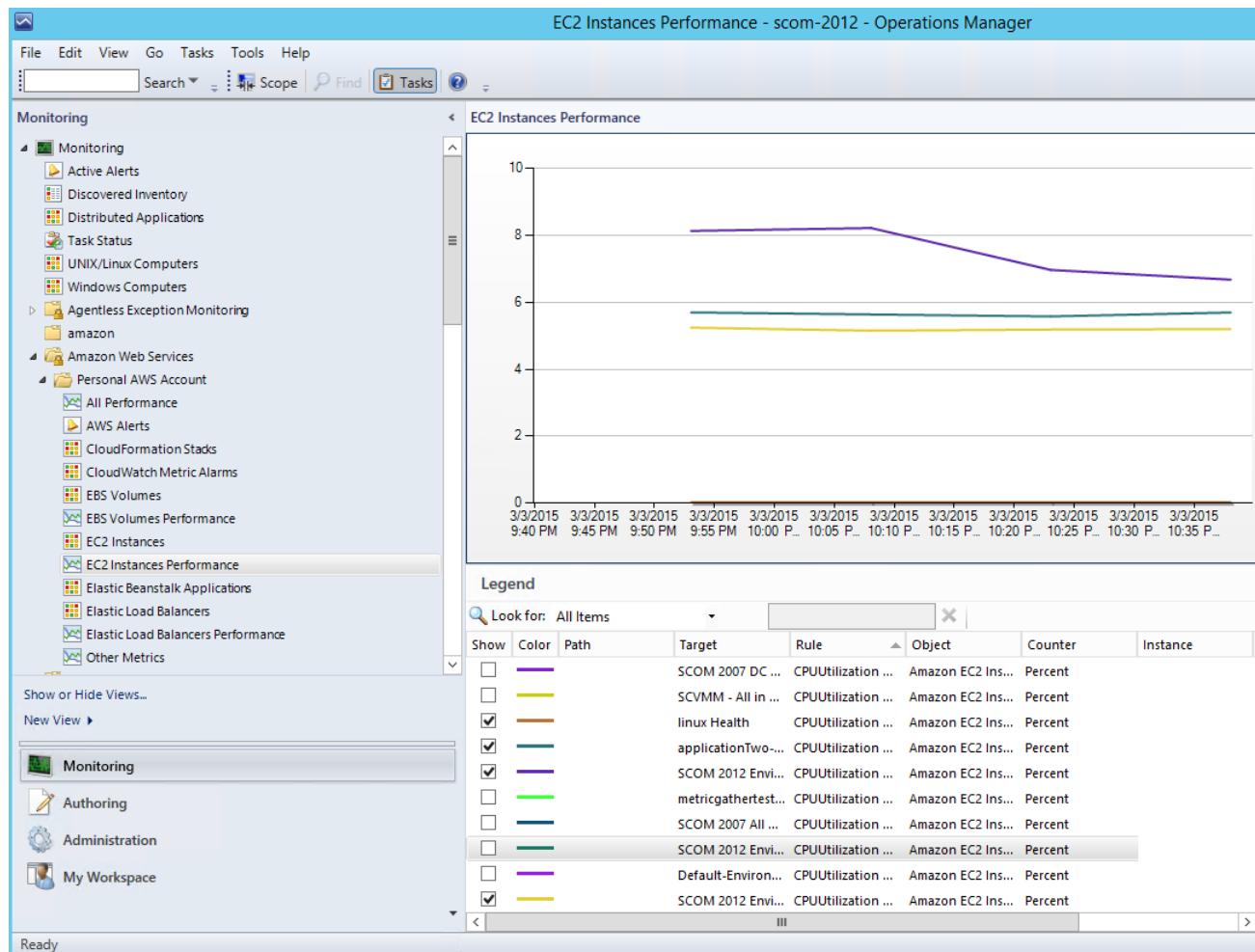
Shows the AWS CloudFormation stack relationship with other components. An AWS CloudFormation stack might contain Amazon EC2 or Elastic Load Balancing resources. The following illustration shows an example:



Amazon Performance Views

Shows the Amazon CloudWatch metrics for Amazon EC2, Amazon EBS, and Elastic Load Balancing, custom metrics, and metrics created from CloudWatch alarms. In addition, there are separate performance views for each resource. The **Other Metrics** performance view contains custom metrics, and metrics created from CloudWatch alarms. For more information about these metrics, see the [CloudWatch Metrics, Namespaces, and Dimensions Reference](#) in the *Amazon CloudWatch Developer Guide*. The following illustration shows an example.

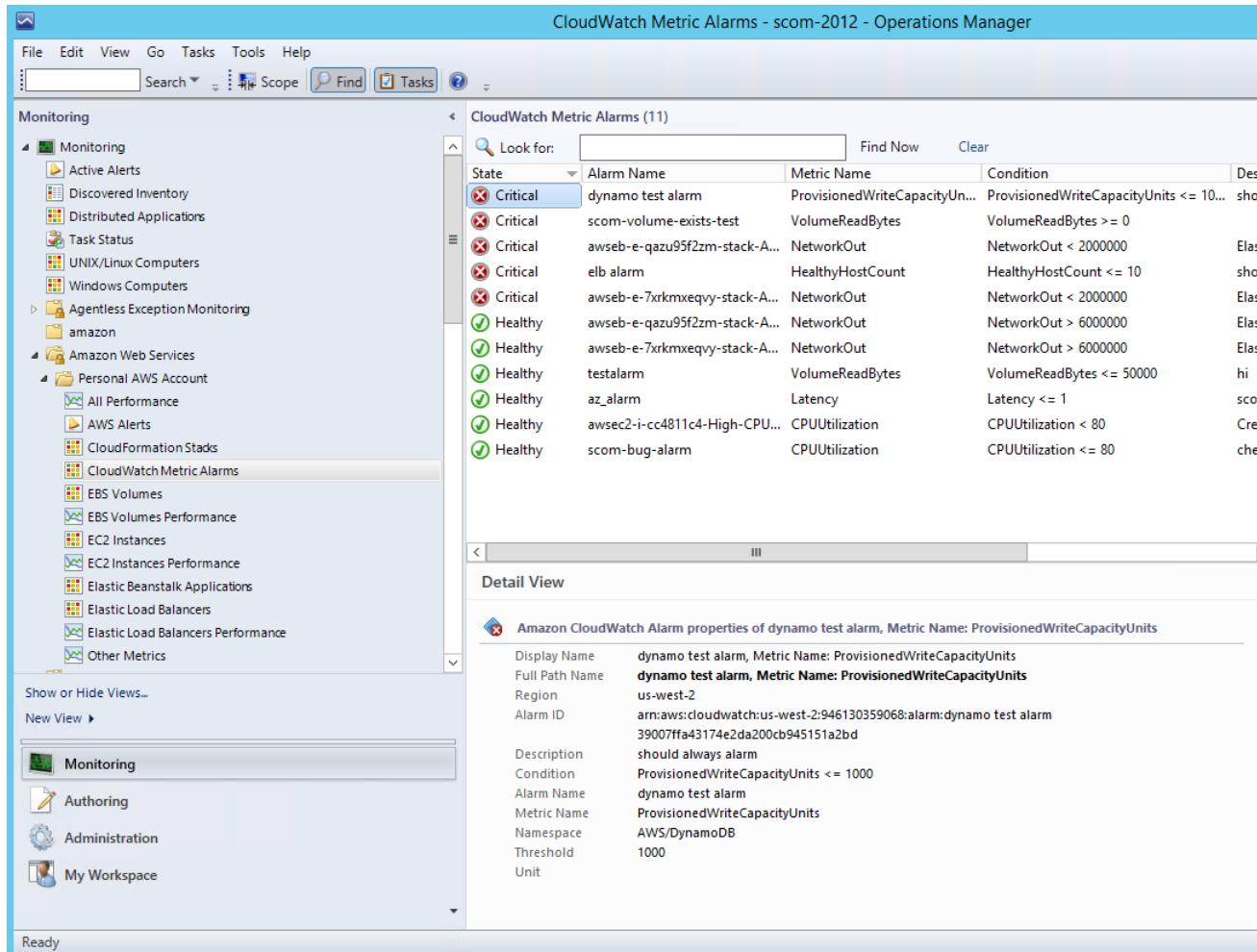
Amazon Elastic Compute Cloud
User Guide for Windows Instances
Views



Amazon CloudWatch Metric Alarms

Shows Amazon CloudWatch alarms related to the discovered AWS resources.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Views



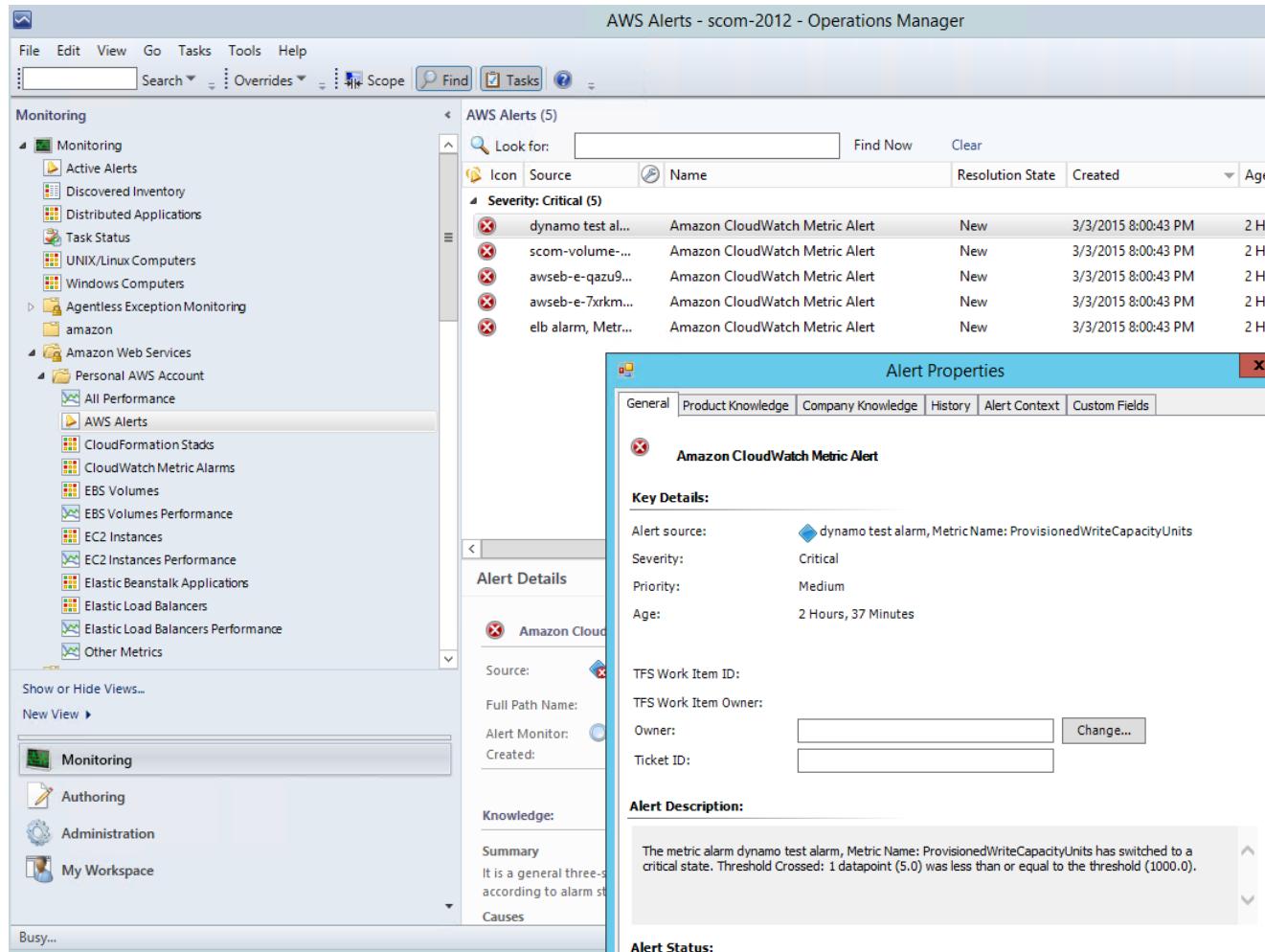
The screenshot shows the 'CloudWatch Metric Alarms - scom-2012 - Operations Manager' window. The left pane displays a navigation tree under 'Monitoring' with sections like Active Alerts, Discovered Inventory, Distributed Applications, Task Status, UNIX/Linux Computers, Windows Computers, Agentless Exception Monitoring, amazon, Amazon Web Services, Personal AWS Account, All Performance, AWS Alerts, CloudFormation Stacks, CloudWatch Metric Alarms (selected), EBS Volumes, EBS Volumes Performance, EC2 Instances, EC2 Instances Performance, Elastic Beanstalk Applications, Elastic Load Balancers, Elastic Load Balancers Performance, and Other Metrics. The right pane shows a list of 'CloudWatch Metric Alarms (11)' with columns for State, Alarm Name, Metric Name, Condition, and Description. The 'State' column uses red X icons for Critical and green checkmarks for Healthy. The 'Metric Name' column lists various metrics like ProvisionedWriteCapacityUnits, VolumeReadBytes, NetworkOut, HealthyHostCount, and Latency. The 'Condition' column contains the alarm expressions. A 'Detail View' pane at the bottom shows the properties for the 'dynamo test alarm', including its display name, full path, region, alarm ID, description, condition, alarm name, metric name, namespace, threshold, and unit.

State	Alarm Name	Metric Name	Condition	Description
Critical	dynamo test alarm	ProvisionedWriteCapacityUn...	ProvisionedWriteCapacityUnits <= 10...	should always alarm
Critical	scom-volume-exists-test	VolumeReadBytes	VolumeReadBytes >= 0	
Critical	awseb-e-qazu95f2zm-stack-A...	NetworkOut	NetworkOut < 2000000	Elastic...
Critical	elb alarm	HealthyHostCount	HealthyHostCount <= 10	Elastic...
Critical	awseb-e-7xrkmxeqvy-stack-A...	NetworkOut	NetworkOut < 2000000	Elastic...
Healthy	awseb-e-qazu95f2zm-stack-A...	NetworkOut	NetworkOut > 6000000	Elastic...
Healthy	awseb-e-7xrkmxeqvy-stack-A...	NetworkOut	NetworkOut > 6000000	Elastic...
Healthy	testalarm	VolumeReadBytes	VolumeReadBytes <= 50000	hi
Healthy	az_alarm	Latency	Latency <= 1	scom...
Healthy	awsec2-i-cc4811c4-High-CPU...	CPUUtilization	CPUUtilization < 80	Create...
Healthy	scom-bug-alarm	CPUUtilization	CPUUtilization <= 80	check...

AWS Alerts

Shows the alerts that the AWS management pack produces when the health of an object is in a critical state.

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Views



The screenshot shows the AWS Alerts interface within System Center Operations Manager 2012. The left pane displays a navigation tree under 'Monitoring' and 'Amazon Web Services'. The right pane shows a list of 'AWS Alerts (5)' with a 'Severity: Critical (5)' filter applied. A detailed view of an 'Amazon CloudWatch Metric Alert' is shown in a modal window.

Icon	Source	Name	Resolution State	Created	Age
[Icon]	dynamo test al...	Amazon CloudWatch Metric Alert	New	3/3/2015 8:00:43 PM	2 H
[Icon]	scom-volume-...	Amazon CloudWatch Metric Alert	New	3/3/2015 8:00:43 PM	2 H
[Icon]	awseb-e-qazu9...	Amazon CloudWatch Metric Alert	New	3/3/2015 8:00:43 PM	2 H
[Icon]	awseb-e-7xrkm...	Amazon CloudWatch Metric Alert	New	3/3/2015 8:00:43 PM	2 H
[Icon]	elb alarm, Metr...	Amazon CloudWatch Metric Alert	New	3/3/2015 8:00:43 PM	2 H

Alert Properties

General tab selected. Alert Details:

- Source: Amazon CloudWatch Metric Alert
- Full Path Name: dynamo test alarm, Metric Name: ProvisionedWriteCapacityUnits
- Severity: Critical
- Priority: Medium
- Age: 2 Hours, 37 Minutes

TFS Work Item ID: [Input Field] Change...

TFS Work Item Owner: [Input Field]

Owner: [Input Field]

Ticket ID: [Input Field]

Alert Description:

The metric alarm dynamo test alarm, Metric Name: ProvisionedWriteCapacityUnits has switched to a critical state. Threshold Crossed: 1 datapoint (5.0) was less than or equal to the threshold (1000.0).

Alert Status:

Watcher Nodes (System Center Operations Manager 2007 R2)

View the health state of the watcher nodes across all of the AWS accounts that are being monitored. A **Healthy** state means that the watcher node is configured correctly and can communicate with AWS.



Discoveries

Discoveries are the AWS resources that are monitored by the AWS Management Pack. The AWS Management Pack discovers the following objects:

- Amazon EC2 instances
- EBS volumes
- ELB load balancers
- AWS CloudFormation stacks
- Amazon CloudWatch alarms
- AWS Elastic Beanstalk applications
- Amazon EC2 Auto Scaling groups and Availability Zones

Amazon CloudWatch metrics are generated for the following resources:

- Amazon EC2 instance
- EBS volume
- Elastic Load Balancing
- Custom Amazon CloudWatch metrics
- Metrics from existing Amazon CloudWatch alarms

For Amazon CloudWatch metrics discovery, the following guidelines apply:

- AWS CloudFormation stacks do not have any default Amazon CloudWatch metrics.
- Stopped Amazon EC2 instances or unused Amazon EBS volumes do not generate data for their default Amazon CloudWatch metrics.
- After starting an Amazon EC2 instance, it can take up to 30 minutes for the Amazon CloudWatch metrics to appear in Operations Manager.

- Amazon CloudWatch retains the monitoring data for two weeks, even if your AWS resources have been terminated. This data appears in Operations Manager.
- An existing Amazon CloudWatch alarm for a resource that is not supported will create a metric and be associated with the Amazon CloudWatch alarm. These metric can be viewed in the Other Metrics performance view.

The AWS Management Pack also discovers the following relationships:

- AWS CloudFormation stack and its Elastic Load Balancing or Amazon EC2 resources
- Elastic Load Balancing load balancer and its EC2 instances
- Amazon EC2 instance and its EBS volumes
- Amazon EC2 instance and its operating system
- AWS Elastic Beanstalk application and its environment, configuration, and resources

The AWS Management Pack automatically discovers the relationship between an EC2 instance and the operating system running on it. To discover this relationship, the Operations Manager Agent must be installed and configured on the instance and the corresponding operating system management pack must be imported in Operations Manager.

Discoveries run on the management servers in the resource pool (System Center 2012) or the watcher node (System Center 2007 R2).

Discovery	Interval (seconds)
Amazon Resources Discovery (SCOM 2012) Discovers EC2 instances, Amazon EBS volumes, load balancers, and CloudFront stacks.	14400
AWS Elastic Beanstalk Discovery Discovers AWS Elastic Beanstalk and its relationship with environment, resources, and configuration.	14400
CloudWatch Alarms Discovery Discovers alarms generated using CloudWatch metrics.	900
Custom CloudWatch Metric Discovery Discovers custom CloudWatch metrics.	14400
Watcher Node Discovery (SCOM 2007 R2) Targets the root management server and creates the watcher node objects.	14400

Monitors

Monitors are used to measure the health of your AWS resources. Monitors run on the management servers in the resource pool (System Center 2012) or the watcher node (System Center 2007 R2).

Monitor	Interval (seconds)
AWS CloudFormation Stack Status	900
Amazon CloudWatch Metric Alarm	300
Amazon EBS Volume Status	900
Amazon EC2 Instance Status	900
Amazon EC2 Instance System Status	900
AWS Elastic Beanstalk Status	900
Watcher Node to Amazon Cloud Connectivity (SCOM 2007 R2)	900

Rules

Rules create alerts (based on Amazon CloudWatch metrics) and collect data for analysis and reporting.

Rule	Interval (seconds)
AWS Resource Discovery Rule (SCOM 2007 R2) Targets the watcher node and uses the AWS API to discover objects for the following AWS resources: EC2 instances, EBS volumes, load balancers, and AWS CloudFormation stacks. (CloudWatch metrics or alarms are not discovered). After discovery is complete, view the objects in the Not Monitored state.	14400
Amazon Elastic Block Store Volume Performance Metrics Data Collection Rule	900
Amazon EC2 Instance Performance Metrics Data Collection Rule	900
Elastic Load Balancing Balancing Performance Metrics Data Collection Rule	900
Custom CloudWatch Metric Data Collection Rule	900

Events

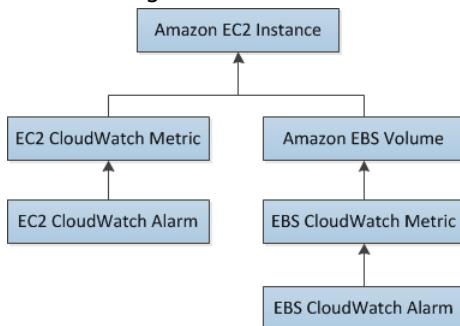
Events report on activities that involve the monitored resources. Events are written to the Operations Manager event log.

Event ID	Description
4101	Amazon EC2 Instance Discovery (General Discovery) finished
4102	Elastic Load Balancing Metrics Discovery, Amazon EBS Volume Metrics Discovery, Amazon EC2 Instance Metrics Discovery finished
4103	Amazon CloudWatch Metric Alarms Discovery finished

Event ID	Description
4104	Amazon Windows Computer Discovery finished
4105	Collecting Amazon Metrics Alarm finished
4106	EC2 Instance Computer Relation Discovery finished
4107	Collecting AWS CloudFormation Stack State finished
4108	Collecting Watcher Node Availability State finished
4109	Amazon Metrics Collection Rule finished
4110	Task to change Amazon Instance State finished
4111	EC2 Instance Status Monitor State finished
4112	Amazon EBS Volume Status Monitor State finished
4113	Amazon EC2 Instance Scheduled Events Monitor State calculated
4114	Amazon EBS Scheduled Events Monitor State calculated
4115	Elastic Beanstalk Discovery finished
4116	Elastic Beanstalk Environment Status State calculated
4117	Elastic Beanstalk Environment Operational State calculated
4118	Elastic Beanstalk Environment Configuration State calculated

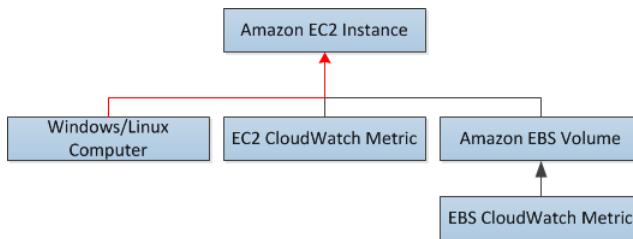
Health Model

The following illustration shows the health model defined by the AWS Management Pack.



The health state for a CloudWatch alarm is rolled up to its corresponding CloudWatch metric. The health state for a CloudWatch metric for Amazon EC2 is rolled up to the EC2 instance. Similarly, the health state for the CloudWatch metrics for Amazon EBS is rolled up to the Amazon EBS volume. The health states for the Amazon EBS volumes used by an EC2 instance are rolled up to the EC2 instance.

When the relationship between an EC2 instance and its operating system has been discovered, the operating system health state is rolled up to the EC2 instance.

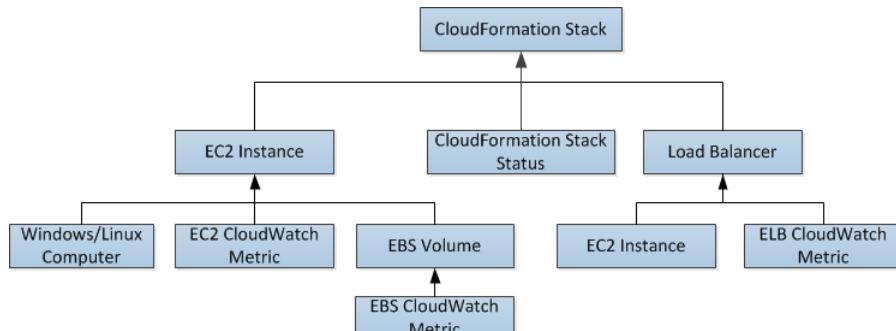


The health state of an AWS CloudFormation stack depends on the status of the AWS CloudFormation stack itself and the health states of its resources, namely the load balancers and EC2 instances.

The following table illustrates how the status of the AWS CloudFormation stack corresponds to its health state.

Health State	AWS CloudFormation Stack Status	Notes
Error	CREATE_FAILED DELETE_IN_PROGRESS DELETE_FAILED UPDATE_ROLLBACK_FAILED	Most likely usable
Warning	UPDATE_ROLLBACK_IN_PROGRESS UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_ROLLBACK_COMPLETE	Recovering after some problem
Healthy	CREATE_COMPLETE UPDATE_IN_PROGRESS UPDATE_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_COMPLETE	Usable

The full health model for an AWS CloudFormation stack is as follows:



Customizing the AWS Management Pack

To change the frequency of discoveries, rules, and monitors, you can override the interval time (in seconds).

To change frequency

1. In the **Operations Manager** toolbar, click **Go**, and then click **Authoring**.
2. In the **Authoring** pane, expand **Management Pack Objects** and then click the object to change (for example, **Object Discoveries**, **Rules**, or **Monitors**).
3. In the toolbar, click **Scope**.
4. In the **Scope Management Pack Objects** dialog box, click **View all targets**.
5. To limit the scope to Amazon objects, type Amazon in the **Look for** field.
6. Select the object want to configure and click **OK**.
7. In the **Operations Manager** center pane, right-click the object to configure, click **Overrides**, and then click the type of override you want to configure.
8. Use the **Override Properties** dialog box to configure different values and settings for objects.

Tip

To disable a discovery, rule, or monitoring object right-click the object to disable in the **Operations Manager** center pane, click **Overrides**, and then click **Disable the Rule**. You might disable rules if, for example, you do not run AWS Elastic Beanstalk applications or use custom Amazon CloudWatch metrics.

For information about creating overrides, see [Tuning Monitoring by Using Targeting and Overrides](#) on the *Microsoft TechNet* website.

For information about creating custom rules and monitors, see [Authoring for System Center 2012 - Operations Manager](#) or [System Center Operations Manager 2007 R2 Management Pack Authoring Guide](#) on the *Microsoft TechNet* website.

Upgrading the AWS Management Pack

The procedure that you'll use to update AWS Management Pack depends on the version of System Center.

System Center 2012

To upgrade the AWS Management Pack

1. On the [AWS Add-Ins for Microsoft System Center](#) website, click **SCOM 2012**. Download **AWS-SCOM-MP-2.0-2.5.zip** to your computer and unzip it. The **.zip** file includes **Amazon.AmazonWebServices.mpb**.
2. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.
3. In the **Tasks** pane, click **Import Management Packs**.
4. On the **Select Management Packs** page, click **Add**, and then click **Add from disk**.
5. In the **Select Management Packs to import** dialog box, select the **Amazon.AmazonWebServices.mpb** file from the location where you downloaded it, and then click **Open**.
6. On the **Select Management Packs** page, under **Import list**, select the **Amazon Web Services** management pack, and then click **Install**.

If the **Install** button is disabled, upgrading to the current version is not supported and you must uninstall the AWS Management Pack before you can install the current version. For more information, see [Uninstalling the AWS Management Pack \(p. 950\)](#).

System Center 2007 R2

To upgrade the AWS Management Pack

1. On the Management Server, go to the [AWS Add-Ins for Microsoft System Center](#) website and click **SCOM 2007**. Save **AWS-MP-Setup-2.5.msi**, and then run it.
2. Click **Next** and follow the directions to upgrade the components that you installed previously.
3. If your root management server, Operations console, and watcher node are on different computers, you must download and run the setup program on each computer.
4. On the watcher node, open a Command Prompt window as an administrator and run the following commands.

```
C:\> net stop HealthService
The System Center Management service is stopping.
The System Center Management service was stopped successfully.

C:\> net start HealthService
The System Center Management service is starting.
The System Center Management service was started successfully.
```

5. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.
 6. In the **Actions** pane, click **Import Management Packs**.
 7. On the **Select Management Packs** page, click **Add**, and then click **Add from disk**.
 8. In the **Select Management Packs to import** dialog box, change the directory to **C:\Program Files (x86)\Amazon Web Services Management Pack**, select the **Amazon.AmazonWebServices.mp** file, and then click **Open**.
 9. On the **Select Management Packs** page, under **Import list**, select the **Amazon Web Services** management pack, and then click **Install**.
- If the **Install** button is disabled, upgrading to the current version is not supported and you must uninstall AWS Management Pack first. For more information, see [Uninstalling the AWS Management Pack \(p. 950\)](#).

Uninstalling the AWS Management Pack

If you need to uninstall the AWS Management Pack, use the following procedure.

System Center 2012

To uninstall the AWS Management Pack

1. In the Operations console, on the **Go** menu, click **Administration**, and then click **Management Packs**.
2. Right-click **Amazon Web Services** and select **Delete**.
3. In the **Dependent Management Packs** dialog box, note the dependent management packs, and then click **Close**.
4. Right-click the dependent management pack and select **Delete**.
5. Right-click **Amazon Web Services** and select **Delete**.

System Center 2007 R2

To uninstall the AWS Management Pack

1. Complete steps 1 through 5 described for System Center 2012 in the previous section.
2. From Control Panel, open Programs and Features. Select **Amazon Web Services Management Pack** and then click **Uninstall**.
3. If your root management server, Operations console, and watcher node are on different computers, you must repeat this process on each computer.

Troubleshooting the AWS Management Pack

The following are common errors, events, and troubleshooting steps.

Contents

- [Errors 4101 and 4105 \(p. 951\)](#)
- [Error 4513 \(p. 951\)](#)
- [Event 623 \(p. 952\)](#)
- [Events 2023 and 2120 \(p. 952\)](#)
- [Event 6024 \(p. 952\)](#)
- [General Troubleshooting for System Center 2012 — Operations Manager \(p. 952\)](#)
- [General Troubleshooting for System Center 2007 R2 \(p. 953\)](#)

Errors 4101 and 4105

If you receive one of the following errors, you must upgrade the AWS Management Pack. For more information, see [Upgrading the AWS Management Pack \(p. 949\)](#).

```
Error 4101
Exception calling "DescribeVolumes" with "1" argument(s): "AWS was not able to validate
the
provided access credentials"
```

```
Error 4105
Exception calling "DescribeApplications" with "0" argument(s): "The security token
included
in the request is invalid"
```

Error 4513

If you receive one of the following error, you must upgrade the AWS Management Pack. For more information, see [Upgrading the AWS Management Pack \(p. 949\)](#).

```
Error 4513
The callback method DeliverDataToModule failed with exception "Resolution of the
dependency
failed, type = "Amazon.SCOM.SDK.Interfaces.IMonitorSdk", name = "(none)".
Exception occurred while: Calling constructor Amazon.SCOM.SDK.CloudWatch.AwsMonitorSdk
(System.String awsAccessKey, System.String awsSecretKey).
Exception is: InvalidOperationException - Collection was modified; enumeration operation
```

may not execute.

Event 623

If you find the following event in the Windows event log, follow the solution described in [KB975057](#).

```
Event ID: 623
HealthService (process_id) The version store for instance instance ("name") has reached
its maximum size of size MB. It is likely that a long-running transaction is preventing
cleanup of the version store and causing it to build up in size. Updates will be rejected
until the long-running transaction has been completely committed or rolled back.
Possible long-running transaction:
SessionId: id
Session-context: value
Session-context ThreadId: id
Cleanup: value
```

Events 2023 and 2120

If you find the following events in the Windows event log, see [Event ID 2023 and 2120](#) for more information.

```
Event ID: 2023
The Health Service has removed some items from the send queue for management group
"Servers"
since it exceeded the maximum allowed size of size megabytes.
```

```
Event ID: 2120
The Health Service has deleted one or more items for management group "Servers" which
could
not be sent in 1440 minutes.
```

Event 6024

If you find the following event in the Windows event log, see [Health Service Restarts](#) for more information.

```
Event ID: 6024
LaunchRestartHealthService.js : Launching Restart Health Service. Health Service exceeded
Process\Handle Count or Private Bytes threshold.
```

General Troubleshooting for System Center 2012 — Operations Manager

Try the following to resolve any issues.

- Verify that you have installed the latest Update Rollup for System Center 2012 — Operations Manager. The AWS Management Pack requires at least Update Rollup 1.
- Ensure that you have configured the AWS Management Pack after importing it by running the Add Monitoring Wizard. For more information, see [Step 1: Installing the AWS Management Pack \(p. 920\)](#).
- Verify that you have waited long enough for the AWS resources to be discovered (10–20 minutes).
- Verify that the management servers are configured properly.

- Management servers must have Internet connectivity.
- The action account for a management server must have local administrator privileges on the management server.
- The management server must have the .NET Framework 4.5. or later.
- Verify that the AWS Run As account is valid.
 - The values for the access key ID and secret access key are correct.
 - The access keys are active: In the AWS Management Console, click your name in the navigation bar and then click **Security Credentials**.
- The IAM user has at least read-only access permission. Note that read-only access allows the user actions that do not change the state of a resource, such as monitoring, but do not allow the user actions like launching or stopping an instance.
 - If an Amazon CloudWatch metric shows as **Not Monitored**, check whether at least one Amazon CloudWatch alarm has been defined for that Amazon CloudWatch metric.
 - For further troubleshooting, use the information in the event logs.
 - Check the Operations Manager event log on the management server. For more information, see [Events \(p. 946\)](#) for a list of the events that the AWS Management Pack writes to the Operations Manager event log.

General Troubleshooting for System Center 2007 R2

Try the following to resolve any issues.

- Ensure that you have configured the AWS Management Pack after importing it by running the Add Monitoring Wizard. For more information, see [Step 1: Installing the AWS Management Pack \(p. 920\)](#).
- Verify that you have waited long enough for the AWS resources to be discovered (10–20 minutes).
- Verify that the watcher node is configured properly.
 - The proxy agent is enabled. For more information, see [Step 2: Configuring the Watcher Node \(p. 922\)](#).
 - The watcher node has Internet connectivity.
 - The action account for the watcher node has local administrator privileges.
 - The watcher node must have the .NET Framework 3.5.1 or later.
- Verify that the watcher node is healthy and resolve all alerts. For more information, see [Views \(p. 930\)](#).
- Verify that the AWS Run As account is valid.
 - The values for the access key ID and secret access key are correct.
 - The access keys are active: In the AWS Management Console, click your name in the navigation bar and then click **Security Credentials**.
- The IAM user has at least read-only access permission. Note that read-only access allows the user actions that do not change the state of a resource, such as monitoring, but do not allow the user actions like launching or stopping an instance.
 - If an Amazon CloudWatch metric shows as **Not Monitored**, check whether at least one Amazon CloudWatch alarm has been defined for that Amazon CloudWatch metric.
 - For further troubleshooting, use the information in the event logs.
 - Check the Operations Manager event log on the management server as well as the watcher node. For more information, see [Events \(p. 946\)](#) for a list of the events that the AWS Management Pack writes to the Operations Manager event log.

Using EC2Rescue for Windows Server

EC2Rescue for Windows Server is an easy-to-use tool that you run on an Amazon EC2 Windows Server instance to diagnose and troubleshoot possible problems. It is valuable for collecting log files and troubleshooting issues and also proactively searching for possible areas of concern. It can even examine Amazon EBS root volumes from other instances and collect relevant logs for troubleshooting Windows Server instances using that volume.

EC2Rescue for Windows Server has two different modules: a data collector module that collects data from all different sources, and an analyzer module that parses the data collected against a series of predefined rules to identify issues and provide suggestions.

The EC2Rescue for Windows Server tool only runs on Amazon EC2 instances running Windows Server 2008 R2 and later. When the tool starts, it checks whether it is running on an Amazon EC2 instance.

Note

If you are using a Linux instance, see [EC2Rescue for Linux](#).

Contents

- [Using EC2Rescue for Windows Server GUI \(p. 954\)](#)
- [Using EC2Rescue for Windows Server with the Command Line \(p. 957\)](#)
- [Using EC2Rescue for Windows Server with Systems Manager Run Command \(p. 962\)](#)

Using EC2Rescue for Windows Server GUI

EC2Rescue for Windows Server is able to perform the following analysis on an offline instance:

Option	Description
Diagnose and Rescue	<p>The following service settings can be detected and modified:</p> <ul style="list-style-type: none">• System Time<ul style="list-style-type: none">• RealTimeisUniversal - Detects whether the RealTimeisUniversal registry key is enabled. If disabled, Windows system time drifts when the timezone is set to a value other than UTC.• Windows Firewall<ul style="list-style-type: none">• Domain networks - Detects whether this Windows Firewall profile is enabled or disabled.• Private networks - Detects whether this Windows Firewall profile is enabled or disabled.

Option	Description
	<ul style="list-style-type: none"> • Guest or public networks - Detects whether this Windows Firewall profile is enabled or disabled. • Remote Desktop • Service Start - Detects whether the Remote Desktop service is enabled. • Remote Desktop Connections - Detects whether this is enabled. • TCP Port - Detects which port the Remote Desktop service is listening on. • EC2Config <ul style="list-style-type: none"> • Installation - Detects which EC2Config version is installed. • Service Start - Detects whether the EC2Config service is enabled. • Ec2SetPassword - Generates a new administrator password. • Ec2HandleUserData - Allows you to execute a user data script on the next boot of the instance. • Network Interface <ul style="list-style-type: none"> • DHCP Service Startup - Detects whether the DHCP service is enabled. • Ethernet detail - Displays information about the network driver version, if detected. • DHCP on Ethernet - Detects whether DHCP is enabled.
Restore	<p>Perform one of the following actions:</p> <ul style="list-style-type: none"> • Last Known Good Configuration - Attempts to boot the instance into the last known bootable state. • Restore registry from backup - Restores the registry from \Windows\System32\config\RegBack.
Capture Logs	Allows you to capture logs on the instance for analysis.

EC2Rescue for Windows Server is able to collect the following data from active and offline instances:

Item	Description
Event Log	Collects application, system, and EC2Config event logs.

Item	Description
Memory Dump	Collects any memory dump files that exist on the instance.
EC2Config File	Collects log files generated by the EC2Config service.
EC2Launch File	Collects log files generated by the EC2Launch scripts.
SSM Agent File	Collects log files generated by the SSM agent.
Sysprep Log	Collects log files generated by the Windows System Preparation tool.
Driver SetupAPI Log	Collects Windows SetupAPI logs (setupapi.dev.log and setupapi.setup.log).
Registry	Collects SYSTEM and SOFTWARE hives.
System Information	Collects MSInfo32.
Boot Configuration	Collects HKEY_LOCAL_MACHINE\BCD00000000 hive.
Windows Update Log	Collects information about the updates that are installed on the instance.
	Note Windows Update logs are not captured on Windows Server 2016 instances.

Video Walkthrough

Brandon shows you how to use the Diagnose and Rescue feature of EC2Rescue for Windows Server:

[AWS Knowledge Center Videos: How do I use the Diagnose and Rescue feature of EC2Rescue?](#)

Analyzing an Offline Instance

The **Offline Instance** option is useful for debugging boot issues with Windows instances.

To perform an action on an offline instance

1. From a working Windows Server instance, download the [EC2Rescue for Windows Server](#) tool and extract the files.
2. Stop the faulty instance, if it is not stopped already.
3. Detach the EBS root volume from the faulty instance and attach the volume to a working Windows instance that has EC2Rescue for Windows Server installed.
4. Run the EC2Rescue for Windows Server tool on the working instance and choose **Offline Instance**.
5. Select the disk of the newly mounted volume and choose **Next**.
6. Confirm the disk selection and choose **Yes**.
7. Choose the offline instance option to perform and choose **Next**.

The EC2Rescue for Windows Server tool scans the volume and collects troubleshooting information based on the selected log files.

Collecting Data from an Active Instance

You can collect logs and other data from an active instance.

To collect data from an active instance

1. Connect to your Windows instance.
2. Download the [EC2Rescue for Windows Server](#) tool to your Windows instance and extract the files.
3. Open the EC2Rescue for Windows Server application and accept the license agreement.
4. Choose **Next, Current instance, Capture logs**.
5. Select the data items to collect and choose **Collect....** Read the warning and choose **Yes** to continue.
6. Choose a file name and location for the ZIP file and choose **Save**.
7. After EC2Rescue for Windows Server completes, choose **Open Containing Folder** to view the ZIP file.
8. Choose **Finish**.

Using EC2Rescue for Windows Server with the Command Line

The EC2Rescue for Windows Server command line interface (CLI) allows you to run an EC2Rescue for Windows Server plugin (referred as an "action") programmatically.

The EC2Rescue for Windows Server tool has two execution modes:

- **/online**—This allows you to take action on the instance that EC2Rescue for Windows Server is installed on, such as collect log files.
- **/offline:<device_id>**—This allows you to take action on the offline root volume that is attached to a separate Amazon EC2 Windows instance, on which you have installed EC2Rescue for Windows Server.

Download the [EC2Rescue for Windows Server](#) tool to your Windows instance and extract the files. You can view the help file with the following command:

```
EC2RescueCmd.exe /help
```

EC2Rescue for Windows Server can perform the following actions on an Amazon EC2 Windows instance:

- [Collect Action \(p. 957\)](#)
- [Rescue Action \(p. 959\)](#)
- [Restore Action \(p. 961\)](#)

Collect Action

EC2Rescue for Windows Server is able to collect the following data from active and offline instances. You can collect all logs, an entire log group, or an individual log within a group.

Log Group	Available Logs	Description
all		Collects all available logs.
system-info	'MSInfo32 Output'	Collects MSInfo32.
eventlog	<ul style="list-style-type: none"> 'Application' 'System' 'EC2ConfigService' 	Collects application, system, and EC2Config event logs.
memory-dump	<ul style="list-style-type: none"> 'Memory Dump File' 'Mini Dump Files' 	Collects any memory dump files that exist on the instance.
ec2config	<ul style="list-style-type: none"> 'Log Files' 'Configuration Files' 	Collects log files generated by the EC2Config service.
ec2launch	<ul style="list-style-type: none"> 'Logs' 'Config' 	Collects log files generated by the EC2Launch scripts.
ssm-agent	'Log Files'	Collects log files generated by the SSM agent.
sysprep	'Log Files'	Collects log files generated by the Windows System Preparation tool.
driver-setup	<ul style="list-style-type: none"> 'SetupAPI Log Files' 'DPInst Log File' 'AWS PV Setup Log File' 	Collects Windows SetupAPI logs (setupapi.dev.log and setupapi.setup.log).
registry	<ul style="list-style-type: none"> 'SYSTEM' 'SOFTWARE' 'BCD' 	Collects SYSTEM and SOFTWARE hives.
gpresult	'GPResult Output'	Collects a Group Policy report.
egpu	<ul style="list-style-type: none"> 'Event Log' 'System Files' 	Collects event logs related to elastic GPUs.
boot-config	'BCDEDIT Output'	Collects HKEY_LOCAL_MACHINE \BCD00000000 hive.
windows-update	'Log Files'	Collects information about the updates that are installed on the instance. Note Windows Update logs are not captured on Windows Server 2016 instances.

The following are the available options:

- **/output:<outputFilePath>** - Required destination file path location to save collected log files in zip format.

- **/no-offline** - Optional attribute used in offline mode. Does not set the volume offline after completing the action.
- **/no-fix-signature** - Optional attribute used in offline mode. Does not fix a possible disk signature collision after completing the action.

Examples

The following are examples using the EC2Rescue for Windows Server CLI.

Online Mode Examples

Collect all available logs:

```
EC2RescueCmd /accepteula /online /collect:all /output:<outputFilePath>
```

Collect only a specific log group:

```
EC2RescueCmd /accepteula /online /collect:ec2config /output:<outputFilePath>
```

Collect individual logs within a log group:

```
EC2RescueCmd /accepteula /online /collect:'ec2config.Log Files,driver-setup.SetupAPI Log Files' /output:<outputFilePath>
```

Offline Mode Examples

Collect all available logs from an EBS volume. The volume is specified by the device_id value.

```
EC2RescueCmd /accepteula /offline:xvdf /collect:all /output:<outputFilePath>
```

Collect only a specific log group:

```
EC2RescueCmd /accepteula /offline:xvdf /collect:ec2config /output:<outputFilePath>
```

Rescue Action

EC2Rescue for Windows Server is able to attempt to detect and modify the following service settings to attempt to fix possible issues:

Service Group	Available Actions	Description
all		
system-time	'RealTimeIsUniversal'	<p>System Time</p> <ul style="list-style-type: none">• RealTimeisUniversal<ul style="list-style-type: none">- Detects whether the RealTimeisUniversal registry key is enabled. If disabled, Windows system time drifts when the timezone is set to a value other than UTC.

Service Group	Available Actions	Description
firewall	<ul style="list-style-type: none"> 'Domain networks' 'Private networks' 'Guest or public networks' 	Windows Firewall <ul style="list-style-type: none"> Domain networks - Detects whether this Windows Firewall profile is enabled or disabled. Private networks - Detects whether this Windows Firewall profile is enabled or disabled. Guest or public networks - Detects whether this Windows Firewall profile is enabled or disabled.
rdp	<ul style="list-style-type: none"> 'Service Start' 'Remote Desktop Connections' 'TCP Port' 	Remote Desktop <ul style="list-style-type: none"> Service Start - Detects whether the Remote Desktop service is enabled. Remote Desktop Connections - Detects whether this is enabled. TCP Port - Detects which port the Remote Desktop service is listening on.
ec2config	<ul style="list-style-type: none"> 'Service Start' 'Ec2SetPassword' 'Ec2HandleUserData' 	EC2Config <ul style="list-style-type: none"> Service Start - Detects whether the EC2Config service is enabled. Ec2SetPassword - Generates a new administrator password. Ec2HandleUserData - Allows you to execute a user data script on the next boot of the instance.
ec2launch	'Reset Administrator Password'	Generates a new Windows administrator password.
network	'DHCP Service Startup'	Network Interface <ul style="list-style-type: none"> DHCP Service Startup - Detects whether the DHCP service is enabled.

The following are the available options:

- **/level:<level>** - Optional attribute for the check level that the action should trigger. Allowed values are: information, warning, error, all. By default, it is set to error.
- **/check-only** - Optional attribute that generates a report but makes no modifications to the offline volume.

- **/no-offline** - Optional attribute that prevents the volume from being set offline after completing the action.
- **/no-fix-signature** - Optional attribute that does not fix a possible disk signature collision after completing the action.

Rescue Examples

The following are examples using the EC2Rescue for Windows Server CLI. The volume is specified using the device_id value.

Attempt to fix all identified issues on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:all
```

Attempt to fix all issues within a service group on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:firewall
```

Attempt to fix a specific item within a service group on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:rdp.'Service Start'
```

Specify multiple issues to attempt to fix on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:'system-
time.RealTimeIsUniversal,ec2config.Service Start'
```

Restore Action

EC2Rescue for Windows Server is able to detect and modify the following service settings to attempt to fix possible issues:

Service Group	Available Actions	Description
Restore Last Known Good Configuration	lkgc	Last Known Good Configuration - Attempts to boot the instance into the last known bootable state.
Restore Windows registry from latest backup	regback	Restore registry from backup - Restores the registry from \Windows\System32\config\RegBack.

The following are the available options:

- **/no-offline**—Optional attribute that prevents the volume from being set offline after completing the action.
- **/no-fix-signature**—Optional attribute that does not fix a possible disk signature collision after completing the action.

Restore Examples

The following are examples using the EC2Rescue for Windows Server CLI. The volume is specified using the device_id value.

Restore last known good configuration on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /restore:lkgc
```

Restore the last Windows registry backup on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /restore:regback
```

Using EC2Rescue for Windows Server with Systems Manager Run Command

AWS Support provides you with a Systems Manager Run Command document to interface with your Systems Manager-enabled instance to run EC2Rescue for Windows Server. The Run Command document is called `AWSSupport-RunEC2RescueForWindowsTool`.

This Systems Manager Run Command document performs the following tasks:

- Downloads and verifies EC2Rescue for Windows Server.
- Imports a PowerShell module to ease your interaction with the tool.
- Runs EC2RescueCmd with the provided command and parameters.

The Systems Manager Run Command document accepts three parameters:

- **Command**—The EC2Rescue for Windows Server action. The current allowed values are:
 - **ResetAccess**—Resets the local Administrator password. The local Administrator password of the current instance will be reset and the randomly generated password will be securely stored in Parameter Store as `/EC2Rescue/Password/<INSTANCE_ID>`. If you select this action and provide no parameters, passwords are encrypted automatically with the default KMS key. Optionally, you can specify a KMS Key ID in Parameters to encrypt the password with your own key.
 - **CollectLogs**—Runs EC2Rescue for Windows Server with the `/collect:all` action. If you select this action, Parameters must include an Amazon S3 bucket name to upload the logs to.
 - **FixAll**—Runs EC2Rescue for Windows Server with the `/rescue:all` action. If you select this action, Parameters must include the block device name to rescue.
- **Parameters**—The PowerShell parameters to pass for the specified command.

Note

In order for the **ResetAccess** action to work, your Amazon EC2 instance needs to have the following policy attached in order to write the encrypted password to Parameter Store. Please wait a few minutes before attempting to reset the password of an instance after you have attached this policy to the related IAM role.

Using the default KMS key:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {
```

```
        "Effect": "Allow",
        "Action": [
            "ssm:PutParameter"
        ],
        "Resource": [
            "arn:aws:ssm:region:account_id:parameter/EC2Rescue/Passwords/<instanceid>"
        ]
    }
}
```

Using a custom KMS key:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": [
                "arn:aws:ssm:region:account_id:parameter/EC2Rescue/Passwords/<instanceid>"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt"
            ],
            "Resource": [
                "arn:aws:kms:region:account_id:key/<kmskeyid>"
            ]
        }
    ]
}
```

The following procedure describes how to view the JSON for this document in the Amazon EC2 console.

To view the JSON for the Systems Manager Run Command document

1. Open the SSM console at <https://console.aws.amazon.com/systems-manager/home>.
2. In the navigation pane, expand **Shared Services** and choose **Documents**.
3. In the search bar, set **Owner** as **Owned by Me or Amazon** and set the **Document name prefix** to **AWSSupport-RunEC2RescueForWindowsTool**.
4. Select the **AWSSupport-RunEC2RescueForWindowsTool** document, choose **Contents**, and then view the JSON.

Examples

Here are some examples on how to use the Systems Manager Run Command document to execute EC2Rescue for Windows Server, using the AWS CLI. For more information about sending commands with the AWS CLI, see the [AWS CLI Command Reference](#).

Attempt to Fix All Identified Issues on an Offline Root Volume

Attempt to fix all identified issues on an offline root volume attached to an Amazon EC2 Windows instance:

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-RunEC2RescueForWindowsTool" --comment "EC2Rescue offline volume xvdf" --parameters "Command=FixAll, Parameters='xvdf'" --output text
```

Collect Logs from the Current Amazon EC2 Windows Instance

Collect all logs from the current online Amazon EC2 Windows instance and upload them to an Amazon S3 bucket:

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-RunEC2RescueForWindowsTool" --comment "EC2Rescue online log collection to S3" --parameters "Command=CollectLogs, Parameters='YOURS3BUCKETNAME'" --output text
```

Collect Logs from an Offline Amazon EC2 Windows Instance Volume

Collect all logs from an offline volume attached to an Amazon EC2 Windows instance and upload them to Amazon S3 with a presigned URL:

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-RunEC2RescueForWindowsTool" --comment "EC2Rescue offline log collection to S3" --parameters "Command=CollectLogs, Parameters='\"-Offline -BlockDeviceName xvdf -S3PreSignedUrl YOURS3PRESIGNEDURL\"'" --output text
```

Reset the Local Administrator Password

The following examples show methods you can use to reset the local Administrator password. The output provides a link to Parameter Store, where you can find the randomly generated secure password you can then use to RDP to your Amazon EC2 Windows instance as the local Administrator.

Reset the local Administrator password of an online instance using the default KMS key alias/aws/ssm:

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-RunEC2RescueForWindowsTool" --comment "EC2Rescue online password reset" --parameters "Command=ResetAccess" --output text
```

Reset the local Administrator password of an online instance using a KMS key:

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-RunEC2RescueForWindowsTool" --comment "EC2Rescue online password reset" --parameters "Command=ResetAccess, Parameters=a133dc3c-a2g4-4fc6-a873-6c0720104bf0" --output text
```

Note

In this example, the KMS key is a133dc3c-a2g4-4fc6-a873-6c0720104bf0.

Troubleshooting Windows Instances

The following procedures and tips can help you troubleshoot problems with your Amazon EC2 Windows instances.

Contents

- [Troubleshooting Instance Launch Issues \(p. 965\)](#)
- [Troubleshoot an Unreachable Instance \(p. 967\)](#)
- [Resetting a Lost or Expired Windows Administrator Password \(p. 974\)](#)
- [Common Issues \(p. 982\)](#)
- [High CPU usage shortly after Windows starts \(p. 984\)](#)
- [No console output \(p. 985\)](#)
- [Remote Desktop can't connect to the remote computer \(p. 986\)](#)
- [Error Using macOS RDP Client \(p. 988\)](#)
- [RDP displays a black screen instead of the desktop \(p. 988\)](#)
- [Instance loses network connectivity or scheduled tasks don't run when expected \(p. 989\)](#)
- [Windows Server 2012 R2 not available on the network \(p. 989\)](#)
- [Troubleshooting Remote Desktop Connection issues using AWS Systems Manager \(p. 989\)](#)
- [Common Messages \(p. 992\)](#)

If you need additional help, you can post a question to the [Amazon EC2 forum](#). Be sure to post the ID of your instance and any error messages, including error messages available through console output.

To get additional information for troubleshooting problems with your instance, use [Using EC2Rescue for Windows Server \(p. 954\)](#). For information about troubleshooting issues with PV drivers, see [Troubleshooting PV Drivers \(p. 419\)](#).

Troubleshooting Instance Launch Issues

The following issues prevent you from launching an instance.

Launch Issues

- [Instance Limit Exceeded \(p. 965\)](#)
- [Insufficient Instance Capacity \(p. 966\)](#)
- [Instance Terminates Immediately \(p. 966\)](#)

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 \(p. 899\)](#).

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 \(p. 182\)](#).
- 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 \(p. 693\)](#).
- 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 \(p. 853\)](#). 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 \(p. 899\)](#).
- 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](#).

Troubleshoot 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 100kb.

- [How to Take a Screenshot of an Unreachable Instance \(p. 968\)](#)
- [Common Screenshots \(p. 968\)](#)

How to Take a Screenshot of an Unreachable Instance

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 output 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)

For API calls, the returned content is base64-encoded. For command line tools, the decoding is performed for you.

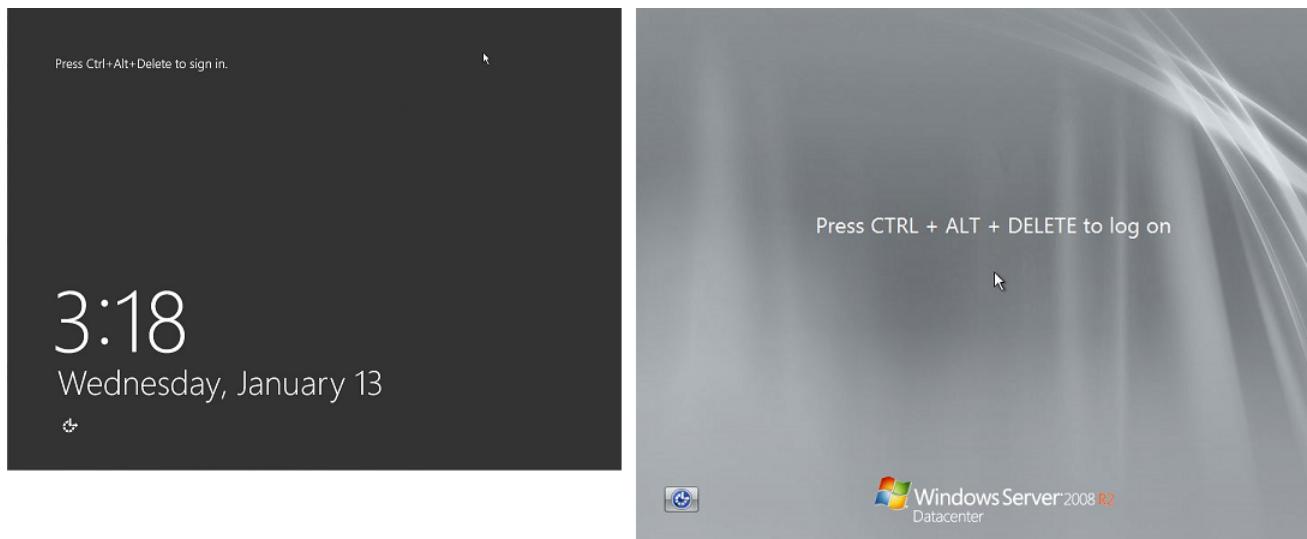
Common Screenshots

You can use the following information to help you troubleshoot an unreachable instance based on screenshots returned by the service.

- [Log On Screen \(Ctrl+Alt+Delete\) \(p. 968\)](#)
- [Recovery Console Screen \(p. 971\)](#)
- [Windows Boot Manager Screen \(p. 972\)](#)
- [Sysprep Screen \(p. 972\)](#)
- [Getting Ready Screen \(p. 973\)](#)
- [Windows Update Screen \(p. 974\)](#)
- [Chkdsk \(p. 974\)](#)

Log On Screen (Ctrl+Alt+Delete)

Console Screenshot Service returned the following.



If an instance becomes unreachable during log on, there could be a problem with your network configuration or Windows Remote Desktop Services. An instance can also be unresponsive if a process is using large amounts of CPU.

Network Configuration

Use the following information, to verify that your AWS, Microsoft Windows, and local (or on-premises) network configurations aren't blocking access to the instance.

AWS Network Configuration

Configuration	Verify
Security group configuration	Verify that port 3389 is open for your security group. Verify you are connecting to the right public IP address. If the instance was not associated with an Elastic IP, the public IP changes after the instance stops/starts. For more information, see Remote Desktop can't connect to the remote computer (p. 986) .
VPC configuration (Network ACLs)	Verify that the access control list (ACL) for your Amazon VPC is not blocking access. For information, see Network ACLs in the <i>Amazon VPC User Guide</i> .
VPN configuration	If you are connecting to your VPC using a virtual private network (VPN), verify VPN tunnel connectivity. For more information, see How do I troubleshoot VPN tunnel connectivity to an Amazon VPC?

Windows Network Configuration

Configuration	Verify
Windows Firewall	Verify that Windows Firewall isn't blocking connections to your instance. Disable Windows

Configuration	Verify
	Firewall as described in bullet 7 of the remote desktop troubleshooting section, Remote Desktop can't connect to the remote computer (p. 986) .
Advanced TCP/IP configuration (Use of static IP)	The instance may be unresponsive because you configured a static IP address. For a VPC, Create a network interface (p. 675) and attach it to the instance (p. 678) . For EC2 Classic, enable DHCP.

Local or On-Premises Network Configuration

Verify that a local network configuration isn't blocking access. Try to connect to another instance in the same VPC as your unreachable instance. If you can't access another instance, work with your local network administrator to determine whether a local policy is restricting access.

Remote Desktop Services Issue

If the instance can't be reached during log on, there could a problem with Remote Desktop Services (RDS) on the instance.

Remote Desktop Services Configuration

Configuration	Verify
RDS is running	Verify that RDS is running on the instance. Connect to the instance using the Microsoft Management Console (MMC) Services snap-in (<code>services.msc</code>). In the list of services, verify that Remote Desktop Services is Running . If it isn't, start it and then set the startup type to Automatic . If you can't connect to the instance by using the Services snap-in, detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, attach the original volume to another instance in the same availability zone as a secondary volume, and modify the Start registry key. When you are finished, reattach the root volume to the original instance. For more information about detaching volumes, see Detaching an Amazon EBS Volume from an Instance (p. 771) .
RDS is enabled	Even if the service is started, it might be disabled. Detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, attach the original volume to another instance in the same Availability Zone as a secondary volume, and enable the service by modifying the Terminal Server registry key as described in the following article: Enable Remote Desktop Remotely By Using Remote Registry . When you are finished, reattach the root volume to the original instance. For more information, see Detaching an Amazon EBS Volume from an Instance (p. 771) .

High CPU

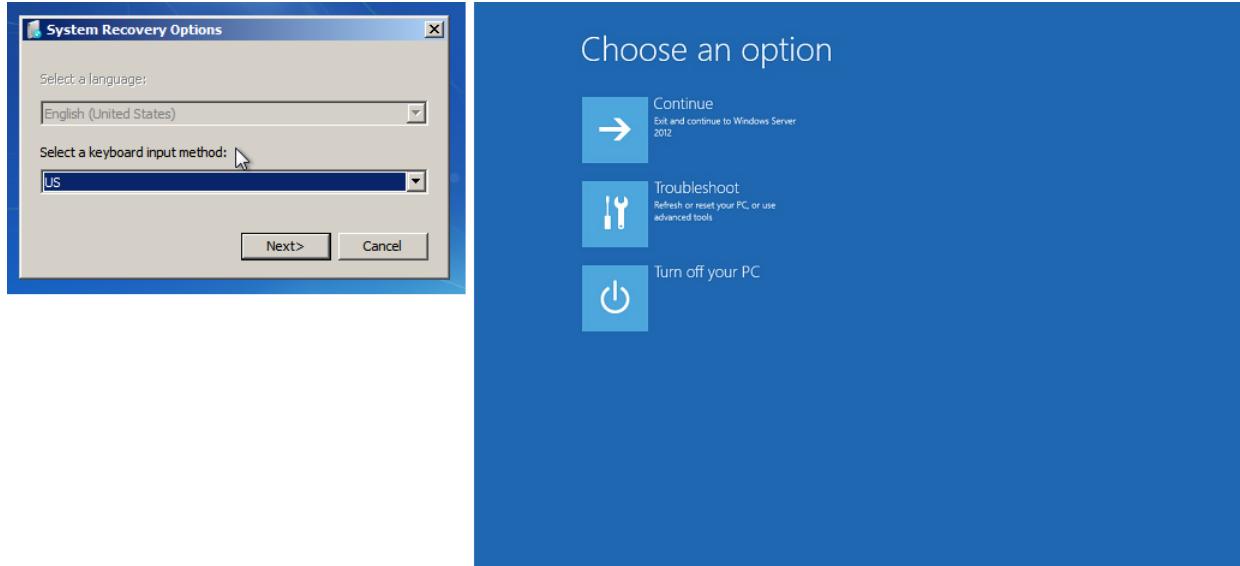
Check the **CPUUtilization (Maximum)** metric on your instance by using Amazon CloudWatch. If **CPUUtilization (Maximum)** is a high number, wait for the CPU to go down and try connecting again. High CPU usage can be caused by:

- Windows Update
- Security Software Scan
- Custom Startup Script
- Task Scheduler

For more information about the **CPUUtilization (Maximum)** metric, see [Get Statistics for a Specific EC2 Instance](#) in the *Amazon CloudWatch User Guide*. For additional troubleshooting tips, see [High CPU usage shortly after Windows starts \(p. 984\)](#).

Recovery Console Screen

Console Screenshot Service returned the following.



The operating system may boot into the Recovery console and get stuck in this state if the `bootstatuspolicy` is not set to `ignoreallfailures`. Use the following procedure to change the `bootstatuspolicy` configuration to `ignoreallfailures`.

By default, the policy configuration for AWS-provided public Windows AMIs is set to `ignoreallfailures`.

1. Stop the unreachable instance.
2. Create a snapshot of the root volume. The root volume is attached to the instance as `/dev/sda1`.

Detach the root volume from the unreachable instance, take a snapshot of the volume or create an AMI from it, and attach it to another instance in the same Availability Zone as a secondary volume. For more information, see [Detaching an Amazon EBS Volume from an Instance \(p. 771\)](#).

Warning

If your temporary instance is based on the same AMI that the original instance is based on, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses an AMI for Windows Server 2008 R2, launch the temporary instance using an AMI for Windows Server 2012. If you must create a temporary instance based on the same AMI, see Step 6 in [Remote Desktop can't connect to the remote computer \(p. 986\)](#) to avoid a disk signature collision.

3. Log in to the instance and execute the following command from a command prompt to change the `bootstatuspolicy` configuration to `ignoreallfailures`:

```
bcdeedit /store Drive Letter:\boot\bcd /set {default} bootstatuspolicy ignoreallfailures
```

4. Reattach the volume to the unreachable instance and start the instance again.

Windows Boot Manager Screen

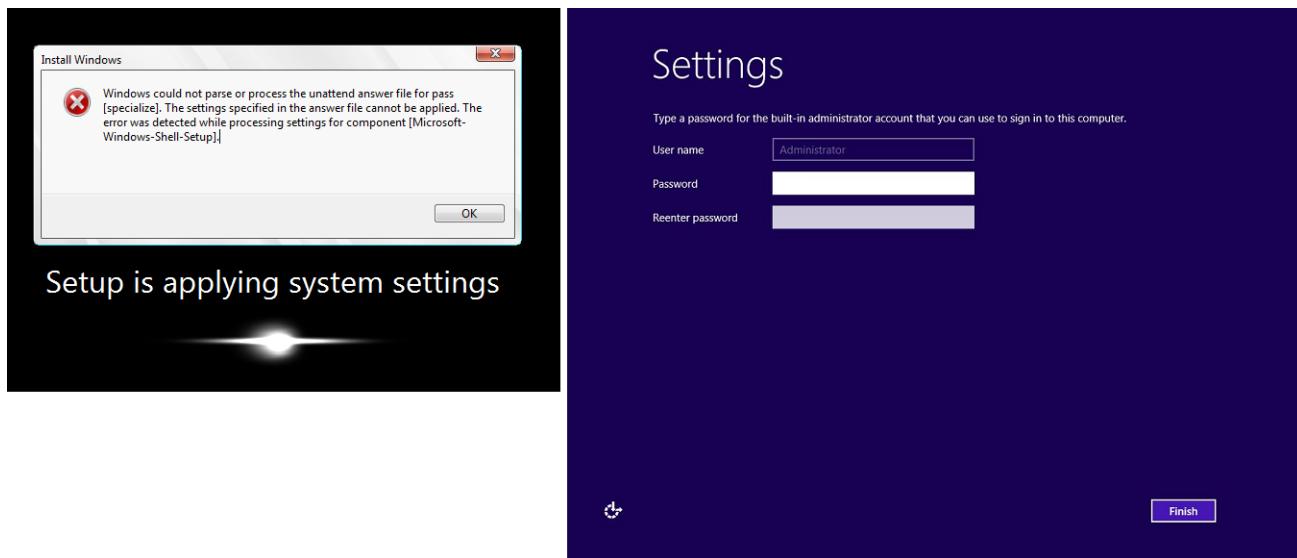
Console Screenshot Service returned the following.



The operating system experienced a fatal corruption in the system file and/or the registry. When the instance is stuck in this state, you should recover the instance from a recent backup AMI or launch a replacement instance. If you need to access data on the instance, detach any root volumes from the unreachable instance, take a snapshot of those volume or create an AMI from them, and attach them to another instance in the same Availability Zone as a secondary volume. For more information, see [Detaching an Amazon EBS Volume from an Instance \(p. 771\)](#).

Sysprep Screen

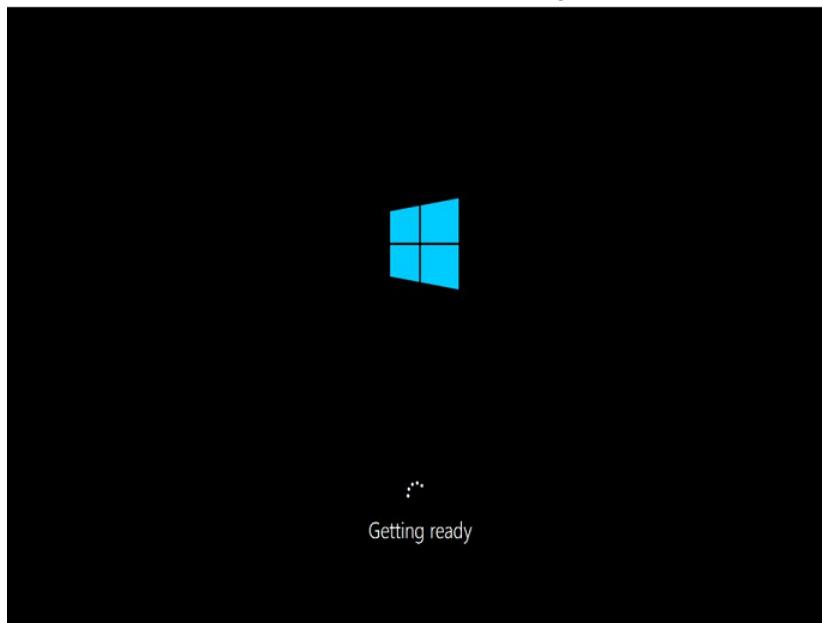
Console Screenshot Service returned the following.



You may see this screen if you did not use the EC2Config Service to call sysprep.exe or if the operating system failed while running Sysprep. To solve this problem, [Create a Standard Amazon Machine Image Using Sysprep \(p. 104\)](#).

Getting Ready Screen

Console Screenshot Service returned the following.

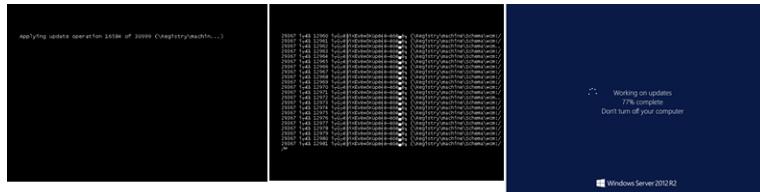


Refresh the Instance Console Screenshot Service repeatedly to verify that the progress ring is spinning. If the ring is spinning, wait for the operating system to start up. You can also check the **CPUUtilization (Maximum)** metric on your instance by using Amazon CloudWatch to see if the operating system is active. If the progress ring is not spinning, the instance may be stuck at the boot process. Reboot the instance. If rebooting does not solve the problem, recover the instance from a recent backup AMI or launch a replacement instance. If you need to access data on the instance, detach the root volume from the unreachable instance, take a snapshot of the volume or create an AMI from it. Then attach it to another instance in the same Availability Zone as a secondary volume. For more information about

CPUUtilization (Maximum), see [Get Statistics for a Specific EC2 Instance](#) in the *Amazon CloudWatch User Guide*.

Windows Update Screen

Console Screenshot Service returned the following.



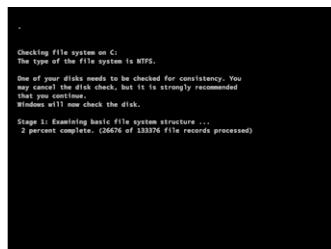
The Windows Update process is updating the registry. Wait for the update to finish. Do not reboot or stop the instance as this may cause data corruption during the update.

Note

The Windows Update process can consume resources on the server during the update. If you experience this problem often, consider using faster instance types and faster EBS volumes.

Chkdsk

Console Screenshot Service returned the following.



Windows is running the chkdsk system tool on the drive to verify file system integrity and fix logical file system errors. Wait for process to complete.

Resetting a Lost or Expired Windows Administrator Password

If you are no longer able to access your Windows Amazon EC2 instance because the Windows administrator password is lost or expired, you can reset the password.

Note

There is an AWS Systems Manager Automation document that automatically applies the manual steps necessary to reset the local administrator password. For more information, see [Reset the Local Administrator Password on Amazon EC2 Windows Instances](#) in the *AWS Systems Manager User Guide*.

The manual methods to reset the administrator password use either EC2Config or EC2Launch.

- For Windows AMIs before Windows Server 2016, use the EC2Config service.
- For Windows Server 2016 AMIs, use the EC2Launch service.

These procedures also describe how to connect to an instance if you lost the key pair that was used to create the instance. Amazon EC2 uses a public key to encrypt a piece of data, such as a password, and

a private key to decrypt the data. The public and private keys are known as a *key pair*. With Windows instances, you use a key pair to obtain the administrator password and then log in using RDP.

Note

If you have disabled the local administrator account on the instance and your instance is configured for Systems Manager, you can also re-enable and reset your local administrator password by using EC2Rescue and Run Command. For more information, see [Using EC2Rescue for Windows Server with Systems Manager Run Command](#).

Contents

- [Resetting the Windows Administrator Password Using EC2Config \(p. 975\)](#)
- [Resetting the Windows Administrator Password Using EC2Launch \(p. 979\)](#)

Resetting the Windows Administrator Password Using EC2Config

If you have lost your Windows administrator password and are using a Windows AMI before Windows Server 2016, you can use the EC2Config service to generate a new password.

If you are using a Windows Server 2016 AMI, see [Resetting the Windows Administrator Password Using EC2Launch \(p. 979\)](#).

Note

If you have disabled the local administrator account on the instance and your instance is configured for Systems Manager, you can also re-enable and reset your local administrator password by using EC2Rescue and Run Command. For more information, see [Using EC2Rescue for Windows Server with Systems Manager Run Command](#).

Note

There is an AWS Systems Manager Automation document that automatically applies the manual steps necessary to reset the local administrator password. For more information, see [Reset the Local Administrator Password on Amazon EC2 Windows Instances](#) in the *AWS Systems Manager User Guide*.

To reset your Windows administrator password using EC2Config, you need to do the following:

- [Step 1: Verify that the EC2Config service is running \(p. 975\)](#)
- [Step 2: Detach the root volume from the instance \(p. 976\)](#)
- [Step 3: Attach the volume to a temporary instance \(p. 976\)](#)
- [Step 4: Modify the configuration file \(p. 977\)](#)
- [Step 5: Restart the original instance \(p. 978\)](#)

Step 1: Verify that the EC2Config service is running

Before you attempt to reset the administrator password, verify that the EC2Config service is installed and running. You use the EC2Config service to reset the administrator password later in this section.

To verify that the EC2Config service is running

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances** and then choose the instance that needs a password reset. This instance is referred to as the *original* instance in this procedure.
3. Choose **Actions, Instance Settings, Get System Log**.

4. Locate the EC2 Agent entry, for example, **EC2 Agent: Ec2Config service v3.18.1118**. If you see this entry, the EC2Config service is running.

If the system log output is empty, or if the EC2Config service is not running, troubleshoot the instance using the Instance Console Screenshot service. For more information, see [Troubleshoot an Unreachable Instance \(p. 967\)](#).

Step 2: Detach the root volume from the instance

You can't use EC2Config to reset an administrator password if the volume on which the password is stored is attached to an instance as the root volume. You must detach the volume from the original instance before you can attach it to a temporary instance as a secondary volume.

To detach the root volume from the 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 that needs a password reset and choose **Actions, Instance State, Stop**. After the instance's status changes to **stopped**, continue with the next step.
4. (Optional) If you have the private key that you specified when you launched this instance, continue with the next step. Otherwise, use the following steps to replace the instance with a new instance that you launch with a new key pair.
 - a. Create a new key pair using the Amazon EC2 console. To give your new key pair the same name as the one for which you lost the private key, you must first delete the existing key pair.
 - b. Select the instance to replace. Note the instance type, VPC, subnet, security group, and IAM role of the instance.
 - c. Choose **Actions, Image, Create Image**. Type a name and a description for the image and choose **Create Image, View pending image**. After the image status changes to **available**, continue to the next step.
 - d. Select the image and choose **Launch**.
 - e. Complete the wizard, selecting the same instance type, VPC, subnet, security group, and IAM role as the instance to replace, and then choose **Launch**.
 - f. When prompted, choose the key pair that you created for the new instance, select the acknowledgement check box, and then choose **Launch Instances**.
 - g. (Optional) If the original instance has an associated Elastic IP address, transfer it to the new instance. If the original instance has EBS volumes in addition to the root volume, transfer them to the new instance.
 - h. Terminate the stopped instance, as it is no longer needed. For the remainder of this procedure, all references to the original instance apply to this instance that you just created.
5. Detach the root volume from the original instance as follows:
 - a. In the **Description** pane of the original instance, note the ID of the EBS volume listed as the **Root device**.
 - b. In the navigation pane, choose **Volumes**.
 - c. In the list of volumes, select the volume noted in the previous step, and choose **Actions, Detach Volume**. After the volume status changes to **available**, continue with the next step.

Step 3: Attach the volume to a temporary instance

Next, launch a temporary instance and attach the volume to it as a secondary volume. This is the instance you use to modify the configuration file.

To launch a temporary instance and attach the volume

1. Launch the temporary instance as follows:
 - a. In the navigation pane, choose **Instances**, choose **Launch Instance**, and then select an AMI.

Important
To avoid disk signature collisions, you must select an AMI for a different version of Windows. For example, if the original instance runs Windows Server 2012 R2, launch the temporary instance using the base AMI for Windows Server 2008 R2.
 - b. Leave the default instance type and choose **Next: Configure Instance Details**.
 - c. On the **Configure Instance Details** page, for **Subnet**, select the same Availability Zone as the original instance and choose **Review and Launch**.

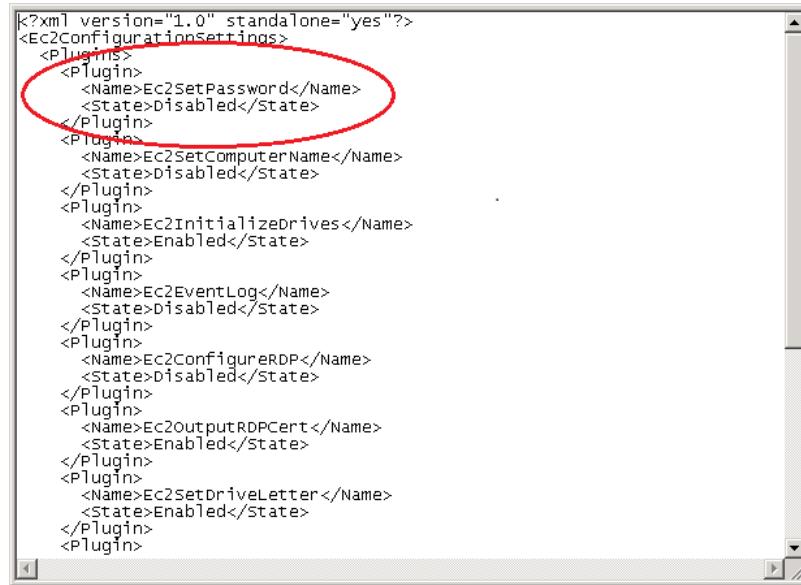
Important
The temporary instance must be in the same Availability Zone as the original instance. If your temporary instance is in a different Availability Zone, you can't attach the original instance's root volume to it.
 - d. On the **Review Instance Launch** page, choose **Launch**.
 - e. When prompted, create a new key pair, download it to a safe location on your computer, and then choose **Launch Instances**.
2. Attach the volume to the temporary instance as a secondary volume as follows:
 - a. In the navigation pane, choose **Volumes**, select the volume that you detached from the original instance, and then choose **Actions, Attach Volume**.
 - b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your temporary instance and select the instance from the list.
 - c. For **Device**, type **xvdf** (if it isn't already there), and choose **Attach**.

Step 4: Modify the configuration file

After you have attached the volume to the temporary instance as a secondary volume, modify the `Ec2SetPassword` plugin in the configuration file.

To modify the configuration file

1. From the temporary instance, modify the configuration file on the secondary volume as follows:
 - a. Launch and connect to the temporary instance.
 - b. Open the **Disk Management** utility, and bring the drive online using these instructions: [Making an Amazon EBS Volume Available for Use](#).
 - c. Navigate to the secondary volume, and open `\Program Files\Amazon\Ec2ConfigService\Settings\config.xml` using a text editor, such as Notepad.
 - d. At the top of the file, find the plugin with the name `Ec2SetPassword`, as shown in the screenshot. Change the state from `Disabled` to `Enabled` and save the file.



2. After you have modified the configuration file, detach the secondary volume from the temporary instance as follows:
 - a. Using the **Disk Management** utility, bring the volume offline.
 - b. Disconnect from the temporary instance and return to the Amazon EC2 console.
 - c. In the navigation pane, choose **Volumes**, select the volume, and then choose **Actions, Detach Volume**. After the volume's status changes to **available**, continue with the next step.

Step 5: Restart the original instance

After you have modified the configuration file, reattach the volume to the original instance as the root volume and connect to the instance using its key pair to retrieve the administrator password.

1. Reattach the volume to the original instance as follows:
 - a. In the navigation pane, choose **Volumes**, select the volume that you detached from the temporary instance, and then choose **Actions, Attach Volume**.
 - b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your original instance and then select the instance.
 - c. For **Device**, type **/dev/sda1**.
 - d. Choose **Attach**. After the volume status changes to **in-use**, continue to the next step.
2. In the navigation pane, choose **Instances**. Select the original instance and choose **Actions, Instance State, Start**. When prompted for confirmation, choose **Yes, Start**. After the instance state changes to **running**, continue to the next step.
3. Retrieve your new Windows administrator password using the private key for the new key pair and connect to the instance. For more information, see [Connecting to Your Windows Instance \(p. 359\)](#).

Important

The instance gets a new public IP address after you stop and start it. Make sure to connect to the instance using its current public DNS name. For more information, see [Instance Lifecycle \(p. 309\)](#).

4. (Optional) If you have no further use for the temporary instance, you can terminate it. Select the temporary instance, and choose **Actions, Instance State, Terminate**.

Resetting the Windows Administrator Password Using EC2Launch

If you have lost your Windows administrator password and are using a Windows Server 2016 AMI, you can use the EC2Rescue tool, which uses the EC2Launch service to generate a new password.

If you are using a Windows Server AMI earlier than Windows Server 2016, see [Resetting the Windows Administrator Password Using EC2Config \(p. 975\)](#).

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

Note

If you have disabled the local administrator account on the instance and your instance is configured for Systems Manager, you can also re-enable and reset your local administrator password by using EC2Rescue and Run Command. For more information, see [Using EC2Rescue for Windows Server with Systems Manager Run Command](#).

Note

There is an AWS Systems Manager Automation document that automatically applies the manual steps necessary to reset the local administrator password. For more information, see [Reset the Local Administrator Password on Amazon EC2 Windows Instances](#) in the *AWS Systems Manager User Guide*.

To reset your Windows administrator password using EC2Launch, you need to do the following:

- [Step 1: Detach the root volume from the instance \(p. 979\)](#)
- [Step 2: Attach the volume to a temporary instance \(p. 980\)](#)
- [Step 3: Reset the administrator password \(p. 981\)](#)
- [Step 4: Restart the original instance \(p. 981\)](#)

Step 1: Detach the root volume from the instance

You can't use EC2Launch to reset an administrator password if the volume on which the password is stored is attached to an instance as the root volume. You must detach the volume from the original instance before you can attach it to a temporary instance as a secondary volume.

To detach the root volume from the 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 that needs a password reset and choose **Actions**, **Instance State**, **Stop**. After the instance's status changes to **stopped**, continue with the next step.
4. (Optional) If you have the private key that you specified when you launched this instance, continue with the next step. Otherwise, use the following steps to replace the instance with a new instance that you launch with a new key pair.
 - a. Create a new key pair using the Amazon EC2 console. To give your new key pair the same name as the one for which you lost the private key, you must first delete the existing key pair.
 - b. Select the instance to replace. Note the instance type, VPC, subnet, security group, and IAM role of the instance.
 - c. Choose **Actions**, **Image**, **Create Image**. Type a name and a description for the image and choose **Create Image**, **View pending image**. After the image status changes to **available**, continue to the next step.

- d. Select the image and choose **Launch**.
 - e. Complete the wizard, selecting the same instance type, VPC, subnet, security group, and IAM role as the instance to replace, and then choose **Launch**.
 - f. When prompted, choose the key pair that you created for the new instance, select the acknowledgement check box, and then choose **Launch Instances**.
 - g. (Optional) If the original instance has an associated Elastic IP address, transfer it to the new instance. If the original instance has EBS volumes in addition to the root volume, transfer them to the new instance.
 - h. Terminate the stopped instance, as it is no longer needed. For the remainder of this procedure, all references to the original instance apply to this instance that you just created.
5. Detach the root volume from the original instance as follows:
 - a. In the **Description** pane of the original instance, note the ID of the EBS volume listed as the **Root device**.
 - b. In the navigation pane, choose **Volumes**.
 - c. In the list of volumes, select the volume noted in the previous step, and choose **Actions, Detach Volume**. After the volume status changes to **available**, continue with the next step.

Step 2: Attach the volume to a temporary instance

Next, launch a temporary instance and attach the volume to it as a secondary volume. This is the instance you use to run EC2Launch.

To launch a temporary instance and attach the volume

1. Launch the temporary instance as follows:
 - a. In the navigation pane, choose **Instances**, choose **Launch Instance**, and then select an AMI.

Important
To avoid disk signature collisions, you must select an AMI for a different version of Windows. For example, if the original instance runs Windows Server 2012 R2, launch the temporary instance using the base AMI for Windows Server 2008 R2.
 - b. Leave the default instance type and choose **Next: Configure Instance Details**.
 - c. On the **Configure Instance Details** page, for **Subnet**, select the same Availability Zone as the original instance and choose **Review and Launch**.

Important
The temporary instance must be in the same Availability Zone as the original instance. If your temporary instance is in a different Availability Zone, you can't attach the original instance's root volume to it.
 - d. On the **Review Instance Launch** page, choose **Launch**.
 - e. When prompted, create a new key pair, download it to a safe location on your computer, and then choose **Launch Instances**.
2. Attach the volume to the temporary instance as a secondary volume as follows:
 - a. In the navigation pane, choose **Volumes**, select the volume that you detached from the original instance, and then choose **Actions, Attach Volume**.
 - b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your temporary instance and select the instance from the list.
 - c. For **Device**, type **xvdf** (if it isn't already there), and choose **Attach**.

Step 3: Reset the administrator password

Next, connect to the temporary instance and use EC2Launch to reset the administrator password.

To reset the administrator password

1. Connect to the temporary instance and use the EC2Rescue for Windows Server tool on the instance to reset the administrator password as follows:
 - a. Download the [EC2Rescue for Windows Server](#) zip file, extract the contents, and run **EC2Rescue.exe**.
 - b. On the **License Agreement** screen, read the license agreement, and, if you accept the terms, choose **I Agree**.
 - c. On the **Welcome to EC2Rescue for Windows Server** screen, choose **Next**.
 - d. On the **Select mode** screen, choose **Offline instance**.
 - e. On the **Select a disk** screen, select the **xvdf** device and choose **Next**.
 - f. Confirm the disk selection and choose **Yes**.
 - g. After the volume has loaded, choose **OK**.
 - h. On the **Select Offline Instance Option** screen, choose **Diagnose and Rescue**.
 - i. On the **Summary** screen, review the information and choose **Next**.
 - j. On the **Detected possible issues** screen, select **Reset Administrator Password** and choose **Next**.
 - k. On the **Confirm** screen, choose **Rescue, OK**.
 - l. On the **Done** screen, choose **Finish**.
 - m. Close the EC2Rescue for Windows Server tool, disconnect from the temporary instance, and then return to the Amazon EC2 console.
2. Detach the secondary (**xvd£**) volume from the temporary instance as follows:
 - a. In the navigation pane, choose **Instances** and select the temporary instance.
 - b. In the **Description** pane of the temporary instance, note the ID of the EBS volume listed as **xvdf**.
 - c. In the navigation pane, choose **Volumes**.
 - d. In the list of volumes, select the volume noted in the previous step, and choose **Actions, Detach Volume**. After the volume status changes to **available**, continue with the next step.

Step 4: Restart the original instance

After you have reset the administrator password using EC2Launch, reattach the volume to the original instance as the root volume and connect to the instance using its key pair to retrieve the administrator password.

To restart the original instance

1. Reattach the volume to the original instance as follows:
 - a. In the navigation pane, choose **Volumes**, select the volume that you detached from the temporary instance, and then choose **Actions, Attach Volume**.
 - b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your original instance and then select the instance.
 - c. For **Device**, type **/dev/sda1**.
 - d. Choose **Attach**. After the volume status changes to **in-use**, continue to the next step.
2. In the navigation pane, choose **Instances**. Select the original instance and choose **Actions, Instance State, Start**. When prompted for confirmation, choose **Yes, Start**. After the instance state changes to **running**, continue to the next step.

3. Retrieve your new Windows administrator password using the private key for the new key pair and connect to the instance. For more information, see [Connecting to Your Windows Instance \(p. 359\)](#).
4. (Optional) If you have no further use for the temporary instance, you can terminate it. Select the temporary instance, and choose **Actions, Instance State, Terminate**.

Common Issues

The following are troubleshooting tips to help you solve common issues with EC2 instance running Windows Server.

Issues

- [EBS volumes don't initialize on Windows Server 2016 AMIs \(p. 982\)](#)
- [Boot an EC2 Windows Instance into Directory Services Restore Mode \(DSRM\) \(p. 983\)](#)
- [Unable to remotely log on to an instance with a user account that is not an Administrator \(p. 984\)](#)

EBS volumes don't initialize on Windows Server 2016 AMIs

Instances created from Windows Server 2016 Amazon Machine Images (AMIs) use the EC2Launch service for a variety of startup tasks, including initializing EBS volumes. By default, EC2Launch does not initialize secondary volumes. You can configure EC2Launch to initialize these disks automatically.

To map drive letters to volumes

1. Connect to the instance to configure and open the C:\ProgramData\Amazon\EC2-Windows\Launch\Config\DriveLetterMappingConfig.json file in a text editor.
2. Specify the volume settings using the following format:

```
{  
    "driveLetterMapping": [  
        {  
            "volumeName": "sample volume",  
            "driveLetter": "H"  
        }  
    ]  
}
```

3. Save your changes and close the file.
4. Open Windows PowerShell and use the following command to run the EC2Launch script that initializes the disks:

```
PS C:> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
```

To initialize the disks each time the instance boots, add the **-Schedule** flag as follows:

```
PS C:> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1 -Schedule
```

Boot an EC2 Windows Instance into Directory Services Restore Mode (DSRM)

If an instance running Microsoft Active Directory experiences a system failure or other critical issues you can troubleshoot the instance by booting into a special version of Safe Mode called *Directory Services Restore Mode* (DSRM). In DSRM you can repair or recover Active Directory.

Driver Support for DSRM

How you enable DSRM and boot into the instance depends on the drivers the instance is running. In the EC2 console you can view driver version details for an instance from the System Log. The following tables shows which drivers are supported for DSRM.

Driver Versions	DSRM Supported?	Next Steps
Citrix PV 5.9	No	Restore the instance from a backup. You cannot enable DSRM.
AWS PV 7.2.0	No	Though DSRM is not supported for this driver, you can still detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, and attach it to another instance in the same availability zone as a secondary volume. You can then enable DSRM (as described in this section).
AWS PV 7.2.2 and later	Yes	Detach the root volume, attach it to another instance, and enable DSRM (as described in this section).
Enhanced Networking	Yes	Detach the root volume, attach it to another instance, and enable DSRM (as described in this section).

For information about how to enable Enhanced Networking, see [Enabling Enhanced Networking on Windows Instances in a VPC](#). For more information about upgrading AWS PV drivers, see [Upgrading PV Drivers on Your Windows Instances \(p. 414\)](#).

Configure an Instance to Boot into DSRM

EC2 Windows instances do not have network connectivity before the operating system is running. For this reason, you cannot press the F8 button on your keyboard to select a boot option. You must use one of the following procedures to boot an EC2 Windows Server instance into DSRM.

If you suspect that Active Directory has been corrupted and the instance is still running, you can configure the instance to boot into DSRM using either the System Configuration dialog box or the command prompt.

To boot an online instance into DSRM using the System Configuration dialog box

1. In the **Run** dialog box, type `msconfig` and press Enter.
2. Choose the **Boot** tab.
3. Under **Boot options** choose **Safe boot**.
4. Choose **Active Directory repair** and then choose **OK**. The system prompts you to reboot the server.

To boot an online instance into DSRM using the command line

From a Command Prompt window, run the following command:

```
bcdeedit /set safeboot dsrepair
```

If an instance is offline and unreachable, you must detach the root volume and attach it to another instance to enable DSRM mode.

To boot an offline instance into DSRM

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. Open the context (right-click) menu for the instance, choose **InstanceState**, and then choose **Stop**.
4. Choose **Launch Instance** and create a temporary instance in the same Availability Zone as the affected instance. Choose an instance type that uses a different version of Windows. For example, if your instance is Windows Server 2008 R1, then choose a Windows Server 2008 R2 instance.

Important

If you do not create the instance in the same Availability Zone as the affected instance you will not be able to attach the root volume of the affected instance to the new instance.

5. In the navigation pane, choose **Volumes**.
6. Locate the root volume of the affected instance. [Detach](#) the volume and [attach](#) it to the temporary instance you created earlier. Attach it with the default device name (xvdf).
7. Use Remote Desktop to connect to the temporary instance, and then use the Disk Management utility to [make the volume available for use](#).
8. Open a command prompt and run the following command. Replace *D* with the actual drive letter of the secondary volume you just attached:

```
bcdeedit /store D:\Boot\BCD /set {default} safeboot dsrepair
```

9. In the Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.
10. In the EC2 console, detach the affected volume from the temporary instance and reattach it to your original instance with the device name /dev/sda1. You must specify this device name to designate the volume as a root volume.
11. [Start](#) the instance.
12. After the instance passes the health checks in the EC2 console, connect to the instance using Remote Desktop and verify that it boots into DSRM mode.
13. (Optional) Delete or stop the temporary instance you created in this procedure.

Unable to remotely log on to an instance with a user account that is not an Administrator

If you are not able to remotely log on to a Windows Server EC2 instance from a user account that is not an administrator account, ensure that you have granted the user the right to log on locally. See [Grant a user or group the right to log on locally to the domain controllers in the domain](#).

High CPU usage shortly after Windows starts

If Windows Update is set to **Check for updates but let me choose whether to download and install them** (the default instance setting) this check can consume anywhere from 50 - 99% of the CPU on

the instance. If this CPU consumption causes problems for your applications, you can manually change Windows Update settings in **Control Panel** or you can use the following script in the Amazon EC2 user data field:

```
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update" /v  
AUOptions /t REG_DWORD /d 3 /f net stop wuauserv net start wuauserv
```

When you execute this script specify a value for /d. The default value is 3. Possible values include the following:

- Never check for updates
- Check for updates but let me choose whether to download and install them
- Download updates but let me choose whether to install them
- Install updates automatically

To modify the user data for a Amazon EBS-backed 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.
3. Choose **Actions**, select **Instance State**, and then choose **Stop**.
4. In the confirmation dialog box, select **Yes, Stop**. It can take a few minutes for the instance to stop.
5. With the instance still selected, select **Actions**, select **Instance Settings**, and then choose **View/Change User Data**. Note that 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**.

After you modify the user data for your instance, you can execute it. For more information, see [User Data Execution \(p. 449\)](#).

No console output

For Windows instances, the instance console displays the output from the EC2Config service running on the instance. The output logs the status of tasks performed during the Windows boot process. If Windows boots successfully, the last message logged is Windows is Ready to use. Note that you can also display event log messages in the console, but this feature is not enabled by default. For more information, see [EC2 Service Properties \(p. 387\)](#).

To get the console output for your instance using the Amazon EC2 console, select the instance, choose **Actions**, **Instance Settings**, and then **Get System Log**. To get the console output using the command line, use one of the following commands: `get-console-output` (AWS CLI) or `Get-EC2ConsoleOutput` (AWS Tools for Windows PowerShell).

If the console output is empty, it could indicate an issue with the EC2Config service, such as a misconfigured configuration file, or that Windows failed to boot properly. To fix the issue, download and install the latest version of EC2Config. For more information, see [Installing the Latest Version of EC2Config \(p. 384\)](#).

Remote Desktop can't connect to the remote computer

Try the following to resolve issues related to connecting to your instance:

- Verify that you're using the correct public DNS hostname. (In the Amazon EC2 console, select the instance and check **Public DNS (IPv4)** in the details pane.) If your instance is in a VPC and you do not see a public DNS name, you must enable DNS hostnames. For more information, see [Using DNS with Your VPC in the Amazon VPC User Guide](#).
- Verify 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 \(p. 660\)](#).
- To connect to your instance using an IPv6 address, check that your local computer has an IPv6 address and is configured to use IPv6. If you launched an instance from a Windows Server 2008 SP2 AMI or earlier, your instance is not automatically configured to recognize an IPv6 address assigned to the instance. For more information, see [Configure IPv6 on Your Instances](#) in the *Amazon VPC User Guide*.
- Verify that your security group has a rule that allows RDP access. For more information, see [Create a Security Group \(p. 17\)](#).
- If you copied the password but get the error `Your credentials did not work`, try typing them manually when prompted. It's possible that you missed a character or got an extra whitespace character when you copied the password.
- Verify that the instance has passed status checks. For more information, see [Status Checks for Your Instances \(p. 490\)](#) and [Troubleshooting Instances with Failed Status Checks \(Amazon EC2 User Guide for Linux Instances\)](#).
- Verify that the route table for the subnet has a route that sends all traffic destined outside the VPC to the Internet gateway for the VPC. For more information, see [Creating a Custom Route Table \(Internet Gateways\)](#) in the *Amazon VPC User Guide*.
- Verify that Windows Firewall, or other firewall software, is not blocking RDP traffic to the instance. We recommend that you disable Windows Firewall and control access to your instance using security group rules. You can use [AWS Support-TroubleshootRDP \(p. 989\)](#) to [disable the Windows Firewall profiles using the SSM agent](#). To disable Windows Firewall on a Windows instance that is not configured for SSM, use [AWS Support-ExecuteEC2Rescue \(p. 992\)](#), or use the following manual steps:

Manual steps

1. Stop the affected instance and detach its root volume.
2. Launch a temporary instance in the same Availability Zone as the affected instance.

Warning

If your temporary instance is based on the same AMI that the original instance is based on, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses the AWS Windows AMI for Windows Server 2008 R2, launch the temporary instance using the AWS Windows AMI for Windows Server 2012.

3. Attach the root volume from the affected instance to this temporary instance. Connect to the temporary instance, open the **Disk Management** utility, and bring the drive online.
4. Open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, choose **Load Hive**. Select the drive, open the file `Windows\System32\config\SYSTEM`, and specify a key name when prompted (you can use any name).
5. Select the key you just loaded and navigate to `ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy`. For each key with a name of the form `xxxxProfile`, select the key

and change `EnableFirewall` from 1 to 0. Select the key again, and from the **File** menu, choose **Unload Hive**.

6. (Optional) If your temporary instance is based on the same AMI that the original instance is based on, you must complete the following steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision.

Warning

The following procedure describes how to edit the Windows Registry using Registry Editor. If you are not familiar with the Windows Registry or how to safely make changes using Registry Editor, see [Configure the Registry](#).

- a. Open a command prompt, type `regedit.exe`, and press Enter.
- b. In the **Registry Editor**, choose **HKEY_LOCAL_MACHINE** from the context menu (right-click), and then choose **Find**.
- c. Type **Windows Boot Manager** and then choose **Find Next**.
- d. Choose the key named `11000001`. This key is a sibling of the key you found in the previous step.
- e. In the right pane, choose **Element** and then choose **Modify** from the context menu (right-click).
- f. Locate the four-byte disk signature at offset `0x38` in the data. Reverse the bytes to create the disk signature, and write it down. For example, the disk signature represented by the following data is `E9EB3AA5`:

```
...  
0030  00 00 00 00 01 00 00 00  
0038  A5 3A EB E9 00 00 00 00  
0040  00 00 00 00 00 00 00 00  
...
```

- g. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
diskpart
```

- h. Run the following DiskPart command to select the volume. (You can verify that the disk number is 1 using the **Disk Management** utility.)

```
DISKPART> select disk 1  
Disk 1 is now the selected disk.
```

- i. Run the following DiskPart command to get the disk signature.

```
DISKPART> uniqueid disk  
Disk ID: 0C764FA8
```

- j. If the disk signature shown in the previous step doesn't match the disk signature from BCD that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

7. Using the **Disk Management** utility, bring the drive offline.

Note

The drive is automatically offline if the temporary instance is running the same operating system as the affected instance, so you won't need to bring it offline manually.

8. Detach the volume from the temporary instance. You can terminate the temporary instance if you have no further use for it.

9. Restore the root volume of the affected instance by attaching it as /dev/sda1.
 10. Start the instance.
- Verify that Network Level Authentication is disabled on instances that are not part of an Active Directory domain (use [AWS Support-TroubleshootRDP \(p. 989\)](#) to [disable NLA](#)).
 - Verify that the Remote Desktop Service (TermService) Startup Type is Automatic and the service is started (use [AWS Support-TroubleshootRDP \(p. 989\)](#) to [enable and start the RDP service](#)).
 - Verify that you are connecting to the correct Remote Desktop Protocol port, which by default is 3389 (use [AWS Support-TroubleshootRDP \(p. 989\)](#) to [read the current RDP port](#) and [change it back to 3389](#)).
 - Verify that Remote Desktop connections are allowed on your instance (use [AWS Support-TroubleshootRDP \(p. 989\)](#) to [enable Remote Desktop connections](#)).
 - Verify that the password has not expired. If the password has expired, you can reset it. For more information, see [Resetting a Lost or Expired Windows Administrator Password \(p. 974\)](#).
 - If you attempt to connect using a user account that you created on the instance and receive the error `The user cannot connect to the server due to insufficient access privileges`, verify that you granted the user the right to log on locally. For more information, see [Grant a Member the Right to Log On Locally](#).
 - If you attempt more than the maximum allowed concurrent RDP sessions, your session is terminated with the message `Your Remote Desktop Services session has ended. Another user connected to the remote computer, so your connection was lost.` By default, you are allowed two concurrent RDP sessions to your instance.

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 Mac App Store and use the app to connect to your instance.

RDP displays a black screen instead of the desktop

Try the following to resolve this issue:

- Check the console output for additional information. To get the console output for your instance using the Amazon EC2 console, select the instance, choose **Actions**, select **Instance Settings**, and then choose **Get System Log**.
- Verify that you are running the latest version of your RDP client.
- Try the default settings for the RDP client. For more information, see [Remote Session Environment](#).

-
- If you are using Remote Desktop Connection, try starting it with the /admin option as follows.

```
mstsc /v:instance /admin
```

- If the server is running a full-screen application, it might have stopped responding. Use Ctrl+Shift+Esc to start Windows Task Manager, and then close the application.
- If the server is over-utilized, it might have stopped responding. To monitor the instance using the Amazon EC2 console, select the instance and then select the **Monitoring** tab. If you need to change the instance type to a larger size, see [Changing the Instance Type \(p. 182\)](#).

Instance loses network connectivity or scheduled tasks don't run when expected

If you restart your instance and it loses network connectivity, it's possible that the instance has the wrong time.

By default, Windows instances use Coordinated Universal Time (UTC). If you set the time for your instance to a different time zone and then restart it, the time becomes offset and the instance temporarily loses its IP address. The instance regains network connectivity eventually, but this can take several hours. The amount of time that it takes for the instance to regain network connectivity depends on the difference between UTC and the other time zone.

This same time issue can also result in scheduled tasks not running when you expect them to. In this case, the scheduled tasks do not run when expected because the instance has the incorrect time.

To use a time zone other than UTC persistently, you must set the **RealTimelsUniversal** registry key. Without this key, an instance uses UTC after you restart it.

To resolve time issues that cause a loss of network connectivity

1. Ensure that you are running the recommended PV drivers. For more information, see [Upgrading PV Drivers on Your Windows Instances \(p. 414\)](#).
2. Verify that the following registry key exists and is set to 1: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimelsUniversal**

Windows Server 2012 R2 not available on the network

For information about troubleshooting a Windows Server 2012 R2 instance that is not available on the network, see [Windows Server 2012 R2 loses network and storage connectivity after an instance reboot \(p. 419\)](#).

Troubleshooting Remote Desktop Connection issues using AWS Systems Manager

AWSSupport-TroubleshootRDP

The AWSSupport-TroubleshootRDP automation document allows the user to check or modify common settings on the target instance that can impact Remote Desktop Protocol (RDP) connections, such as

the **RDP Port**, **Network Layer Authentication (NLA)**, and **Windows Firewall** profiles. By default, the document reads and outputs the values of these settings.

The AWSSupport-TroubleshootRDP automation document can be used only with instances that are enabled for [AWS Systems Manager](#), also referred to as a [managed instance](#). Your instance must have the SSM agent installed and have an IAM role attached with permissions to Systems Manager. You must change the default parameter values to modify the common RDP settings on the target instance.

1. Log in to the [Systems Manager Console](#).
2. Verify that you are in the same region as the impaired instance.
3. Open the [AWSSupport-TroubleshootRDP](#) document.
4. In **Execution Mode**, choose **Execute the entire automation at once**.
5. In **Input parameters**, in the **InstanceId** field, enable **Show interactive instance picker**.
6. Choose your Amazon EC2 instance. **Note:** If you don't see your instance in the list, it's not enabled for Systems Manager. To configure AWS Identity and Access Management (IAM) for SSM Agent, see [Create an Instance Profile for Systems Manager](#).
7. Review the [examples \(p. 990\)](#), then choose **Execute automation**.
8. To monitor the execution progress, in **Execution** status, wait for the status to change from **Pending** to **Success**. Expand **Outputs** to view the results. To view the output of individual steps, in **Executed Steps**, choose the **Step ID**.

Examples

The examples below show you how to accomplish common troubleshooting tasks with AWSSupport-TroubleshootRDP. Make sure to replace **INSTANCEID** and **REGION** with your values.

Example Check the current RDP status

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --parameters "InstanceId=INSTANCEID" --region REGION
```

Example

AWS Systems Manager Console Deeplink:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-TroubleshootRDP?region=REGION#documentVersion=$LATEST
```

Example Disable the Windows Firewall

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --parameters "InstanceId=INSTANCEID,Firewall=Disable" --region REGION
```

Example

AWS Systems Manager Console Deeplink:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-TroubleshootRDP?region=REGION#documentVersion=$LATEST&Firewall=Disable
```

Example Disable Network Level Authentication

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --  
parameters "InstanceId=INSTANCEID,NLASettingAction"
```

Example

AWS Systems Manager Console Deeplink:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-TroubleshootRDP?region=REGION#documentVersion
```

Example Set RDP Service Startup Type to Automatic and start the RDP service

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --  
parameters "InstanceId=INSTANCEID,RDPServiceStartupType=Auto, RDPServiceAction=Start" --  
region REGION
```

Example

AWS Systems Manager Console Deeplink:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-TroubleshootRDP?region=REGION#documentVersion=\$LATEST&RDPServiceStartupType=Auto&RDPServiceAction=Start
```

Example Restore the default RDP Port (3389)

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --  
parameters "InstanceId=INSTANCEID,RDPPortAction=Modify" --region REGION
```

Example

AWS Systems Manager Console Deeplink:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-TroubleshootRDP?region=REGION#documentVersion=\$LATEST&RDPPortAction=Modify
```

Example Allow remote connections

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --  
parameters "InstanceId=INSTANCEID,RemoteConnections=Enable" --region REGION
```

Example

AWS Systems Manager Console Deeplink:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-TroubleshootRDP?region=REGION#documentVersion=\$LATEST&RemoteConnections=Enable
```

AWSSupport-ExecuteEC2Rescue

The AWSSupport-ExecuteEC2Rescue automation document uses [Using EC2Rescue for Windows Server \(p. 954\)](#) to automatically troubleshoot and restore EC2 instance connectivity and RDP issues. For more information, see [Run the EC2Rescue Tool on Unreachable Instances](#).

Note

The AWSSupport-ExecuteEC2Rescue automation document can be used on any Amazon EC2 instance, but it requires a stop and restart of the instance. Systems Manager Automation stops the instance and creates an Amazon Machine Image (AMI). Data stored in instance store volumes are lost. The public IP address changes if you are not using an Elastic IP. For more information, see [Run the EC2Rescue Tool on Unreachable Instances](#).

1. Open the [Systems Manager console](#).
2. Verify that you are in the same region as the impaired Amazon EC2 instance.
3. Open the [AWSSupport-ExecuteEC2Rescue](#) document.
4. In **Execution Mode**, choose **Execute the entire automation at once**.
5. In the **Input parameters** section, for **UnreachableInstanceId**, enter the Amazon EC2 instance ID of the unreachable instance.
6. (Optional) For **LogDestination**, enter the Amazon Simple Storage Service (Amazon S3) bucket name if you want to collect operating system logs for troubleshooting your Amazon EC2 instance. Logs are automatically uploaded to the specified bucket.
7. Choose **Execute automation**.
8. To monitor the execution progress, in **Execution status**, wait for the status to change from **Pending** to **Success**. Expand **Outputs** to view the results. To view the output of individual steps, in **Executed Steps**, choose the **Step ID**.

Common Messages

This section includes tips to help you troubleshoot issues based on common messages.

Topics

- "[Password is not available](#)" (p. 992)
- "[Password not available yet](#)" (p. 993)
- "[Cannot retrieve Windows password](#)" (p. 993)
- "[Waiting for the metadata service](#)" (p. 993)
- "[Unable to activate Windows](#)" (p. 996)
- "[Windows is not genuine \(0x80070005\)](#)" (p. 997)
- "[No Terminal Server License Servers available to provide a license](#)" (p. 998)

"Password is not available"

To connect to a Windows instance using Remote Desktop, you must specify an account and password. The accounts and passwords provided are based on the AMI that you used to launch the instance. You can either retrieve the auto-generated password for the Administrator account, or use the account and password that were in use in the original instance from which the AMI was created.

If your Windows instance isn't configured to generate a random password, you'll receive the following message when you retrieve the auto-generated password using the console:

Password is not available.

The instance was launched from a custom AMI, or the default password has changed. A password cannot be retrieved for this instance. If you have forgotten your password, you can reset it using the Amazon EC2 configuration service. For more information, see [Passwords for a Windows Server instance](#).

Check the console output for the instance to see whether the AMI that you used to launch it was created with password generation disabled. If password generation is disabled, the console output contains the following:

```
Ec2SetPassword: Disabled
```

If password generation is disabled and you don't remember the password for the original instance, you can reset the password for this instance. For more information, see [Resetting a Lost or Expired Windows Administrator Password \(p. 974\)](#).

"Password not available yet"

To connect to a Windows instance using Remote Desktop, you must specify an account and password. The accounts and passwords provided are based on the AMI that you used to launch the instance. You can either retrieve the auto-generated password for the Administrator account, or use the account and password that were in use in the original instance from which the AMI was created.

Your password should be available within a few minutes. If the password isn't available, you'll receive the following message when you retrieve the auto-generated password using the console:

```
Password not available yet.  
Please wait at least 4 minutes after launching an instance before trying to retrieve the  
auto-generated password.
```

If it's been longer than four minutes and you still can't get the password, it's possible that EC2Config is disabled. Verify by checking whether the console output is empty. For more information, see [No console output \(p. 985\)](#).

Also verify that the AWS Identity and Access Management (IAM) account being used to access the Management Portal has the `ec2:GetPasswordData` action allowed. For more information about IAM permissions, see [What is IAM?](#).

"Cannot retrieve Windows password"

To retrieve the auto-generated password for the Administrator account, you must use the private key for the key pair that you specified when you launched the instance. If you didn't specify a key pair when you launched the instance, you'll receive the following message:

```
Cannot retrieve Windows password
```

You can terminate this instance and launch a new instance using the same AMI, making sure to specify a key pair.

"Waiting for the metadata service"

A Windows instance must obtain information from its instance metadata before it can activate itself. By default, the `WaitForMetaDataAvailable` setting ensures that the EC2Config service waits for the

instance metadata to be accessible before continuing with the boot process. For more information, see [Instance Metadata and User Data \(p. 452\)](#).

If the instance is failing the instance reachability test, try the following to resolve this issue.

- Check the CIDR block for your VPC. A Windows instance cannot boot correctly if it's launched into a VPC that has an IP address range from 224.0.0.0 to 255.255.255.255 (Class D and Class E IP address ranges). These IP address ranges are reserved, and should not be assigned to host devices. We recommend that you create a VPC with a CIDR block from the private (non-publicly routable) IP address ranges as specified in [RFC 1918](#).
- It's possible that the system has been configured with a static IP address. Try [creating a network interface \(p. 675\)](#) and [attaching it to the instance \(p. 678\)](#).
- **To enable DHCP on a Windows instance that you can't connect to**
 1. Stop the affected instance and detach its root volume.
 2. Launch a temporary instance in the same Availability Zone as the affected instance.

Warning

If your temporary instance is based on the same AMI that the original instance is based on, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses the AWS Windows AMI for Windows Server 2008 R2, launch the temporary instance using the AWS Windows AMI for Windows Server 2012.

3. Attach the root volume from the affected instance to this temporary instance. Connect to the temporary instance, open the **Disk Management** utility, and bring the drive online.
4. From the temporary instance, open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, choose **Load Hive**. Select the drive, open the file `Windows\System32\config\SYSTEM`, and specify a key name when prompted (you can use any name).
5. Select the key that you just loaded and navigate to `ControlSet001\Services\Tcpip\Parameters\Interfaces`. Each network interface is listed by a GUID. Select the correct network interface. If DHCP is disabled and a static IP address assigned, `EnableDHCP` is set to 0. To enable DHCP, set `EnableDHCP` to 1, and delete the following keys if they exist: `NameServer`, `SubnetMask`, `IPAddress`, and `DefaultGateway`. Select the key again, and from the **File** menu, choose **Unload Hive**.

Note

If you have multiple network interfaces, you'll need to identify the correct interface to enable DHCP. To identify the correct network interface, review the following key values `NameServer`, `SubnetMask`, `IPAddress`, and `DefaultGateway`. These values display the static configuration of the previous instance.

6. (Optional) If DHCP is already enabled, it's possible that you don't have a route to the metadata service. Updating EC2Config can resolve this issue.
 - a. [Download](#) and install the latest version of the EC2Config service. For more information about installing this service, see [Installing the Latest Version of EC2Config \(p. 384\)](#).
 - b. Extract the files from the .zip file to the Temp directory on the drive you attached.
 - c. Open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, choose **Load Hive**. Select the drive, open the file `Windows\System32\config\SOFTWARE`, and specify a key name when prompted (you can use any name).
 - d. Select the key that you just loaded and navigate to `Microsoft\Windows\CurrentVersion`. Select the `RunOnce` key. (If this key doesn't exist, right-click `CurrentVersion`, point to **New**, select **Key**, and name the key `RunOnce`.) Right-click, point to **New**, and select **String Value**. Enter `Ec2Install` as the name and `C:\Temp\Ec2Install.exe -q` as the data.
 - e. Select the key again, and from the **File** menu, choose **Unload Hive**.

7. (Optional) If your temporary instance is based on the same AMI that the original instance is based on, you must complete the following steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision.

Warning

The following procedure describes how to edit the Windows Registry using Registry Editor. If you are not familiar with the Windows Registry or how to safely make changes using Registry Editor, see [Configure the Registry](#).

- a. Open a command prompt, type **regedit.exe**, and press Enter.
- b. In the **Registry Editor**, choose **HKEY_LOCAL_MACHINE** from the context menu (right-click), and then choose **Find**.
- c. Type **Windows Boot Manager** and then choose **Find Next**.
- d. Choose the key named **11000001**. This key is a sibling of the key you found in the previous step.
- e. In the right pane, choose **Element** and then choose **Modify** from the context menu (right-click).
- f. Locate the four-byte disk signature at offset **0x38** in the data. Reverse the bytes to create the disk signature, and write it down. For example, the disk signature represented by the following data is **E9EB3AA5**:

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
...
```

- g. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
diskpart
```

- h. Run the following DiskPart command to select the volume. (You can verify that the disk number is 1 using the **Disk Management** utility.)

```
DISKPART> select disk 1

Disk 1 is now the selected disk.
```

- i. Run the following DiskPart command to get the disk signature.

```
DISKPART> uniqueid disk

Disk ID: 0C764FA8
```

- j. If the disk signature shown in the previous step doesn't match the disk signature from BCD that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

8. Using the **Disk Management** utility, bring the drive offline.

Note

The drive is automatically offline if the temporary instance is running the same operating system as the affected instance, so you won't need to bring it offline manually.

9. Detach the volume from the temporary instance. You can terminate the temporary instance if you have no further use for it.

10. Restore the root volume of the affected instance by attaching the volume as /dev/sda1.
11. Start the affected instance.

If you are connected to the instance, open an Internet browser from the instance and enter the following URL for the metadata server:

```
http://169.254.169.254/latest/meta-data/
```

If you can't contact the metadata server, try the following to resolve the issue:

- **Download** and install the latest version of the EC2Config service. For more information about installing this service, see [Installing the Latest Version of EC2Config \(p. 384\)](#).
- Check whether the Windows instance is running RedHat PV drivers. If so, update to Citrix PV drivers. For more information, see [Upgrading PV Drivers on Your Windows Instances \(p. 414\)](#).
- Verify that the firewall, IPSec, and proxy settings do not block outgoing traffic to the metadata service (169.254.169.254) or the KMS servers (the addresses are specified in TargetKMSServer elements in C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml).
- Verify that you have a route to the metadata service (169.254.169.254) using the following command.

```
route print
```

- Check for network issues that might affect the Availability Zone for your instance. Go to <http://status.aws.amazon.com/>.

"Unable to activate Windows"

Windows instances use Windows KMS activation. You can receive this message: A problem occurred when Windows tried to activate. Error Code 0xC004F074, if your instance can't reach the KMS server. Windows must be activated every 180 days. EC2Config attempts to contact the KMS server before the activation period expires to ensure that Windows remains activated.

If you encounter a Windows activation issue, use the following procedure to resolve the issue.

For EC2Config (Windows Server 2012 R2 AMIs and earlier)

1. **Download** and install the latest version of the EC2Config service. For more information about installing this service, see [Installing the Latest Version of EC2Config \(p. 384\)](#).
2. Log onto the instance and open the following file: C:\Program Files\Amazon\Ec2ConfigService\Settings\config.xml.
3. Locate the **Ec2WindowsActivate** plugin in the config.xml file. Change the state to **Enabled** and save your changes.
4. In the Windows Services snap-in, restart the EC2Config service or reboot the instance.

If this does not resolve the activation issue, follow these additional steps.

1. Set the KMS target: C:\> slmgr.vbs /skms 169.254.169.250:1688
2. Activate Windows: C:\> slmgr.vbs /ato

For EC2Launch (Windows Server 2016 AMIs and later)

1. Import the EC2Launch module:

```
PS C:\> Import-Module "C:\ProgramData\Amazon\EC2-Windows\Launch\Module\Ec2Launch.psd1"
```

2. Call the Add-Routes function:

```
PS C:\> Add-Routes
```

3. Call the Set-ActivationSettings function:

```
PS C:\> Set-Activationsettings
```

4. Then, run the following script to activate Windows:

```
PS C:\> cscript "${env:SYSTEMROOT}\system32\slmgr.vbs" /ato
```

For both EC2Config and EC2Launch, if you are still receiving an activation error, verify the following information.

- Verify that you have routes to the KMS servers. Open C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml and locate the TargetKMSServer elements. Run the following command and check whether the addresses for these KMS servers are listed.

```
route print
```

- Verify that the KMS client key is set. Run the following command and check the output.

```
C:\Windows\System32\slmgr.vbs /dlv
```

If the output contains Error: product key not found, the KMS client key isn't set. If the KMS client key isn't set, look up the client key as described in this Microsoft article: [KMS Client Setup Keys](#), and then run the following command to set the KMS client key.

```
C:\Windows\System32\slmgr.vbs /ipk client_key
```

- Verify that the system has the correct time and time zone. If you are using Windows Server 2008 or later and a time zone other than UTC, add the following registry key and set it to 1 to ensure that the time is correct: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal.
- If Windows Firewall is enabled, temporarily disable it using the following command.

```
netsh advfirewall set allprofiles state off
```

"Windows is not genuine (0x80070005)"

Windows instances use Windows KMS activation. If an instance is unable to complete the activation process, it reports that the copy of Windows is not genuine.

Try the suggestions for ["Unable to activate Windows" \(p. 996\)](#).

"No Terminal Server License Servers available to provide a license"

By default, Windows Server is licensed for two simultaneous users through Remote Desktop. If you need to provide more than two users with simultaneous access to your Windows instance through Remote Desktop, you can purchase a Remote Desktop Services client access license (CAL) and install the Remote Desktop Session Host and Remote Desktop Licensing Server roles.

Check for the following issues:

- You've exceeded the maximum number of concurrent RDP sessions.
- You've installed the Windows Remote Desktop Services role.
- Licensing has expired. If the licensing has expired, you can't connect to your Windows instance as a user. You can try the following:
 - Connect to the instance from the command line using an `/admin` parameter, for example:

```
mstsc /v:instance /admin
```

For more information, see the following Microsoft article: [Access Remote Desktop Via Command Line](#).

- Stop the instance, detach its Amazon EBS volumes, and attach them to another instance in the same Availability Zone to recover your data.

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
p3dn.24xlarge instances	2016-11-15	New p3dn.xlarge instances provide 100 Gbps of network bandwidth.	7 December 2018
Hibernate EC2 Linux instances	2016-11-15	You can hibernate a Linux instance if it's enabled for hibernation and it meets the hibernation prerequisites. For more information, see Hibernate Your Instance (p. 365) .	28 November 2018
Instances featuring 100 Gbps of network bandwidth	2016-11-15	New C5n instances can utilize up to 100 Gbps of network bandwidth.	26 November 2018
Spot console recommends a fleet of instances	2016-11-15	The Spot console recommends a fleet of instances based on Spot best practice (instance diversification) to meet the minimum hardware specifications (vCPUs, memory, and storage) for your application need. For more information, see Creating a Spot Fleet Request (p. 251) .	20 November 2018
New EC2 Fleet request type: instant	2016-11-15	EC2 Fleet now supports a new request type, instant, that you can use to synchronously provision capacity across instance types and purchase models. The instant request returns the launched instances in the API response, and takes no further action, enabling you to control if and when instances are launched. For more information, see EC2 Fleet Request Types (p. 336) .	14 November 2018
Instances featuring AMD EYPC processors	2016-11-15	New general purpose (M5a) and memory optimized instances (R5a) offer lower-priced options for microservices, small to medium databases, virtual desktops, development and test environments, business applications, and more.	6 November 2018
Spot savings information	2016-11-15	You can view the savings made from using Spot Instances for a single Spot Fleet or for all Spot Instances. For more information, see Savings From Purchasing Spot Instances (p. 236) .	5 November 2018
Console support for optimizing CPU options	2016-11-15	When you launch an instance, you can optimize the CPU options to suit specific workloads or business needs using the Amazon EC2 console. For more information, see Optimizing CPU Options (p. 425) .	31 October 2018

Feature	API Version	Description	Release Date
Console support for creating a launch template from an instance	2016-11-15	You can create a launch template using an instance as the basis for a new launch template using the Amazon EC2 console. For more information, see Creating a Launch Template (p. 323) .	30 October 2018
On-Demand Capacity Reservations	2016-11-15	You can reserve capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. This allows you to create and manage capacity reservations independently from the billing discounts offered by Reserved Instances (RI). For more information, see On-Demand Capacity Reservations (p. 302) .	25 October 2018
Bring Your Own IP Addresses (BYOIP)	2016-11-15	You can bring part or all of your public IPv4 address range from your on-premises network to your AWS account. After you bring the address range to AWS, it appears in your account as an address pool. You can create an Elastic IP address from your address pool and use it with your AWS resources. For more information, see Bring Your Own IP Addresses (BYOIP) (p. 657) .	23 October 2018
g3s.xlarge instances	2016-11-15	Expands the range of the accelerated-computing G3 instance family with the introduction of g3s.xlarge instances.	11 October 2018
Dedicated Host tag on create and console support	2016-11-15	You can tag your Dedicated Hosts on creation, and you can manage your Dedicated Host tags using the Amazon EC2 console. For more information, see Allocating Dedicated Hosts (p. 287) .	08 October 2018
High memory instances	2016-11-15	These instances are purpose-built to run large in-memory databases. They offer bare metal performance with direct access to host hardware. For more information, see Memory Optimized Instances (p. 152) .	27 September 2018
f1.4xlarge instances	2016-11-15	Expands the range of the accelerated-computing F1 instance family with the introduction of f1.4xlarge instances.	25 September 2018
Console support for scheduled scaling for Spot Fleet	2016-11-15	Increase or decrease the current capacity of the fleet based on the date and time. For more information, see Scale Spot Fleet Using Scheduled Scaling (p. 269) .	20 September 2018
T3 instances	2016-11-15	T3 instances are the next generation burstable general-purpose instance type that provide a baseline level of CPU performance with the ability to burst CPU usage at any time for as long as required. For more information, see Burstable Performance Instances (p. 116) .	21 August 2018

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 Strategies for Spot Instances (p. 336) .	26 July 2018
Allocation strategies for Spot Fleets	2016-11-15	You can specify whether On-Demand capacity is fulfilled by price (lowest price first) or priority (highest priority first). You can specify the number of Spot pools across which to allocate your target Spot capacity. For more information, see Allocation Strategy for Spot Instances (p. 230) .	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. 152) .	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. 152) .	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. 798) .	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. 323) .	11 July 2018
Tag Dedicated Hosts	2016-11-15	You can tag your Dedicated Hosts. For more information, see Tagging Dedicated Hosts (p. 291) .	3 July 2018
i3.metal instances	2016-11-15	i3.metal instances provide your applications with direct access to the physical resources of the host server, such as processors and memory. For more information, see Storage Optimized Instances (p. 159) .	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 <code>get-console-output</code> AWS CLI command.	9 May 2018

Feature	API Version	Description	Release Date
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. 425) .	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. 333) .	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. 229) .	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. 789) .	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. 695) .	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. 882) .	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. 113) .	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. 662) .	21 December 2017
Amazon Time Sync Service	2016-11-15	You can use the Amazon Time Sync Service to keep accurate time on your instance. For more information, see Setting the Time for a Windows Instance (p. 435) .	29 November 2017
T2 Unlimited	2016-11-15	T2 Unlimited instances can burst above the baseline for as long as required. For more information, see Burstable Performance Instances (p. 116) .	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. 321) .	29 November 2017

Feature	API Version	Description	Release Date
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. 693) .	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. 159) .	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. 277) .	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. 266) .	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. 152) .	28 November 2017
C5 instances	2016-11-15	C5 instances are designed for compute-heavy applications. For more information, see Compute Optimized Instances (p. 148) .	6 November 2017
Merge and split Convertible Reserved Instances	2016-11-15	You can exchange (merge) two or more Convertible Reserved Instances for a new Convertible Reserved Instance. You can also use the modification process to split a Convertible Reserved Instance into smaller reservations. For more information, see Exchanging Convertible Reserved Instances (p. 217) .	6 November 2017
P3 instances	2016-11-15	P3 instances are the next generation of compute-optimized GPU instances. For more information, see Windows Accelerated Computing Instances (p. 163) .	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. 301) .	16 October 2017

Feature	API Version	Description	Release Date
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. 276) .	18 September 2017
Tag NAT gateways	2016-11-15	You can tag your NAT gateway. For more information, see Tagging Your Resources (p. 892) .	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. 549) .	31 August 2017
Elastic Graphics	2016-11-15	Attach Elastic Graphics accelerators to your instances to accelerate the graphics performance of your applications. For more information, see Amazon Elastic Graphics (p. 475) .	29 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. 664) .	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 Windows Accelerated Computing Instances (p. 163) .	13 July 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. 892) . 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. 599) .	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. 159) .	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 Windows (p. 773) .	13 February 2017

Feature	API Version	Description	Release Date
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. 633) .	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. 239) .	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. 643) .	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. 152)	30 November 2016
New t2.xlarge and t2.2xlarge instance types	2016-09-15	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see Burstable Performance Instances (p. 116) .	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 Windows Accelerated Computing Instances (p. 163) .	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. 266) .	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. 684) .	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. 882) .	23 June 2016

Feature	API Version	Description	Release Date
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. 793) .	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 Troubleshoot an Unreachable Instance (p. 967) .	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. 152) .	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. 736) .	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. 500) .	23 March 2016
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. 264) .	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. 221) .	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. 882) .	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. 714) .	11 January 2016

Feature	API Version	Description	Release Date
New t2.nano instance type	2015-10-01	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see Burstable Performance Instances (p. 116) .	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. 283) .	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. 239) .	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. 254) .	29 September 2015
Spot fleet diversified allocation strategy	2015-04-15	You can now allocate Spot instances in multiple Spot pools using a single Spot fleet request. For more information, see Allocation Strategy for Spot Instances (p. 230) .	15 September 2015
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. 232) .	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. 517) .	23 July 2015
New t2.large instance type		T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see Burstable Performance Instances (p. 116) .	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

Feature	API Version	Description	Release Date
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. 229) .	18 May 2015
Migrate Elastic IP addresses to EC2-Classic	2015-04-15	You can migrate an Elastic IP address that you've allocated for use in EC2-Classic to be used in a VPC. For more information, see Migrating an Elastic IP Address from EC2-Classic (p. 706) .	15 May 2015
Importing VMs with multiple disks as AMIs	2015-03-01	The VM Import process now supports importing VMs with multiple disks as AMIs. For more information, see Importing a VM as an Image Using VM Import/Export in the <i>VM Import/Export User Guide</i> .	23 April 2015
New g2.8xlarge instance type		The new g2.8xlarge instance is backed by four high-performance NVIDIA GPUs, making it well suited for GPU compute workloads including large scale rendering, transcoding, machine learning, and other server-side workloads that require massive parallel processing power.	7 April 2015
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. 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. 159).</p>	24 March 2015
Amazon EC2 Systems Manager (SSM)		SSM enables you to configure and manage your EC2 instances.	17 February 2015
AWS Systems Manager for Microsoft SCVMM 1.5		You can now use AWS Systems Manager for Microsoft SCVMM to launch an instance and to import a VM from SCVMM to Amazon EC2. For more information, see Creating an EC2 Instance (p. 907) and Importing Your Virtual Machine (p. 911) .	21 January 2015

Feature	API Version	Description	Release Date
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. 373).</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. 148).</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. 708).</p>	7 January 2015
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. 279).</p>	5 January 2015
AWS Systems Manager for Microsoft SCVMM		<p>AWS Systems Manager for Microsoft SCVMM provides a simple, easy-to-use interface for managing AWS resources, such as EC2 instances, from Microsoft SCVMM. For more information, see AWS Systems Manager for Microsoft System Center VMM (p. 902).</p>	29 October 2014

Feature	API Version	Description	Release Date
DescribeVolumes pagination support	2014-09-01	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> .	23 October 2014
Added support for Amazon CloudWatch Logs		You can use Amazon CloudWatch Logs to monitor, store, and access your system, application, and custom log files from your instances or other sources. You can then retrieve the associated log data from CloudWatch Logs using the Amazon CloudWatch console, the CloudWatch Logs commands in the AWS CLI, or the CloudWatch Logs SDK.	10 July 2014
T2 instances	2014-06-15	T2 instances are designed to provide moderate base performance and the capability to burst to significantly higher performance as required by your workload. They are intended for applications that need responsiveness, high performance for limited periods of time, and a low cost. For more information, see Burstable Performance Instances (p. 116) .	30 June 2014
New EC2 Service Limits page		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.	19 June 2014
Amazon EBS General Purpose SSD Volumes	2014-05-01	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. 739) .	16 June 2014
Windows Server 2012 R2		AMIs for Windows Server 2012 R2 use the new AWS PV drivers. For more information, see AWS PV Drivers (p. 410) .	3 June 2014
AWS Management Pack		AWS Management Pack now supports for System Center Operations Manager 2012 R2. For more information, see AWS Management Pack for Microsoft System Center (p. 916) .	22 May 2014

Feature	API Version	Description	Release Date
Amazon EBS encryption	2014-05-01	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. 815) .	21 May 2014
R3 instances	2014-02-01	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. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instance Types .	9 April 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. 901) .	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. 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. 159) .	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
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 <code>RunInstances</code> API action. For more information and example policies, see Controlling Access to Amazon EC2 Resources (p. 562) .	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. 332) .	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 Windows Accelerated Computing Instances (p. 163) .	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. 315) .	10 October 2013
Modifying Amazon EC2 Reserved Instances	2013-08-15	You can now modify Reserved Instances in a region.	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. 647) .	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. 564) .	8 July 2013
Incremental Snapshot Copies	2013-02-01	You can now perform incremental snapshot copies. For more information, see Copying an Amazon EBS Snapshot (p. 793) .	11 June 2013

Feature	API Version	Description	Release Date
AWS Management Pack		The AWS Management Pack 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. For more information, see AWS Management Pack for Microsoft System Center (p. 916) .	8 May 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. 890) .	04 April 2013
Additional EBS-optimized instance types	2013-02-01	<p>The following instance types can now be launched as EBS-optimized instances: c1.xlarge, m2.2xlarge, m3.xlarge, and m3.2xlarge.</p> <p>For more information, see Amazon EBS-Optimized Instances (p. 807).</p>	19 March 2013
PV Drivers		To learn how to upgrade the paravirtualized (PV) drivers on your Windows AMI, see Upgrading PV Drivers on Your Windows Instances (p. 414) .	March 2013
Copy an AMI from one region to another	2013-02-01	<p>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.</p> <p>For more information, see Copying an AMI (p. 71).</p>	11 March 2013
Launch instances into a default VPC	2013-02-01	Your AWS account is capable of launching instances into either EC2-Classic or a VPC, or only into a VPC, on a region-by-region basis. If you can launch instances only into a VPC, we create a default VPC for you. When you launch an instance, we launch it into your default VPC, unless you create a nondefault VPC and specify it when you launch the instance.	11 March 2013
High-memory cluster (cr1.8xlarge) instance type	2012-12-01	Have large amounts of memory coupled with high CPU and network performance. These instances are well suited for in-memory analytics, graph analysis, and scientific computing applications.	21 January 2013
High storage (hs1.8xlarge) instance type	2012-12-01	High storage instances provide very high storage density and high sequential read and write performance per instance. They are well-suited for data warehousing, Hadoop/MapReduce, and parallel file systems.	20 December 2012
EBS snapshot copy	2012-12-01	You can use snapshot copies to create backups of data, to create new Amazon EBS volumes, or to create Amazon Machine Images (AMIs). For more information, see Copying an Amazon EBS Snapshot (p. 793) .	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. 758) . Also added new status checks for Provisioned IOPS SSD volumes. For more information, see Monitoring Volumes with Status Checks (p. 762) .	20 November 2012
Support for Windows Server 2012		<p>Amazon EC2 now provides you with several pre-configured Windows Server 2012 AMIs. These AMIs are immediately available for use in every region and for every 64-bit instance type. The AMIs support the following languages:</p> <ul style="list-style-type: none"> • English • Chinese Simplified • Chinese Traditional • Chinese Traditional Hong Kong • Japanese • Korean • Portuguese • Portuguese Brazil • Czech • Dutch • French • German • Hungarian • Italian • Polish • Russian • Spanish • Swedish • Turkish 	19 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

Feature	API Version	Description	Release Date
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. 736) .	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
IAM roles on Amazon EC2 instances	2012-06-01	IAM roles for Amazon EC2 provide: <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.		You can now manage your Spot instances as follows: <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 exporting Windows Server instances that you originally imported into EC2. Added support for timestamps on instance status and system status to indicate the date and time that a status check failed.	25 May 2012

Feature	API Version	Description	Release Date
EC2 instance export, and timestamps in instance and system status checks for Amazon VPC	2012-05-01	Added support for EC2 instance export to Citrix Xen, Microsoft Hyper-V, and VMware vSphere. Added support for timestamps in instance and system status checks.	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
Medium instances, support for 64-bit on all AMIs	2011-12-15	Added support for a new instance type and 64-bit information.	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
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. 665) .	21 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. 490) .	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
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

Feature	API Version	Description	Release Date
Simplified VM import process for users of the CLI tools	2011-07-15	The VM Import process is simplified with the enhanced functionality of <code>ImportInstance</code> and <code>ImportVolume</code> , which now will perform the upload of the images into Amazon EC2 after creating the import task. In addition, with the introduction of <code>ResumeImport</code> , users can restart an incomplete upload at the point the task stopped.	15 September 2011
Support for importing in VHD file format		VM Import can now import virtual machine image files in VHD format. The VHD file format is compatible with the Citrix Xen and Microsoft Hyper-V virtualization platforms. With this release, VM Import now supports RAW, VHD and VMDK (VMware ESX-compatible) image formats. For more information, see the VM Import/Export User Guide .	24 August 2011
Support for Windows Server 2003 R2		VM Import now supports Windows Server 2003 (R2). With this release, VM Import supports all versions of Windows Server supported by Amazon EC2.	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
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. 562) .	26 April 2011

Feature	API Version	Description	Release Date
Dedicated instances		Launched within your Amazon Virtual Private Cloud (Amazon VPC), Dedicated Instances are instances that are physically isolated at the host hardware level. Dedicated Instances let you take advantage of Amazon VPC and the AWS cloud, with benefits including on-demand elastic provisioning and pay only for what you use, while isolating your Amazon EC2 compute instances at the hardware level. For more information, see Dedicated Instances (p. 297) .	27 March 2011
Reserved Instances updates to the AWS Management Console		Updates to the AWS Management Console make it easier for users to view their Reserved Instances and purchase additional Reserved Instances, including Dedicated Reserved Instances.	27 March 2011
Support for Windows Server 2008 R2		Amazon EC2 now provides you with several pre-configured Windows Server 2008 R2 AMIs. These AMIs are immediately available for use in every region and in most 64-bit instance types, excluding t1.micro and HPC families. The AMIs will support multiple languages.	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. 452) and Instance Metadata Categories (p. 456) .	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. 771) .	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. 369) .	23 February 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

Feature	API Version	Description	Release Date
Basic monitoring for instances	2010-08-31	Added information about basic monitoring for EC2 instances.	12 December 2010
Filters and Tags	2010-08-31	Added information about listing, filtering, and tagging resources. For more information, see Listing and Filtering Your Resources (p. 887) and Tagging Your Amazon EC2 Resources (p. 890) .	19 September 2010
Idempotent Instance Launch	2010-08-31	Added information about ensuring idempotency when running instances.	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. 170) .	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. 562) .	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. 758) .	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
Reserved Instances with Windows		Amazon EC2 now supports Reserved Instances with Windows.	22 February 2010

AWS Glossary

For the latest AWS terminology, see the [AWS Glossary](#) in the *AWS General Reference*.