Amazon Elastic Compute Cloud: User Guide for Windows Instances

Copyright © 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
Access Amazon EC2 ................................................................................................................................ 3
Pricing for Amazon EC2 ............................................................................................................................ 3
PCI DSS compliance ................................................................................................................................. 4
Set up ...................................................................................................................................................... 5
Sign up for AWS ......................................................................................................................................... 5
Create a key pair ....................................................................................................................................... 5
Create a security group ............................................................................................................................. 6
Get started tutorial .................................................................................................................................. 9
Overview ................................................................................................................................................ 9
Prerequisites .......................................................................................................................................... 10
Step 1: Launch an instance ..................................................................................................................... 10
Step 2: Connect to your instance ............................................................................................................. 11
Step 3: Clean up your instance ................................................................................................................ 17
Next steps ............................................................................................................................................... 17
Best practices ........................................................................................................................................... 18
Amazon Machine Images ......................................................................................................................... 21
Boot modes .............................................................................................................................................. 21
Considerations ........................................................................................................................................ 22
Requirements for launching an instance with UEFI .................................................................................. 22
Determine the boot mode parameter of an AMI ....................................................................................... 22
Determine the supported boot modes of an instance type ........................................................................ 23
Determine the boot mode of an instance .................................................................................................... 24
Determine the boot mode of the OS .......................................................................................................... 25
Set the boot mode of an AMI ..................................................................................................................... 25
AWS Windows AMIs ............................................................................................................................... 27
Select an initial Windows AMI .................................................................................................................. 28
Keep your AMIs up-to-date ...................................................................................................................... 28
Virtualization types ................................................................................................................................. 28
Managed AWS Windows AMIs .................................................................................................................. 29
Create a custom Windows AMI ................................................................................................................ 37
Deregister your Windows AMI .................................................................................................................. 51
Specialized Windows AMIs ..................................................................................................................... 52
AWS Windows AMI Version History ....................................................................................................... 58
Find a Windows AMI ............................................................................................................................. 97
Find a Windows AMI using the Amazon EC2 console ........................................................................... 97
Find an AMI using the AWS Tools for Windows PowerShell ................................................................... 98
Find an AMI using the AWS CLI ............................................................................................................. 98
Find the latest Windows AMI using Systems Manager ......................................................................... 99
Use a Systems Manager parameter to find an AMI ................................................................................. 99
Shared AMIs .......................................................................................................................................... 102
Find shared AMIs .................................................................................................................................. 102
Make an AMI public ............................................................................................................................... 104
Share an AMI with specific AWS accounts .......................................................................................... 106
Use bookmarks ....................................................................................................................................... 108
Best Practices for shared Windows AMIs ................................................................................................ 108
Paid AMIs ............................................................................................................................................... 109
Sell your AMI ......................................................................................................................................... 110
Find a paid AMI ..................................................................................................................................... 110
Purchase a paid AMI ............................................................................................................................ 111
Get the product code for your instance ................................................................................................... 112
Retire ............................................................................................................................................ 439
Terminate .................................................................................................................................. 441
Recover ..................................................................................................................................... 447
Configure instances .................................................................................................................. 448
EC2Launch v2 ........................................................................................................................... 449
EC2Launch ............................................................................................................................... 486
EC2Config service .................................................................................................................... 493
PV drivers .................................................................................................................................. 519
AWS NVMe drivers .................................................................................................................. 538
Optimize CPU options ............................................................................................................ 540
Set the time ............................................................................................................................... 559
Set the password ...................................................................................................................... 563
Add Windows components ....................................................................................................... 564
Configure a secondary private IPv4 Address .......................................................................... 567
Run commands at launch ......................................................................................................... 572
Instance metadata and user data ............................................................................................. 579
SQL Server Clustering in EC2 ................................................................................................. 625
Upgrade Windows instances .................................................................................................. 631
Perform an in-place upgrade .................................................................................................... 632
Perform an automated upgrade ............................................................................................... 636
Migrate to latest generation instance types ............................................................................ 642
Migrate Microsoft SQL Server from Windows to Linux ........................................................... 648
Troubleshoot an upgrade ......................................................................................................... 655
Identify instances .................................................................................................................... 655
Inspect the instance identity document .................................................................................. 655
Inspect the system UUID ......................................................................................................... 655
Set up a Windows HPC cluster ............................................................................................... 656
Prerequisites .............................................................................................................................. 656
Step 1: Create your security groups ....................................................................................... 656
Step 2: Set up your Active Directory domain controller ....................................................... 659
Step 3: Configure your head node ........................................................................................... 660
Step 4: Set up the compute node ............................................................................................. 661
Step 5: Scale your HPC compute nodes (optional) ................................................................. 662
Fleets .......................................................................................................................................... 664
EC2 Fleet .................................................................................................................................... 664
EC2 Fleet limitations ................................................................................................................ 665
Burstable performance instances ............................................................................................. 665
EC2 Fleet request types .......................................................................................................... 666
EC2 Fleet configuration strategies .......................................................................................... 684
Work with EC2 Fleets ............................................................................................................. 692
Spot Fleet ................................................................................................................................... 711
How Spot Fleet works .............................................................................................................. 711
Spot Fleet requests .................................................................................................................. 718
CloudWatch metrics for Spot Fleet ........................................................................................ 737
Automatic scaling for Spot Fleet ............................................................................................. 740
Monitor fleet events ................................................................................................................ 746
EC2 Fleet event types ............................................................................................................... 746
Spot Fleet event types ............................................................................................................. 751
Create EventBridge rules ......................................................................................................... 755
Tutorials ..................................................................................................................................... 760
Tutorial: Use EC2 Fleet with instance weighting .................................................................... 760
Tutorial: Use EC2 Fleet with On-Demand as the primary capacity ....................................... 763
Tutorial: Use Spot Fleet with instance weighting ................................................................... 764
Example configurations ......................................................................................................... 766
EC2 Fleet example configurations .......................................................................................... 766
Spot Fleet example configurations ......................................................................................... 780
Fleet quotas .............................................................................................................................. 791
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic Graphics basics</td>
<td>793</td>
</tr>
<tr>
<td>Elastic Graphics limitations</td>
<td>795</td>
</tr>
<tr>
<td>Work with Elastic Graphics</td>
<td>796</td>
</tr>
<tr>
<td>Configure your security groups</td>
<td></td>
</tr>
<tr>
<td>Launch an instance with an Elastic Graphics accelerator</td>
<td>797</td>
</tr>
<tr>
<td>Install the required software for Elastic Graphics</td>
<td>798</td>
</tr>
<tr>
<td>Verify Elastic Graphics functionality on your instance</td>
<td>798</td>
</tr>
<tr>
<td>View Elastic Graphics information</td>
<td>800</td>
</tr>
<tr>
<td>Submit feedback</td>
<td>801</td>
</tr>
<tr>
<td>Use CloudWatch metrics to monitor Elastic Graphics</td>
<td>801</td>
</tr>
<tr>
<td>Use CloudWatch metrics to monitor Elastic Graphics</td>
<td></td>
</tr>
<tr>
<td>Elastic Graphics metrics</td>
<td>801</td>
</tr>
<tr>
<td>Elastic Graphics dimensions</td>
<td>802</td>
</tr>
<tr>
<td>View CloudWatch metrics for Elastic Graphics</td>
<td>802</td>
</tr>
<tr>
<td>Create CloudWatch alarms to monitor Elastic Graphics</td>
<td>802</td>
</tr>
<tr>
<td>Troubleshoot</td>
<td>803</td>
</tr>
<tr>
<td>Investigate application performance issues</td>
<td>803</td>
</tr>
<tr>
<td>Resolve unhealthy status issues</td>
<td>805</td>
</tr>
<tr>
<td>Automated and manual monitoring</td>
<td>807</td>
</tr>
<tr>
<td>Automated monitoring tools</td>
<td>807</td>
</tr>
<tr>
<td>Manual monitoring tools</td>
<td>808</td>
</tr>
<tr>
<td>Best practices for monitoring</td>
<td>808</td>
</tr>
<tr>
<td>Monitor the status of your instances</td>
<td>809</td>
</tr>
<tr>
<td>Instance status checks</td>
<td>809</td>
</tr>
<tr>
<td>Scheduled events</td>
<td>816</td>
</tr>
<tr>
<td>Monitor your instances using CloudWatch</td>
<td>839</td>
</tr>
<tr>
<td>Enable detailed monitoring</td>
<td>840</td>
</tr>
<tr>
<td>List available metrics</td>
<td>842</td>
</tr>
<tr>
<td>Get statistics for metrics</td>
<td>853</td>
</tr>
<tr>
<td>Graph metrics</td>
<td>861</td>
</tr>
<tr>
<td>Create an alarm</td>
<td>861</td>
</tr>
<tr>
<td>Create alarms that stop, terminate, reboot, or recover an instance</td>
<td>863</td>
</tr>
<tr>
<td>Automate Amazon EC2 with EventBridge</td>
<td>875</td>
</tr>
<tr>
<td>Log API calls with AWS CloudTrail</td>
<td>876</td>
</tr>
<tr>
<td>Amazon EC2 and Amazon EBS information in CloudTrail</td>
<td>876</td>
</tr>
<tr>
<td>Understand Amazon EC2 and Amazon EBS log file entries</td>
<td>877</td>
</tr>
<tr>
<td>Audit users that connect via EC2 Instance Connect</td>
<td>878</td>
</tr>
<tr>
<td>Monitor your .NET and SQL Server applications</td>
<td>879</td>
</tr>
<tr>
<td>Networking</td>
<td>880</td>
</tr>
<tr>
<td>Regions and Zones</td>
<td>880</td>
</tr>
<tr>
<td>Regions</td>
<td>881</td>
</tr>
<tr>
<td>Availability Zones</td>
<td>885</td>
</tr>
<tr>
<td>Local Zones</td>
<td>887</td>
</tr>
<tr>
<td>Wavelength Zones</td>
<td>890</td>
</tr>
<tr>
<td>AWS Outposts</td>
<td>892</td>
</tr>
<tr>
<td>Instance IP addressing</td>
<td>893</td>
</tr>
<tr>
<td>Private IPv4 addresses and internal DNS hostnames</td>
<td>893</td>
</tr>
<tr>
<td>Public IPv4 addresses and external DNS hostnames</td>
<td>894</td>
</tr>
<tr>
<td>Elastic IP addresses (IPv4)</td>
<td>895</td>
</tr>
<tr>
<td>Amazon DNS server</td>
<td>895</td>
</tr>
<tr>
<td>IPv6 addresses</td>
<td>895</td>
</tr>
<tr>
<td>Work with the IPv4 addresses for your instances</td>
<td>896</td>
</tr>
<tr>
<td>Work with the IPv6 addresses for your instances</td>
<td>899</td>
</tr>
<tr>
<td>Multiple IP addresses</td>
<td>901</td>
</tr>
<tr>
<td>Bring your own IP addresses</td>
<td>908</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Requirements and quotas</td>
<td>909</td>
</tr>
<tr>
<td>Configure your BYOIP address range</td>
<td>909</td>
</tr>
<tr>
<td>Work with your address range</td>
<td>916</td>
</tr>
<tr>
<td>Learn more</td>
<td>917</td>
</tr>
<tr>
<td>Assigning prefixes</td>
<td>917</td>
</tr>
<tr>
<td>Basics for assigning prefixes</td>
<td>918</td>
</tr>
<tr>
<td>Considerations and limits for prefixes</td>
<td>918</td>
</tr>
<tr>
<td>Work with prefixes</td>
<td>918</td>
</tr>
<tr>
<td>Elastic IP addresses</td>
<td>926</td>
</tr>
<tr>
<td>Elastic IP address pricing</td>
<td>926</td>
</tr>
<tr>
<td>Elastic IP address basics</td>
<td>926</td>
</tr>
<tr>
<td>Work with Elastic IP addresses</td>
<td>927</td>
</tr>
<tr>
<td>Use reverse DNS for email applications</td>
<td>933</td>
</tr>
<tr>
<td>Elastic IP address limit</td>
<td>934</td>
</tr>
<tr>
<td>Network interfaces</td>
<td>934</td>
</tr>
<tr>
<td>Network interface basics</td>
<td>935</td>
</tr>
<tr>
<td>IP addresses per network interface per instance type</td>
<td>936</td>
</tr>
<tr>
<td>Work with network interfaces</td>
<td>948</td>
</tr>
<tr>
<td>Scenarios for network interfaces</td>
<td>955</td>
</tr>
<tr>
<td>Best practices for configuring network interfaces</td>
<td>957</td>
</tr>
<tr>
<td>Requester-managed network interfaces</td>
<td>958</td>
</tr>
<tr>
<td>Network bandwidth</td>
<td>959</td>
</tr>
<tr>
<td>Available instance bandwidth</td>
<td>959</td>
</tr>
<tr>
<td>Monitor instance bandwidth</td>
<td>960</td>
</tr>
<tr>
<td>Enhanced networking</td>
<td>960</td>
</tr>
<tr>
<td>Enhanced networking support</td>
<td>961</td>
</tr>
<tr>
<td>Enable enhanced networking on your instance</td>
<td>961</td>
</tr>
<tr>
<td>Enhanced networking: ENA</td>
<td>961</td>
</tr>
<tr>
<td>Enhanced networking: Intel 82599 VF</td>
<td>968</td>
</tr>
<tr>
<td>Operating system optimizations</td>
<td>972</td>
</tr>
<tr>
<td>Network performance metrics</td>
<td>973</td>
</tr>
<tr>
<td>Placement groups</td>
<td>975</td>
</tr>
<tr>
<td>Cluster placement groups</td>
<td>976</td>
</tr>
<tr>
<td>Partition placement groups</td>
<td>976</td>
</tr>
<tr>
<td>Spread placement groups</td>
<td>977</td>
</tr>
<tr>
<td>Placement group rules and limitations</td>
<td>978</td>
</tr>
<tr>
<td>Create a placement group</td>
<td>979</td>
</tr>
<tr>
<td>Tag a placement group</td>
<td>980</td>
</tr>
<tr>
<td>Launch instances in a placement group</td>
<td>982</td>
</tr>
<tr>
<td>Describe instances in a placement group</td>
<td>983</td>
</tr>
<tr>
<td>Change the placement group for an instance</td>
<td>985</td>
</tr>
<tr>
<td>Delete a placement group</td>
<td>986</td>
</tr>
<tr>
<td>Network MTU</td>
<td>987</td>
</tr>
<tr>
<td>Jumbo frames (9001 MTU)</td>
<td>987</td>
</tr>
<tr>
<td>Path MTU Discovery</td>
<td>988</td>
</tr>
<tr>
<td>Check the path MTU between two hosts</td>
<td>988</td>
</tr>
<tr>
<td>Check and set the MTU on your Windows instance</td>
<td>989</td>
</tr>
<tr>
<td>Troubleshoot</td>
<td>991</td>
</tr>
<tr>
<td>Virtual private clouds</td>
<td>991</td>
</tr>
<tr>
<td>Amazon VPC documentation</td>
<td>991</td>
</tr>
<tr>
<td>Ports and Protocols</td>
<td>992</td>
</tr>
<tr>
<td>AllJoyn Router</td>
<td>992</td>
</tr>
<tr>
<td>Cast to Device</td>
<td>993</td>
</tr>
<tr>
<td>Core Networking</td>
<td>995</td>
</tr>
<tr>
<td>Delivery Optimization</td>
<td>1013</td>
</tr>
<tr>
<td>Diag Track</td>
<td>1014</td>
</tr>
<tr>
<td>DIAL Protocol Server</td>
<td>1014</td>
</tr>
</tbody>
</table>

Amazon Elastic Compute Cloud
User Guide for Windows Instances
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, hibernate, 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 Get 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

- Set up to use Amazon EC2 (p. 5)
- Tutorial: Get started with Amazon EC2 Windows instances (p. 9)

Basics

- Amazon EC2 Windows instances (p. 138)
- Instance types (p. 141)
- Tags (p. 1450)

Networking and security

- Key pairs (p. 1127)
- Security groups (p. 1135)
- Elastic IP addresses (p. 926)
- Virtual private clouds (p. 991)

Storage

- Amazon EBS (p. 1159)
- Instance store (p. 1392)

Working with Windows instances

- AWS Systems Manager Run Command in the AWS Systems Manager User Guide

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

Related services

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

- Amazon EC2 Auto Scaling User Guide
- AWS CloudFormation User Guide
- AWS Elastic Beanstalk Developer Guide
- AWS OpsWorks User Guide

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

To get a managed relational database in the cloud, use Amazon Relational Database Service (Amazon RDS) to launch a database instance. Although you can set up a database on an EC2 instance, Amazon RDS offers the advantage of handling your database management tasks, such as patching the software, backing up, and storing the backups. For more information, see the Amazon Relational Database Service Developer Guide.
To make it easier to manage Docker containers on a cluster of EC2 instances, use Amazon Elastic Container Service (Amazon ECS). For more information, see the Amazon Elastic Container Service Developer Guide or the Amazon Elastic Container Service User Guide for AWS Fargate.

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

To detect potentially unauthorized or malicious use of your EC2 instances, use Amazon GuardDuty. For more information see the Amazon GuardDuty User Guide.

Access Amazon EC2

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

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

AWS Command Line Interface (CLI)

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

AWS Tools for Windows PowerShell

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

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

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

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

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.

Savings Plans

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

Reserved Instances

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

Spot Instances

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

For a complete list of charges and prices for Amazon EC2, see Amazon EC2 pricing.

To calculate the cost of a sample provisioned environment, see Cloud Economics Center.

To see your bill, go to the Billing and Cost Management Dashboard in the AWS Billing and Cost Management console. Your bill contains links to usage reports that provide details about your bill. To learn more about AWS account billing, see AWS Billing and Cost Management User Guide.

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.
Set up to use Amazon EC2

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

1. Sign up for AWS (p. 5)
2. Create a key pair (p. 5)
3. Create a security group (p. 6)

When you are finished, you will be ready for the Amazon EC2 Getting started (p. 9) tutorial.

Sign up for AWS

When you sign up for Amazon Web Services, 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

2. Follow the online instructions.

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

Create a key pair

AWS uses public-key cryptography to secure the login information for your instance. 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 by using the Amazon EC2 console. Note that if you plan to launch instances in multiple Regions, you’ll need to create a key pair in each Region. For more information about Regions, see Regions and Zones (p. 880).

To create your key pair

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Key Pairs.
3. Choose Create key pair.
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 Zones (p. 880).

Prerequisites

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

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

New console

To create a security group with least privilege

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select a Region for the security group. Security groups are specific to a Region, so you should select the same Region in which you created your key pair.
3. In the navigation pane, choose Security Groups.
4. Choose Create security group.
5. For Basic details, do the following:
   a. Enter a name for the new security group and a description. Use a name that is easy for you to remember, such as your user name, followed by _SG_, plus the Region name. For example, me_SG_uswest2.
b. In the VPC list, select your default VPC for the Region.

6. For **Inbound rules**, create rules that allow specific traffic to reach your instance. For example, use the following rules for a web server that accepts HTTP and HTTPS traffic. For more examples, see Security group rules for different use cases (p. 1149).

   a. Choose **Add rule**. For **Type**, choose HTTP. For **Source**, choose Anywhere.
   
   b. Choose **Add rule**. For **Type**, choose HTTPS. For **Source**, choose Anywhere.
   
   c. Choose **Add rule**. For **Type**, choose RDP. For **Source**, do one of the following:
      
      • Choose My IP to automatically add the public IPv4 address of your local computer.
      
      • 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, do not choose Anywhere for **Source** with a rule for RDP. This would allow access to your instance from all IP addresses on the internet. This is acceptable for a short time in a test environment, but it is unsafe for production environments.

7. For **Outbound rules**, keep the default rule, which allows all outbound traffic.

8. Choose **Create security group**.

**Old console**

**To create a security group with least privilege**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).

2. In the navigation pane, choose **Security Groups**.

3. Choose **Create Security Group**.

4. Enter a name for the new security group and a description. Use a name that is easy for you to remember, such as your user name, followed by _SG_, plus the Region name. For example, me_SG_uswest2.

5. In the VPC list, select your default VPC for the Region.

6. On the **Inbound** tab, create the following rules (choose **Add rule** for each new rule):

   • Choose HTTP from the **Type** list, and make sure that **Source** is set to Anywhere (0.0.0.0/0).
   
   • Choose HTTPS from the **Type** list, and make sure that **Source** is set to Anywhere (0.0.0.0/0).
   
   • Choose 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, do not allow RDP access from all IP addresses to your instance. This is acceptable for a short time in a test environment, but it is unsafe for production environments.

7. On the **Outbound** tab, keep the default rule, which allows all outbound traffic.

8. Choose **Create**.
Command line

To create a security group with least privilege

Use one of the following commands:

- `create-security-group` (AWS CLI)

For more information, see Amazon EC2 security groups for Windows instances (p. 1135).
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.
Prerequisites

Before you begin, be sure that you've completed the steps in Set up to use Amazon EC2 (p. 5).

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 Launch an instance using the Launch Instance Wizard (p. 392). For information about other ways to launch your instance, see Launch your instance (p. 390).

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 later. Notice that these AMIs are marked "Free tier eligible."
4. On the Choose an Instance Type page, you can select the hardware configuration of your instance. Select the t2.micro instance type, which is selected by default. The t2.micro instance type is eligible for the free tier. In Regions where t2.micro is unavailable, you can use a t3.micro instance under the free tier. For more information, see AWS Free Tier.
5. On the Choose an Instance Type page, 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.
Step 2: Connect to your instance

To connect to a Windows instance, you must retrieve the initial administrator password and then enter this password when you connect to your instance using Remote Desktop. It takes a few minutes after instance launch before this password is available.

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.

If you receive an error while attempting to connect to your instance, see Remote Desktop can't connect to the remote computer (p. 1469).

New console

To connect to your Windows instance using an RDP client

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, select Instances. Select the instance and then choose Connect.
3. On the Connect to instance page, choose the RDP client tab, and then choose Get password.
4. Choose **Browse** and navigate to the private key (.pem) file you created when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file to this window.

5. Choose **Decrypt Password**. The console displays the default administrator password for the instance under **Password**, replacing the **Get password** link shown previously. Save the password in a safe place. This password is required to connect to the instance.
6. Choose **Download remote desktop file**. Your browser prompts you to either open or save the RDP shortcut file. When you have finished downloading the file, choose **Cancel** to return to the **Instances** page.
   - 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. Choose **Connect** to continue to connect to your instance.
8. The administrator account is chosen by default. Copy and paste the password that you saved previously.

**Tip**
If you receive a "Password Failed" error, try entering the password manually. Copying and pasting content can corrupt it.

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 (Windows) or Continue (Mac OS X) if you trust the certificate.
a. If you are using Remote Desktop Connection on a Windows computer, 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 Thumbprint (Windows) or SHA1 Fingerprints (Mac OS X). This is the unique identifier for the remote computer's security certificate.

c. In the Amazon EC2 console, select the instance, choose Actions, Monitor and troubleshoot, Get system log.

d. In the system log output, look for 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 on a Windows computer, 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 X] 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, see Add spaces and switch between them.
Old console

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 (.pem) 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. 1469).

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. 17).

**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. 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 Stop and start your instance (p. 425).

To terminate your instance

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

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

Next steps

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

- Learn how to remotely manage your EC2 instance using Run Command. For more information, see AWS Systems Manager Run Command in the AWS Systems Manager User Guide.
- Configure a CloudWatch alarm to notify you if your usage exceeds the Free Tier. For more information, see Tracking your AWS Free Tier usage in the AWS Billing and Cost Management User Guide.
- Add an EBS volume. For more information, see Create an Amazon EBS volume (p. 1183) and Attach an Amazon EBS volume to an instance (p. 1186).
Best practices for Windows on Amazon EC2

This list of practices will help you get the best results from running Windows on Amazon EC2.

**Update Windows drivers**

Maintain the latest drivers on all Windows EC2 instances to ensure the latest issue fixes and performance enhancements are applied across your fleet. Depending on your instance type, you should update AWS PV, ENA, and NVMe drivers.

- Leverage Trusted Advisor to keep Amazon EC2 Windows up to date with AWS-provided Windows drivers.
- Use SNS topics to receive updates for new driver releases.
- Use the AWS Systems Manager SSM document `AWSSupport-UpgradeWindowsAWSDrivers` to easily apply the updates across your instances.

**Launch new instances with the latest Windows AMIs**

AWS releases new Windows AMIs each month, which contain the latest OS patches, drivers, and launch agents. You should leverage the latest AMI when you launch new instances or when you build your own custom images.

- To build with the latest available AMIs, see [Query for the Latest Windows AMI Using Systems Manager Parameter Store](#).

**Test system/application performance before migration**

Migrating enterprise applications to AWS can involve many variables and configurations. Always performance test the EC2 solution to ensure that:

- Instance types are properly configured, including instance size, enhanced networking, and tenancy (shared or dedicated).
- Instance topology is appropriate for the workload and leverages high-performance features when necessary (dedicated tenancy, placement groups, instance store volumes, bare metal).

**Update launch agents**

Update to the latest EC2Launch v2 (Windows Server 2008 and later) agent to ensure that the latest issue fixes are applied across your fleet. To update, see the instructions at [Install the latest version of EC2Launch v2](#).

If you want to continue to use the EC2Config (Windows Server 2012 R2 and earlier) or EC2Launch (Windows Server 2016 and later) agents, ensure that the latest issue fixes are applied across your fleet.

- For EC2Config update instructions, see [Installing the Latest Version of EC2Config](#).
- For EC2Launch update instructions, see [Installing the Latest Version of EC2Launch](#).
Security

When securing Windows instances, we recommend that you implement Active Directory Domain Services to enable a scalable, secure, and manageable infrastructure for distributed locations. Additionally, after launching instances through the AWS Console or using an Amazon EC2 provisioning tool, such as AWS CloudFormation, it is good practice to utilize native OS features, such as Microsoft Windows PowerShell DSC to maintain configuration state in the event that configuration drift occurs.

Windows instances in AWS should adhere to the following high-level best practices:

• **Least Access:** Grant access only to systems and locations that are trusted and expected. This applies to all Microsoft products such as Active Directory, Microsoft business productivity servers, and infrastructure services such as Remote Desktop Services, reverse proxy servers, IIS web servers, etc. Use AWS capabilities such as Amazon EC2 instance security groups, network access control lists (ACLs), and Amazon VPC public/private subnets to layer security across multiple locations in an architecture. Within a Windows instance, customers can use Windows Firewall to further layer a defense-in-depth strategy within their deployment. Install only the OS components and applications that are necessary for the system to function as designed. Configure infrastructure services such as IIS to run under service accounts or to use features such as application pool identities to access resources locally and remotely across your infrastructure.

• **Least Privilege:** Determine the minimum set of privileges that instances and accounts need in order to perform their functions. Restrict these servers and users to only allow these defined permissions. Use techniques such as Role Based Access Controls to reduce the surface area of administrative accounts and create the most limited roles to accomplish a task. Use OS features such as Encrypting File System (EFS) within NTFS to encrypt sensitive data at rest and control application and user access to it.

• **Configuration Management:** Create a baseline server configuration that incorporates up-to-date security patches and host-based protection suites that include anti-virus, anti-malware, intrusion detection/prevention, and file integrity monitoring. Assess each server against the current recorded baseline to identify and flag any deviations. Ensure each server is configured to generate and securely store appropriate log and audit data. For more information about updating your Windows instance, see Update your Windows instance.

• **Change Management:** Create processes to control changes to server configuration baselines and work toward fully automated change processes. Also, leverage Just Enough Administration (JEA) with Windows PowerShell DSC to limit administrative access to the minimum required functions.

• **Audit Logs:** Audit access and all changes to Amazon EC2 instances to verify server integrity and ensure only authorized changes are made. Leverage features such as Enhanced Logging for IIS to enhance default logging capabilities. AWS capabilities such as VPC Flow Logs and AWS CloudTrail are also available to audit network access, including allowed/denied requests and API calls, respectively.

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 Preserve Amazon EBS volumes on instance termination (p. 445).

• 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, hibernate, or terminate your instance. If you use instance store for database storage, ensure that you have a cluster with a replication factor that ensures fault tolerance.

• Encrypt EBS volumes and snapshots. For more information, see Amazon EBS encryption (p. 1327).

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. 579) and Tag your Amazon EC2 resources (p. 1450).
• View your current limits for Amazon EC2. Plan to request any limit increases in advance of the time that you'll need them. For more information, see Amazon EC2 service quotas (p. 1463).

Backup and recovery

• Regularly back up your EBS volumes using Amazon EBS snapshots (p. 1207), and create an Amazon Machine Image (AMI) (p. 21) 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. 893).
• Monitor and respond to events. For more information, see Monitor Amazon EC2 (p. 806).
• 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. 934). For an automated solution, you can use Amazon EC2 Auto Scaling. For more information, see the Amazon EC2 Auto Scaling User Guide.
• Regularly test the process of recovering your instances and Amazon EBS volumes if they fail.

Networking

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

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

An AMI includes the following:

- One or more Amazon Elastic Block Store (Amazon EBS) snapshots, or, for instance-store-backed AMIs, a template for the root volume of the instance (for example, an operating system, an application server, and applications).
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it's launched.

Contents

- Boot modes (p. 21)
- AWS Windows AMIs (p. 27)
- Find a Windows AMI (p. 97)
- Shared AMIs (p. 102)
- Paid AMIs (p. 109)
- AMI lifecycle (p. 113)
- Use encryption with EBS-backed AMIs (p. 127)
- Understand AMI billing information (p. 133)

Boot modes

When a computer boots, the first software that it runs is responsible for initializing the platform and providing an interface for the operating system to perform platform-specific operations.

Default boot modes

In EC2, two variants of the boot mode software are supported: Legacy BIOS and Unified Extensible Firmware Interface (UEFI). By default, Intel and AMD instance types run on Legacy BIOS, and Graviton instance types run on UEFI.

Running Intel and AMD instances types on UEFI

Most Intel and AMD instance types can run on both UEFI and Legacy BIOS. To use UEFI, you must select an AMI with the boot mode parameter set to uefi, and the operating system contained in the AMI must be configured to support UEFI.

Purpose of the AMI boot mode parameter

The AMI boot mode parameter signals to EC2 which boot mode to use when launching an instance. When the boot mode parameter is set to uefi, EC2 attempts to launch the instance on UEFI. If the operating system is not configured to support UEFI, the instance launch might be unsuccessful.

Warning

Setting the boot mode parameter does not automatically configure the operating system for the specified boot mode. The configuration is specific to the operating system. For the configuration instructions, see the manual for your operating system.

Possible boot mode parameter on an AMI
The AMI boot mode parameter is optional. An AMI can have one of the following boot mode parameter values: `uefi` or `legacy-bios`. Some AMIs do not have a boot mode parameter. For AMIs with no boot mode parameter, the instances launched from these AMIs use the default value of the instance type—`uefi` on Graviton, and `legacy-bios` on all Intel and AMD instance types.

**Considerations**

- Default boot modes:
  - Intel and AMD instance types: Legacy BIOS
  - Graviton instance types: UEFI
  - Intel and AMD instance types that support UEFI, in addition to Legacy BIOS:
    - Virtualized: C5, C5a, C5ad, C5d, C5n, D3, D3en, G4, i3en, M5, M5a, M5ad, M5d, M5dn, M5n, M5zn, R5, R5a, R5ad, R5b, R5d, R5dn, R5n, T3, T3a, and z1d
- UEFI Secure Boot is currently not supported.

**Requirements for launching an instance with UEFI**

To launch an instance in UEFI mode, you must select an instance type that supports UEFI, and configure the AMI and the OS for UEFI, as follows:

- **Instance type** – When launching an instance, you must select an instance type that supports UEFI. For more information, see Determine the supported boot modes of an instance type (p. 23).
- **AMI** – When launching an instance, you must select an AMI that is configured for UEFI. The AMI must be configured as follows:
  - **OS** – The operating system contained in the AMI must be configured to use UEFI; otherwise, the instance launch will fail. For more information, see Determine the boot mode of the OS (p. 25).
  - **AMI boot mode parameter** – The boot mode parameter of the AMI must be set to `uefi`. For more information, see Determine the boot mode parameter of an AMI (p. 22).

AWS does not provide AMIs that are already configured to support UEFI. You must configure the AMI (p. 25), import the AMI through VM Import/Export, or import the AMI through CloudEndure.

**Determine the boot mode parameter of an AMI**

The AMI boot mode parameter is optional. An AMI can have one of the following boot mode parameter values: `uefi` and `legacy-bios`.

Some AMIs do not have a boot mode parameter. When an AMI has no boot mode parameter, the instances launched from the AMI use the default value of the instance type, which is `uefi` on Graviton, and `legacy-bios` on Intel and AMD instance types.
To determine the boot mode parameter of an AMI (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose AMIs, and then select the AMI.
3. On the Details tab, inspect the Boot mode field.

To determine the boot mode parameter of an AMI when launching an instance (console)

When launching an instance using the launch instance wizard, at the step to select an AMI, inspect the Boot mode field. For more information, see Step 1: Choose an Amazon Machine Image (AMI) (p. 392).

To determine the boot mode parameter of an AMI (AWS CLI version 1.19.34 and later and version 2.1.32 and later)

Use the describe-images command to determine the boot mode of an AMI.

```
aws ec2 --region us-east-1 describe-images --image-id ami-0abcdef1234567890
```

Expected output

```
{
   "Images": [ {
      ...
      "EnaSupport": true,
      "Hypervisor": "xen",
      "ImageOwnerAlias": "amazon",
      "Name": "UEFI_Boot_Mode_Enabled-Windows_Server-2016-English-Full-Base-2020.09.30",
      "RootDeviceName": "/dev/sda1",
      "RootDeviceType": "ebs",
      "SriovNetSupport": "simple",
      "VirtualizationType": "hvm",
      "BootMode": "uefi"
   }
   ]
}
```

Determine the supported boot modes of an instance type

To determine the supported boot modes of an instance type (AWS CLI version 1.19.34 and later and version 2.1.32 and later)

Use the describe-instance-types command to determine the supported boot modes of an instance type. By including the --query parameter, you can filter the output. In this example, the output is filtered to return only the supported boot modes.

The following example shows that m5.2xlarge supports both UEFI and Legacy BIOS boot modes.

```
aws ec2 --region us-east-1 describe-instance-types --instance-types m5.2xlarge --query "InstanceTypes[\*].SupportedBootModes"
```

Expected output
The following example shows that t2.xlarge supports only Legacy BIOS.

.aws ec2 --region us-east-1 describe-instance-types --instance-types t2.xlarge --query "InstanceTypes[*].SupportedBootModes"

Expected output

```
[
  [
      "legacy-bios"
  ]
]
```

**Determine the boot mode of an instance**

When an instance is launched, the value for its boot mode parameter is determined by the value of the boot mode parameter of the AMI used to launch it, as follows:

- An AMI with a boot mode parameter of **uefi** creates an instance with a boot mode parameter of **uefi**.
- An AMI with a boot mode parameter of **legacy-bios** creates an instance with no boot mode parameter. An instance with no boot mode parameter uses its default value, which is **legacy-bios** in this case.
- An AMI with no boot mode parameter value creates an instance with no boot mode parameter value.

The value of the instance's boot mode parameter determines the mode in which it boots. If there is no value, the default boot mode is used, which is **uefi** on Graviton, and **legacy-bios** on Intel and AMD instance types.

**To determine the boot mode of an instance (console)**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. On the **Details** tab, inspect the **Boot mode** field.

**To determine the boot mode of an instance (AWS CLI version 1.19.34 and later and version 2.1.32 and later)**

Use the `describe-instances` command to determine the boot mode of an instance.

```
aws ec2 --region us-east-1 describe-instances --instance-ids i-1234567890abcdef0
```

Expected output

```
{
  "Reservations": [
    {
      "Groups": [],
      "Instances": [
```
Determine the boot mode of the OS

The boot mode of the OS guides EC2 on which boot mode to use to boot an instance. To view whether the operating system of your instance is configured for UEFI, you need to connect to your instance via RDP.

To determine the boot mode of the instance's OS

1. Connect to your Windows instance using RDP (p. 413).
2. Go to System Information and check the BIOS Mode row.

Set the boot mode of an AMI

When you create an AMI using the register-image command, you can set the boot mode of the AMI to either uefi or legacy-bios.

To convert an existing Legacy BIOS-based instance to UEFI, or an existing UEFI-based instance to Legacy BIOS, you need to perform a number of steps: First, modify the instance's volume and OS to support the selected boot mode. Then, create a snapshot of the volume. Finally, use register-image to create the AMI using the snapshot.

You can't set the boot mode of an AMI using the create-image command. With create-image, the AMI inherits the boot mode of the EC2 instance used for creating the AMI. For example, if you create an AMI from an EC2 instance running on Legacy BIOS, the AMI boot mode will be configured as legacy-bios.

Warning
Before proceeding with these steps, you must first make suitable modifications to the instance's volume and OS to support booting via the selected boot mode; otherwise, the resulting AMI will not be usable. For example, if you are converting a Legacy BIOS-based instance to UEFI, you can use the MBR2GPT tool from Microsoft to convert the system disk from MBR to GPT. The modifications that are required are operating system-specific. For more information, see the manual for your operating system.

To set the boot mode of an AMI (AWS CLI version 1.19.34 and later and version 2.1.32 and later)

1. Make suitable modifications to the instance's volume and OS to support booting via the selected boot mode. The modifications that are required are operating system-specific. For more information, see the manual for your operating system.

Note
If you don't perform this step, the AMI will not be usable.
2. To find the volume ID of the instance, use the `describe-instances` command. You'll create a snapshot of this volume in the next step.

```bash
aws ec2 describe-instances --region us-east-1 --instance-ids i-1234567890abcdef0
```

Expected output

```json
...
  "BlockDeviceMappings": [
    {
      "DeviceName": "/dev/sda1",
      "Ebs": {
        "AttachTime": "",
        "DeleteOnTermination": true,
        "Status": "attached",
        "VolumeId": "vol-1234567890abcdef0"
      }
    }
  ]
...
```

3. To create a snapshot of the volume, use the `create-snapshot` command. Use the volume ID from the previous step.

```bash
aws ec2 create-snapshot --region us-east-1 --volume-id vol-1234567890abcdef0 --description "add text"
```

Expected output

```json
{
  "Description": "add text",
  "Encrypted": false,
  "OwnerId": "123",
  "Progress": "",
  "SnapshotId": "snap-01234567890abcdef",
  "StartTime": "",
  "State": "pending",
  "VolumeId": "vol-1234567890abcdef0",
  "VolumeSize": 30,
  "Tags": []
}
```

4. Note the snapshot ID in the output from the previous step.

5. Wait until the snapshot creation is completed before going to the next step. To query the state of the snapshot, use the `describe-snapshots` command.

```bash
aws ec2 describe-snapshots --region us-east-1 --snapshot-ids snap-01234567890abcdef
```

Example output

```json
{
  "Snapshots": [
    {
      "Description": "This is my snapshot",
      "Encrypted": false,
      "VolumeId": "vol-049df61146c4d7901",
      "State": "completed",
      "VolumeSize": 8,
      "StartTime": "2019-02-28T21:28:32.000Z",
      "Progress": "100%",
    }
  ]
}
```
6. To create a new AMI, use the `register-image` command. Use the snapshot ID that you noted in the earlier step. To set the boot mode to UEFI, add the `--boot-mode uefi` parameter to the command.

   ```bash
   aws ec2 register-image
   --region us-east-1
   --description "add description"
   --name "add name"
   --block-device-mappings "DeviceName=/dev/sda1,Ebs={SnapshotId=snap-01234567890abcdef,DeleteOnTermination=true}" 
   --architecture x86_64
   --root-device-name /dev/sda1
   --virtualization-type hvm
   --ena-support
   --boot-mode uefi
   
   Expected output
   
   ```
   
   ```json
   {
   "ImageId": "ami-new_ami_123"
   }
   ```

7. To verify that the newly-created AMI has the boot mode that you specified in the previous step, use the `describe-images` command.

   ```bash
   aws ec2 describe-images --region us-east-1 --image-id ami-new_ami_123
   
   Expected output
   
   ```
   
   ```json
   {
   "Images": [
   {
   "Architecture": "x86_64",
   "CreationDate": "2021-01-06T14:31:04.000Z",
   "ImageId": "ami-new_ami_123",
   "ImageLocation": "",
   ...
   "BootMode": "uefi"
   }
   ]
   ```

8. Launch a new instance using the newly-created AMI. All new instances created from this AMI will inherit the same boot mode.

9. To verify that the new instance has the expected boot mode, use the `describe-instances` command.

---

**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.
When you launch an instance from a Windows AMI, the root device for the Windows instance is an Amazon Elastic Block Store (Amazon EBS) volume. Windows AMIs do not support instance store for the root device.

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 Upgrade an Amazon EC2 Windows instance to a newer version of Windows Server (p. 631).

Windows AMI topics
- Select an initial Windows AMI (p. 28)
- Keep your AMIs up-to-date (p. 28)
- Virtualization types (p. 28)
- Managed AWS Windows AMIs (p. 29)
- Create a custom Windows AMI (p. 37)
- Deregister your Windows AMI (p. 51)
- Specialized Windows AMIs (p. 52)
- AWS Windows AMI Version History (p. 58)

Select 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 Find a Windows AMI (p. 97).

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

Keep 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. 30).

The AWS Windows AMIs contain the latest security updates available at the time they were created. For more information, see Patches, security updates, and AMI IDs (p. 30).

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.

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 and later)
- 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 AWS Windows AMIs (p. 27).

Contents

- Details about AWS Windows AMI versions (p. 30)
  - What to expect in an official AWS Windows AMI (p. 30)
  - How AWS decides which Windows AMIs to offer (p. 30)
  - Patches, security updates, and AMI IDs (p. 30)
  - Semiannual channel releases (p. 31)
- Configuration changes for AWS Windows AMIs (p. 31)
- Update your Windows instance (p. 33)
- Upgrade or migrate to a newer version of Windows Server (p. 34)
- Subscribe to Windows AMI notifications (p. 34)
- Changes in Windows Server 2016 and later AMIs (p. 35)
- Docker container conflict on Windows Server 2016 instances (p. 35)
- Issue with the Hibernate Agent (2018.03.16 AMIs) (p. 35)
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. For each of these images, AWS:

- Installs all Microsoft recommended Windows security patches. We release images shortly after the 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 Configure a Windows instance using the EC2Config service (p. 493) for Server 2012 R2 and earlier, or Configure a Windows instance using EC2Launch (p. 486) for Server 2016 and later.
- Configures Windows Time to use the Set the time for a Windows instance (p. 559).
- 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. The primary difference between Base and Core offerings is that Base offerings have a desktop/GUI whereas Core offerings are PowerShell command line only. For more information about Windows Server Core, see https://docs.microsoft.com/en-us/windows-server/administration/server-core/what-is-server-core.
- 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.

Note
Instances launched from the latest Windows Server 2019 AMIs may show a Windows Update dialog message stating “Some settings are managed by your organization.” This message appears as a result of changes in Windows Server 2019 and does not impact the behavior of Windows Update or your ability to manage update settings.
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 Semianual channel release versions of Windows Server.

### Configuration changes for AWS Windows AMIs

The following changes are applied to each AWS Windows AMI.

#### Clean and prepare

<table>
<thead>
<tr>
<th>Change</th>
<th>Applies to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check for pending file renames or reboots, and reboot as needed</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Delete .dmp files</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Delete logs (event logs, Systems Manager, EC2Config)</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Delete temporary folders and files for sysprep</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Clear recent history (Start menu, Windows Explorer, and so on)</td>
<td>Windows Server 2012 R2 and earlier</td>
</tr>
<tr>
<td>Perform virus scan</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Pre-compile queued .NET assemblies (before sysprep)</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Run Windows maintenance tools</td>
<td>Windows Server 2012 R2 and later</td>
</tr>
<tr>
<td>Restore default values for Internet Explorer</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Restore default values for EC2Config</td>
<td>Windows Server 2012 R2 and earlier</td>
</tr>
<tr>
<td>Set EC2Launch to run at the next launch</td>
<td>Windows Server 2016 and later</td>
</tr>
<tr>
<td>Reset the Windows wallpaper</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Run sysprep</td>
<td>All AMIs</td>
</tr>
</tbody>
</table>
## Install and configure

<table>
<thead>
<tr>
<th>Change</th>
<th>Applies to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add links to the Amazon EC2 Windows Guide</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Attach instance storage volumes to extended mount points</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Install the current AWS Tools for Windows PowerShell</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Install the current AWS CloudFormation helper scripts</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Install the current EC2Config and SSM Agent</td>
<td>Windows Server 2012 R2 and earlier</td>
</tr>
<tr>
<td>Install the current EC2Launch and SSM Agent</td>
<td>Windows Server 2016 and later</td>
</tr>
<tr>
<td>Install the current AWS PV, ENA, and NVMe drivers</td>
<td>Windows Server 2008 R2 and later</td>
</tr>
<tr>
<td>Install the current SRIOV drivers</td>
<td>Windows Server 2012 R2 and later</td>
</tr>
<tr>
<td>Install the current Citrix PV driver</td>
<td>Windows Server 2008 SP2 and earlier</td>
</tr>
<tr>
<td>Install the current EC2WinUtil driver</td>
<td>Windows Server 2008 R2 and later</td>
</tr>
<tr>
<td>Install PowerShell 2.0 and 3.0</td>
<td>Windows Server 2008 SP2 and R2</td>
</tr>
<tr>
<td>If Microsoft SQL Server is installed:</td>
<td>All AMIs</td>
</tr>
<tr>
<td>• Install service packs</td>
<td>All AMIs</td>
</tr>
<tr>
<td>• Configure to start automatically</td>
<td>All AMIs</td>
</tr>
<tr>
<td>• Add BUILTIN\Administrators to the SysAdmin role</td>
<td>All AMIs</td>
</tr>
<tr>
<td>• Open TCP port 1433 and UDP port 1434</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Apply the following hotfixes</td>
<td>Windows Server 2008 SP2 and R2</td>
</tr>
<tr>
<td>• MS15-011</td>
<td>Windows Server 2008 SP2 and R2</td>
</tr>
<tr>
<td>• KB2582281</td>
<td>Windows Server 2008 SP2 and R2</td>
</tr>
<tr>
<td>• KB2634328</td>
<td>Windows Server 2008 SP2 and R2</td>
</tr>
<tr>
<td>• KB2800213</td>
<td>Windows Server 2008 SP2 and R2</td>
</tr>
<tr>
<td>• KB2922223</td>
<td>Windows Server 2008 SP2 and R2</td>
</tr>
<tr>
<td>• KB2394911</td>
<td>Windows Server 2008 SP2 and R2</td>
</tr>
<tr>
<td>• KB2780879</td>
<td>Windows Server 2008 SP2 and R2</td>
</tr>
<tr>
<td>Allow ICMP traffic through the firewall</td>
<td>Windows Server 2012 R2 and earlier</td>
</tr>
<tr>
<td>Enable file and printer sharing</td>
<td>Windows Server 2012 R2 and earlier</td>
</tr>
<tr>
<td>Disable RunOnce for Internet Explorer</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Enable remote PowerShell</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Change</td>
<td>Applies to</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>Configure a paging file on the system volume as follows:</td>
<td>All AMIs</td>
</tr>
<tr>
<td>• Windows Server 2019 - Managed by the system</td>
<td></td>
</tr>
<tr>
<td>• Windows Server 2016 - Managed by the system</td>
<td></td>
</tr>
<tr>
<td>• Windows Server 2012 R2 - Initial size and max size are 8 GB</td>
<td>Windows Server 2012 R2 and earlier</td>
</tr>
<tr>
<td>• Windows Server 2012 and earlier - Initial size is 512 MB, max size</td>
<td></td>
</tr>
<tr>
<td>• Windows Server 2012 and earlier - Initial size is 8 GB</td>
<td></td>
</tr>
<tr>
<td>Configure an additional system managed paging file on Z:, if available</td>
<td>Windows Server 2012 R2 and earlier</td>
</tr>
<tr>
<td>Disable hibernation and delete the hibernation file</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Set the performance options for best performance</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Set the power setting to high performance</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Disable the screen saver password</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Set the RealTimeIsUniversal registry key</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Set the timezone to UTC</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Disable Windows updates and notifications</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Run Windows Update and reboot until there are no pending updates</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Set the display in all power schemes to never turn off</td>
<td>All AMIs</td>
</tr>
<tr>
<td>Set the PowerShell execution policy to &quot;Unrestricted&quot;</td>
<td>All AMIs</td>
</tr>
</tbody>
</table>

**Update 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 [Find a Windows AMI](p. 97).

**Note**
Instances should be stateless when updating. For more information, see [Managing Your AWS Infrastructure at Scale](p. 33).

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

- **Microsoft Windows Server**
- **Microsoft SQL Server**
- **Windows PowerShell**
- **Install the latest version of EC2Launch** (p. 487)
- **Install the latest version of EC2Config** (p. 494)
- **AWS Systems Manager SSM Agent**
- **Enable enhanced networking on Windows** (p. 963)
- **Install or upgrade AWS NVMe drivers** (p. 538)
- **Upgrade PV drivers on Windows instances** (p. 524)
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. 438).

Upgrade or migrate 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 Upgrade an Amazon EC2 Windows instance to a newer version of Windows Server (p. 631).

Subscribe 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

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):
   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

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.
Changes in Windows Server 2016 and later AMIs

AWS provides AMIs for Windows Server 2016 and later. 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 Configure a Windows instance using EC2Launch (p. 486).

- 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 and later AMIs, you can use the CloudWatch agent to configure integration with Amazon CloudWatch. For more information about configuring instances to send log data to CloudWatch, see Collect Metrics and Logs from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent. For information about joining an EC2 instance to a domain, see Join an Instance to a Domain Using the AWS-JoinDirectoryServiceDomain JSON Document in the AWS Systems Manager User Guide.

Other Differences

Note these additional important differences for instances created from Windows Server 2016 and later 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 Configure EC2Launch (p. 488).

- 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 and later AMIs.

For more information, see Windows Server 2019 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 Subscribe to Windows AMI notifications (p. 34).

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**

2. Use the following command to update the configuration, adding the missing quotation marks:

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

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

   ```bash
   (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:

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

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

   ```bash
   Get-Service EC2HibernateAgent
   ```

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

   ```bash
   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:

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

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

   ```bash
   sc qc EC2HibernateAgent
   ```

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

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

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

You can launch an instance from an existing Windows 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.

To help categorize and manage your AMIs, you can assign custom tags to them. For more information, see Tag your Amazon EC2 resources (p. 1450).

To create a custom Linux AMI, use the procedure for the type of volume for the instance. For more information, see Create an Amazon EBS-backed Linux AMI or Create an instance store-backed Linux AMI in the Amazon EC2 User Guide for Linux Instances.

Topics

- How the creation of a custom AMI works (p. 37)
- Create a Windows AMI from a running instance (p. 38)
- Create a standardized Amazon Machine Image (AMI) using Sysprep (p. 40)

How the creation of a custom AMI works

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 Deregister your Windows AMI (p. 51). 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. 1327).

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 Create Amazon EBS snapshots (p. 1211).

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.

Note

A Windows AMI must be created from an Amazon EC2 instance. Creation of a Windows AMI from an EBS snapshots is not supported.
If you add instance store volumes or Amazon Elastic Block Store (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 mappings (p. 1413).

**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 Initialize Amazon EBS volumes.

### Create 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/](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-2019`
   - `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 check box.
4. Choose **Launch**. Accept the default values as you step through the wizard. For more information, see Launch an instance using the Launch Instance Wizard (p. 392). When the instance is ready, connect to it. For more information, see Connect to your Windows instance (p. 413).
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 and later] Configure settings using EC2Launch. To generate a random password at launch time, use the `adminPasswordType` setting. For more information, see Configure EC2Launch (p. 488).
- [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. 500).
- [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. 527).

6. In the navigation pane, choose **Instances** and select your instance. Choose **Actions, Image and templates**, 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, 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 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 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 Launch an instance using the Launch Instance Wizard (p. 392). 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).

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 Access Amazon EC2 (p. 3).

- create-image (AWS CLI)
- New-EC2Image (AWS Tools for Windows PowerShell)

Create a standardized Amazon Machine Image (AMI) using Sysprep

The Microsoft System Preparation (Sysprep) tool simplifies the process of duplicating a customized installation of Windows. You can 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 recommend that you use EC2 Image Builder to automate the creation, management, and deployment of customized, secure, and up-to-date "golden" server images that are pre-installed and preconfigured with software and settings.

If you use Sysprep to create a standardized AMI, we recommend that you run Sysprep with EC2Launch v2 (p. 449). If you are still using the EC2Config (Windows Server 2012 R2 and earlier) or EC2Launch (Windows Server 2016 and later) agents, see the documentation for using Sysprep with EC2Config and EC2Launch below.

**Important**
Do not use Sysprep to create an instance backup. Sysprep removes system-specific information; removing this information might have unintended consequences for an instance backup.

To troubleshoot Sysprep, see Troubleshoot Sysprep (p. 1498).

Contents

- Before you begin (p. 40)
- Use Sysprep with EC2Launch v2 (p. 41)
- Use Sysprep with EC2Launch (p. 44)
- Use Sysprep with EC2Config (p. 47)

Before you begin

- Before performing Sysprep, we recommend that you remove all local user accounts and all account profiles other than a single administrator account under which Sysprep will be run. If you perform Sysprep with additional accounts and profiles, unexpected behavior could result, including loss of profile data or failure to complete Sysprep.
- Learn more about Sysprep on Microsoft TechNet.
- Learn which server roles are supported for Sysprep.
Use Sysprep with EC2Launch v2

This section contains details about the different Sysprep execution phases and the tasks performed by the EC2Launch v2 service as the image is prepared. It also includes the steps to create a standardized AMI using Sysprep with the EC2Launch v2 service.

Sysprep with EC2Launch v2 topics

- Sysprep phases (p. 41)
- Sysprep actions (p. 41)
- Post Sysprep (p. 43)
- Run Sysprep with EC2Launch v2 (p. 43)

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 EC2Launch v2 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 run commands in this phase.

- **Out-of-Box Experience (OOBE)**: The system runs an abbreviated version of Windows Setup and asks you to enter information such as system language, time zone, and registered organization. When you run Sysprep with EC2Launch v2, the answer file automates this phase.

Sysprep actions

Sysprep and EC2Launch v2 perform the following actions when preparing an image.

1. When you choose **Shutdown with Sysprep** in the **EC2Launch settings** dialog box, the system runs the ec2launch sysprep command.

2. EC2Launch v2 edits the content of the unattend.xml file by reading the registry value at HKEY_USERS\DEFAULT\Control Panel\International\LocaleName. This file is located in the following directory: C:\ProgramData\Amazon\EC2Launch\sysprep.

3. The system run the BeforeSysprep.cmd. This command creates a registry key as follows:

   ```cmd
   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 EC2Launch v2 service calls Sysprep by running the following command:

   ```cmd
   sysprep.exe /oobe /generalize /shutdown /unattend: "C:\ProgramData\Amazon\EC2Launch\sysprep\unattend.xml"
   ```
Generalize phase

- EC2Launch v2 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 `unattend.xml` answer file includes the following settings that 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 an SID. The system also performs the following actions based on configurations that you specify in the `unattend.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 make to that account are carried over to the new image. The default value is True.

  `CopyProfile` replaces the default profile with the existing local administrator profile. All accounts that you log in to after running Sysprep receive a copy of that profile and its contents at first login.
   
  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 runs the following command, which 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 runs 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. The system specifies the following configurations using the EC2Launch v2 answer file:
   - `<InputLocale>en-US</InputLocale>`
   - `<SystemLocale>en-US</SystemLocale>`
   - `<UILanguage>en-US</UILanguage>`
Create a custom Windows AMI

- `<UserLocale>en-US</UserLocale>`
- `<HideEULAPage>true</HideEULAPage>`
- `<HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>`
- `<ProtectYourPC>3</ProtectYourPC>`
- `<BluetoothTaskbarIconEnabled>false</BluetoothTaskbarIconEnabled>`
- `<TimeZone>UTC</TimeZone>`
- `<RegisteredOrganization>Amazon.com</RegisteredOrganization>`
- `<RegisteredOwner>EC2</RegisteredOwner>`

**Note**
During the generalize and specialize phases, EC2Launch v2 monitors the status of the OS. If EC2Launch v2 detects that the OS is in a Sysprep phase, then it publishes the following message to the system log:
Windows is being configured. SysprepState=IMAGE_STATE_UNDEPLOYABLE

2. The system runs EC2Launch v2.

**Post Sysprep**

After Sysprep completes, EC2Launch v2 sends the following message to the console output:

```
Windows sysprep configuration complete.
```

EC2Launch v2 then performs the following actions:

1. Reads the content of the `agent-config.yml` file and runs configured tasks.
2. Executes all tasks in the `preReady` stage.
3. After it is finished, sends a *Windows is ready* message to the instance system logs.
4. Executes all tasks in the `PostReady` stage.

For more information about EC2Launch v2, see Configure a Windows instance using EC2Launch v2 (p. 449).

**Run Sysprep with EC2Launch v2**

Use the following procedure to create a standardized AMI using Sysprep with EC2Launch v2.

1. In the Amazon EC2 console, locate or create (p. 37) an AMI that you want to duplicate.
2. Launch and connect to your Windows instance.
3. Customize it.
4. From the Windows *Start* menu, search for and choose *Amazon EC2Launch settings*. For more information about the options and settings in the Amazon *EC2Launch settings* dialog box, see EC2Launch v2 settings (p. 456).
5. Select *Shutdown with Sysprep* or *Shutdown without Sysprep*.

When you are asked to confirm that you want to run Sysprep and shut down the instance, click *Yes*. EC2Launch v2 runs Sysprep. Next, 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 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\ec2launch\ec2launch.exe" sysprep --shutdown=true

Use Sysprep with EC2Launch

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.

Sysprep with EC2Launch topics
- EC2Launch answer and batch files for Sysprep (p. 44)
- Run Sysprep with EC2Launch (p. 44)
- Update metadata/KMS routes for Server 2016 and later when launching a custom AMI (p. 47)

EC2Launch answer and batch files for Sysprep

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.

Run Sysprep with EC2Launch

On the full installation of Windows Server 2016 and later (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 or later 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.
Set Computer Name
- Set the computer name of the instance ip-<hex internal IP>. Disable this feature to persist your own computer name setting.

Set Wallpaper
- Overlay instance information on the current wallpaper.

Extend Boot Volume
- Extend OS partition to consume free space for boot volume.

Add DNS Suffix List
- Add DNS suffix list to allow DNS resolution of servers running in EC2 without providing the fully qualified domain name.

Handle User Data
- Execute user data provided at instance launch. Note: This will be re-enabled when running shutdown with sysprep below.

Administrator Password
- Random (Retrieve from console)
- Specify (Temporarily store in config file)
- Do Nothing (Customize Unattend.xml for sysprep)

These changes will take effect on next boot if Ec2Launch script is scheduled. By default, it is scheduled by shutdown options below.

Sysprep
- Sysprep is a Microsoft tool that prepares an image for multiple launches.
- Ec2Launch Script Location: Found
  - C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInsta
- Run EC2Launch on every boot (instead of just the next boot).
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 or later 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.
Update metadata/KMS routes for Server 2016 and later when launching a custom AMI

To update metadata/KMS routes for Server 2016 and later when launching a custom AMI, do one of the following:

- Run the EC2LaunchSettings GUI (C:\ProgramData\Amazon\EC2-Windows\Launch\Settings \Ec2LaunchSettings.exe) and select the option to shut down with Sysprep.
- Run EC2LaunchSettings and shut down without Sysprep before creating the AMI. This sets the EC2 Launch Initialize tasks to run at the next boot, which will set routes based on the subnet for the instance.
- Manually reschedule EC2 Launch initialize tasks before creating an AMI from PowerShell (p. 488).

**Important**
Take note of the default password reset behavior before rescheduling tasks.

- To update the routes on a running instance that is experiencing Windows activation or communication with instance metadata failures, see “Unable to activate Windows” (p. 1534).

Use Sysprep with EC2Config

This section contains details about the different Sysprep execution phases and the tasks performed by the EC2Config service as the image is prepared. It also includes the steps to create a standardized AMI using Sysprep with the EC2Config service.

**Sysprep with EC2Config topics**

- Sysprep phases (p. 41)
- Sysprep actions (p. 47)
- Post Sysprep (p. 50)
- Run Sysprep with the EC2Config service (p. 50)

**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 run 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 runs BeforeSysprep.cmd. This command creates a registry key as follows:

   ```cmd
   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:

   ```cmd
   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 that 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.

  **CopyProfile** replaces the default profile with the existing local administrator profile. All accounts logged into after running Sysprep will receive a copy of that profile and its contents at first login.
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 runs the following command that enables the administrator account and specifies the password requirement.

  ```
  net user Administrator /ACTIVE:YES /LOGONPASSWORDDCHG: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 runs 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 runs `SetupComplete.cmd` from the following location: `C:\Windows\Setup\Scripts\SetupComplete.cmd`. In Amazon public AMIs before April 2015 this file was empty and ran 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 runs `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.
   - Amazon CloudWatch Logs
   - UserData
   - AWS Systems Manager (Systems Manager)

For more information about Windows plug-ins, see Configure a Windows instance using the EC2Config service (p. 493).

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. 37) 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. 493).
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. 500)](#). For an example of configuring and then running Sysprep from the command line, see Ec2ConfigService\Scripts \InstallUpdates.ps1.

**Deregister 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 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 Instance state, Terminate instance. When prompted for confirmation, choose Terminate.

Specialized Windows AMIs

This section contains information about specialized Windows AMIs and Windows AMIs developed for Microsoft workload solutions.

Topics
SQL Server AMIs provided by AWS

For SQL Server license-included AMIs, use the installation and setup media included in C:\SQLServerSetup to make changes to the default installation, add new features, or install additional named instances.

Amazon EC2 Windows Server AMIs for STIG compliance

Security Technical Implementation Guides (STIGs) are the configuration standards created by the Defense Information Systems Agency (DISA) to secure information systems and software. To make your systems compliant with STIG standards, you must install, configure, and test a variety of security settings. Amazon EC2 Windows Server AMIs for STIG Compliance are pre-configured with over 160 required security settings. STIG-compliant operating systems include Windows Server 2012 R2, Windows Server 2016, and Windows Server 2019. The STIG-compliant AMIs include updated Department of Defense (DoD) certificates to help you get started and achieve STIG compliance. There are no additional charges for using STIG-compliant AMIs.

Amazon EC2 Windows Server AMIs for STIG compliance are available in all public AWS and GovCloud Regions. You can launch instances from these AMIs directly from the Amazon EC2 console. They are billed using standard Windows pricing.

The STIG-compliant Amazon EC2 AMIs for Windows Server can be found in the Community AMIs when you create an instance. The AMI names are as follows:

**Note**
The date suffix for the AMI (YYYY.MM.DD) is the date on which the latest version was created. You can search for the version without the date suffix.

- Windows_Server-2019-English-STIG-Core-YYYY.MM.DD
- Windows_Server-2016-English-STIG-Full-YYYY.MM.DD
- Windows_Server-2016-English-STIG-Core-YYYY.MM.DD
- Windows_Server-2012-R2-English-STIG-Full-YYYY.MM.DD
- Windows_Server-2012-R2-English-STIG-Core-YYYY.MM.DD

Compliance levels

- **High (Category I)**
  The most severe risk. Includes any vulnerability that can result in loss of confidentiality, availability, or integrity.
- **Medium (Category II)**
  Includes any vulnerability that could result in loss of confidentiality, availability, or integrity but where the risks could be mitigated.
- **Low (Category III)**
  Includes any vulnerability that degrades measures to protect against loss of confidentiality, availability, or integrity.

The following sections show the STIGs that have been applied to Windows Operating Systems and components.
Topics

- Core and base operating systems (p. 54)
- Microsoft .NET Framework 4.0 STIG V2 Release 1 (p. 56)
- Windows Firewall STIG V1 Release 7 (p. 56)
- Internet Explorer (IE) 11 STIG V1 Release 19 (p. 57)
- Version history (p. 57)

Core and base operating systems

STIG-compliant Amazon EC2 AMIs are designed for use as standalone servers, and have the highest level of STIG settings applied (Category 1). Each compliance level includes all STIG settings from lower levels, which means that the highest level has all applicable settings from all levels.

Some STIG settings are not automatically applied. This can be due to technical limitations – for instance, the STIG setting might not be applicable for standalone servers. Organization-specific policies can also prevent automatic application of STIG settings, such as a requirement for administrators to review document settings. For more details about which STIGs are applied to Amazon EC2 Windows AMIs, you can download our spreadsheet.

For a complete list of Windows STIGs, see the STIGs Document Library. For information about how to view the complete list, see How to View SRGs and STIGs.

Windows Server 2019 STIG V2 Release 2

All of the following STIG settings for Windows operating systems are applied:

- **Windows\Low**
  - V-205691, V-205819, V-205858, V-205859, V-205860, V-205870, V-205871, and V-205923

- **Windows\Medium**

- **Windows\High**
Amazon Elastic Compute Cloud
User Guide for Windows Instances
Specialized Windows AMIs

Windows Server 2016 STIG V2 Release 2
All of the following STIG settings for Windows operating systems are applied:
• Windows\Low
V-224916, V-224917, V-224918, V-224919, V-224931, V-224942, and V-225060
• Windows\Medium
V-224850, V-224852, V-224853, V-224854, V-224855, V-224856, V-224857, V-224858, V-224859,
V-224866, V-224867, V-224868, V-224869, V-224870, V-224871, V-224872, V-224873, V-224881,
V-224882, V-224883, V-224884, V-224885, V-224886, V-224887, V-224888, V-224889, V-224890,
V-224891, V-224892, V-224893, V-224894, V-224895, V-224896, V-224897, V-224898, V-224899,
V-224900, V-224901, V-224902, V-224903, V-224904, V-224905, V-224906, V-224907, V-224908,
V-224909, V-224910, V-224911, V-224912, V-224913, V-224914, V-224915, V-224920, V-224922,
V-224924, V-224925, V-224926, V-224927, V-224928, V-224929, V-224930, V-224935, V-224936,
V-224937, V-224938, V-224939, V-224940, V-224941, V-224943, V-224944, V-224945, V-224946,
V-224947, V-224948, V-224949, V-224951, V-224952, V-224953, V-224955, V-224956, V-224957,
V-224959, V-224960, V-224962, V-224963, V-225010, V-225013, V-225014, V-225015, V-225016,
V-225017, V-225018, V-225019, V-225021, V-225022, V-225023, V-225024, V-225028, V-225029,
V-225030, V-225031, V-225032, V-225033, V-225034, V-225035, V-225038, V-225039, V-225040,
V-225041, V-225042, V-225043, V-225047, V-225049, V-225050, V-225051, V-225052, V-225055,
V-225056, V-225057, V-225058, V-225061, V-225062, V-225063, V-225064, V-225065, V-225066,
V-225067, V-225068, V-225069, V-225072, V-225073, V-225074, V-225076, V-225078, V-225080,
V-225081, V-225082, V-225083, V-225084, V-225086, V-225087, V-225088, V-225089, V-225092,
V-225093, and V-236000
• Windows\High
V-224874, V-224932, V-224933, V-224934, V-224954, V-224958, V-224961, V-225025, V-225044,
V-225045, V-225046, V-225048, V-225053, V-225054, and V-225079

Windows Server 2012 R2 STIG V3 Release 2
All of the following STIG settings for Windows operating systems are applied:
• Windows\Low
V-225537, V-225536, V-225526, V-225525, V-225514, V-225511, V-225490, V-225489, V-225488,
V-225487, V-225485, V-225484, V-225483, V-225482, V-225481, V-225480, V-225479, V-225476,
V-225473, V-225468, V-225462, V-225460, V-225459, V-225412, V-225394, V-225392, V-225376,
V-225363, V-225362, V-225360, V-225359, V-225358, V-225357, V-225355, V-225343, V-225342,
V-225336, V-225335, V-225334, V-225333, V-225332, V-225331, V-225330, V-225328, V-225327,
V-225324, V-225319, V-225318, and V-225250
• Windows\Medium
V-225574, V-225573, V-225572, V-225571, V-225570, V-225569, V-225568, V-225567, V-225566,
V-225565, V-225564, V-225563, V-225562, V-225561, V-225560, V-225559, V-225558, V-225557,
V-225555, V-225554, V-225553, V-225551, V-225550, V-225549, V-225548, V-225546, V-225545,
V-225544, V-225543, V-225542, V-225541, V-225540, V-225539, V-225538, V-225535, V-225534,
V-225533, V-225532, V-225531, V-225530, V-225529, V-225528, V-225527, V-225524, V-225523,
V-225522, V-225521, V-225520, V-225519, V-225518, V-225517, V-225516, V-225515, V-225513,
V-225510, V-225509, V-225508, V-225506, V-225504, V-225503, V-225502, V-225501, V-225500,
V-225494, V-225486, V-225478, V-225477, V-225475, V-225474, V-225472, V-225471, V-225470,
V-225469, V-225464, V-225463, V-225461, V-225458, V-225457, V-225456, V-225455, V-225454,
V-225453, V-225452, V-225448, V-225443, V-225442, V-225441, V-225415, V-225414, V-225413,
V-225411, V-225410, V-225409, V-225408, V-225407, V-225406, V-225405, V-225404, V-225402,
V-225401, V-225400, V-225398, V-225397, V-225395, V-225393, V-225391, V-225389, V-225386,
55



- **Windows\High**


**Microsoft .NET Framework 4.0 STIG V2 Release 1**

The following list contains STIG settings that are applied to Windows operating system components for STIG-compliant Amazon EC2 AMIs. Some STIG settings are not automatically applied. This can be due to technical limitations—for instance, the STIG setting might not be applicable for standalone servers. Organization-specific policies can also prevent automatic application of STIG settings, such as a requirement for administrators to review document settings. For more details about which STIGs are applied to Amazon EC2 Windows AMIs, you can download our spreadsheet.

For a complete list of Windows STIGs, see the STIGs Document Library. For information about how to view the complete list, see How to View SRGs and STIGs.

- **Windows\Low**

  No STIG settings are applied to the Microsoft .NET Framework for Category III vulnerabilities.

- **Windows\Medium**

  V-225238

- **Windows\High**

  No additional STIG settings are applied for Category I vulnerabilities.

**Windows Firewall STIG V1 Release 7**

The following list contains STIG settings that are applied to Windows operating system components for STIG-compliant Amazon EC2 AMIs. Some STIG settings are not automatically applied. This can be due to technical limitations—for instance, the STIG setting might not be applicable for standalone servers. Organization-specific policies can also prevent automatic application of STIG settings, such as a requirement for administrators to review document settings. For more details about which STIGs are applied to Amazon EC2 Windows AMIs, you can download our spreadsheet.

For a complete list of Windows STIGs, see the STIGs Document Library. For information about how to view the complete list, see How to View SRGs and STIGs.

- **Windows\Low**

  V-17425, V-17426, V-17427, V-17435, V-17436, V-17437, V-17445, V-17446, and V-17447

- **Windows\Medium**
Internet Explorer (IE) 11 STIG V1 Release 19

The following list contains STIG settings that are applied to Windows operating system components for STIG-compliant Amazon EC2 AMIs. Some STIG settings are not automatically applied. This can be due to technical limitations – for instance, the STIG setting might not be applicable for standalone servers. Organization-specific policies can also prevent automatic application of STIG settings, such as a requirement for administrators to review document settings. For more details about which STIGs are applied to Amazon EC2 Windows AMIs, you can download our spreadsheet.

For a complete list of Windows STIGs, see the STIGs Document Library. For information about how to view the complete list, see How to View SRGs and STIGs.

- **Windows\High**
  - V-17418, V-17428, and V-17438

- **Windows\Low**
  - V-46477, V-46629, and V-97527

- **Windows\Medium**

- **Windows\High**
  - No additional STIG settings are applied for Category I vulnerabilities.

Version history

The following table shows version history updates for STIG settings that are applied to Windows operating systems and Windows components.

<table>
<thead>
<tr>
<th>Date</th>
<th>AMIs</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/9/2021</td>
<td>Windows Server 2019 STIG V2 R2</td>
<td>Updated versions where applicable, and applied STIGs.</td>
</tr>
<tr>
<td></td>
<td>Windows Server 2016 STIG V2 R2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Windows Server 2012 R2 STIG V3 R2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Microsoft .NET Framework 4.0 STIG V2 R1</td>
<td></td>
</tr>
</tbody>
</table>
### AWS Windows AMI Version History

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 2021 (to date) (p. 59)
- Monthly AMI updates for 2020 (to date) (p. 64)
- Monthly AMI updates for 2019 (p. 69)

<table>
<thead>
<tr>
<th>Date</th>
<th>AMIs</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/5/2021</td>
<td>Windows Server 2019 STIG V2 R 1&lt;br&gt;Windows Server 2016 STIG V2 R 1&lt;br&gt;Windows Server 2012 R2 STIG V3 R 1&lt;br&gt;Microsoft .NET Framework 4.0 STIG V2 R 1&lt;br&gt;Windows Firewall STIG V1 R 7&lt;br&gt;Internet Explorer 11 STIG V1 R 19</td>
<td>Updated versions where applicable, and applied STIGs.</td>
</tr>
<tr>
<td>9/18/2020</td>
<td>Windows Server 2019 STIG V1 R 5&lt;br&gt;Windows Server 2016 STIG V1 R 12&lt;br&gt;Windows Server 2012 R2 STIG V2 R 19&lt;br&gt;Internet Explorer 11 STIG V1 R 19&lt;br&gt;Microsoft .NET Framework 4.0 STIG V1 R 9&lt;br&gt;Windows Firewall STIG V1 R 7</td>
<td>Updated versions and applied STIGs.</td>
</tr>
<tr>
<td>12/6/2019</td>
<td>Server 2012 R2 Core and Base V2 R17&lt;br&gt;Server 2016 Core and Base V1 R11&lt;br&gt;Internet Explorer 11 V1 R18&lt;br&gt;Microsoft .NET Framework 4.0 V1 R9&lt;br&gt;Windows Firewall STIG V1 R17</td>
<td>Updated versions and applied STIGs.</td>
</tr>
<tr>
<td>9/17/2019</td>
<td>Server 2012 R2 Core and Base V2 R16&lt;br&gt;Server 2016 Core and Base V1 R9&lt;br&gt;Server 2019 Core and Base V1 R2&lt;br&gt;Internet Explorer 11 V1 R17&lt;br&gt;Microsoft .NET Framework 4.0 V1 R8</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>
• Monthly AMI updates for 2018 (p. 75)
• Monthly AMI updates for 2017 (p. 81)
• Monthly AMI updates for 2016 (p. 86)
• Monthly AMI updates for 2015 (p. 89)
• Monthly AMI updates for 2014 (p. 92)
• Monthly AMI updates for 2013 (p. 93)
• Monthly AMI updates for 2012 (p. 95)
• Monthly AMI updates for 2011 and earlier (p. 96)

For more information about components included in these AMIs, see the following:
• EC2Config version history (p. 506)
• EC2Launch version history
• Systems Manager SSM Agent Release Notes
• Amazon ENA driver versions
• AWS PV drivers

**Monthly AMI updates for 2021 (to date)**

For more information about Microsoft updates, see Description of Software Update Services and Windows Server Update Services changes in content for 2021.

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021.08.11</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to August 10th, 2021</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.13571</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch version 1.3.2003411</td>
</tr>
<tr>
<td></td>
<td>• SSM version 3.0.1181.0</td>
</tr>
<tr>
<td></td>
<td>• SQL Server CUs installed:</td>
</tr>
<tr>
<td></td>
<td>• SQL_2019: CU11</td>
</tr>
<tr>
<td></td>
<td><strong>EC2LaunchV2 Preview AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• EC2Launch v2 version 2.0.548</td>
</tr>
<tr>
<td></td>
<td>Previous versions of Amazon-published Windows AMIs dated April 14th, 2021 and earlier were made private.</td>
</tr>
<tr>
<td>2021.07.14</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to July 13th, 2021</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.1350</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch version 1.3.2003364</td>
</tr>
<tr>
<td></td>
<td>• SQL Server CUs installed:</td>
</tr>
<tr>
<td></td>
<td>• SQL_2017: CU24</td>
</tr>
<tr>
<td>2021.07.07</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>Out-of-band AMI release that applies the July out-of-band security update recently released by Microsoft as an additional mitigation to CVE-34527.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\Printers\PointAndPrint is not defined on AWS-provided windows AMIs, which is the default state.</td>
</tr>
<tr>
<td></td>
<td>For more information, see:</td>
</tr>
<tr>
<td></td>
<td>• <a href="https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-34527">https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-34527</a></td>
</tr>
<tr>
<td></td>
<td>• <a href="https://support.microsoft.com/en-us/topic/kb5005010-restricting-installation-of-new-printer-drivers-after-applying-the-july-6-2021-updates-31b91c02-05bc-4ada-a7ea-183b129578a7">https://support.microsoft.com/en-us/topic/kb5005010-restricting-installation-of-new-printer-drivers-after-applying-the-july-6-2021-updates-31b91c02-05bc-4ada-a7ea-183b129578a7</a></td>
</tr>
<tr>
<td></td>
<td>Previous versions of Amazon-published Windows AMIs dated March 10th, 2021 and earlier were made private.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2021.06.09</th>
<th><strong>All AMIs</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Windows security updates current to June 8th, 2021</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.1326</td>
</tr>
<tr>
<td></td>
<td>• SSM version 3.0.1124.0</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2012RTM/2012 R2 AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.4419</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2021.05.12</th>
<th><strong>All AMIs</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Windows security updates current to May 11th, 2021</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.1302</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch version 1.3.2003312</td>
</tr>
<tr>
<td></td>
<td>• SQL Server CUs installed:</td>
</tr>
<tr>
<td></td>
<td>• SQL_2019: CU10</td>
</tr>
<tr>
<td></td>
<td>• Previous versions of Amazon-published Windows AMIs dated February 10th, 2021 and earlier were made private.</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2012RTM/2012 R2 AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.4381</td>
</tr>
<tr>
<td></td>
<td>• SSM version 3.0.529.0</td>
</tr>
<tr>
<td></td>
<td><strong>NVIDIA GPU AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• GRID version 462.31</td>
</tr>
<tr>
<td></td>
<td>• Tesla version 462.31</td>
</tr>
<tr>
<td></td>
<td><strong>Radeon GPU AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Radeon version 20.10.25.04</td>
</tr>
</tbody>
</table>
## AWS Windows AMI Version History

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021.04.14</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to April 13th, 2021</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.1280</td>
</tr>
<tr>
<td></td>
<td>• AWS PV version 8.4.0</td>
</tr>
<tr>
<td></td>
<td>• cfn-init version 2.0.6. This package includes Microsoft Visual C++ 2015-2019 Redistributable version 14.28.29913.0 as a dependency.</td>
</tr>
<tr>
<td></td>
<td>• AWS ENA version 2.2.3</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch version 1.3.2003284</td>
</tr>
<tr>
<td></td>
<td>• SQL Server CUs installed:</td>
</tr>
<tr>
<td></td>
<td>• SQL_2017: CU23</td>
</tr>
<tr>
<td></td>
<td>• Previous versions of Amazon-published Windows AMIs dated January 13th, 2021 and earlier were made private.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>All public versions of the following images will be made private on</td>
</tr>
<tr>
<td></td>
<td>May 11th, 2021. Existing instances and custom images owned by your</td>
</tr>
<tr>
<td></td>
<td>account that are based on Windows Server 1909 will not be impacted.</td>
</tr>
<tr>
<td></td>
<td>To retain access to Windows Server 1909, create a custom image in your</td>
</tr>
<tr>
<td></td>
<td>account prior to May 11th, 2021.</td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-1909-English-Core-Base</td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-1909-English-Core-ContainersLatest</td>
</tr>
<tr>
<td></td>
<td><strong>EC2LaunchV2_Preview AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• EC2Launch v2 version 2.0.285</td>
</tr>
</tbody>
</table>
## AWS Windows AMI Version History

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021.03.11</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to March 9th, 2021</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.1248</td>
</tr>
<tr>
<td></td>
<td>• cfn-init version 2.0.5. This package includes Microsoft Visual C++ 2015-2019 Redistributable version 14.28.29910.0 as a dependency.</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch version 1.3.2003236</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent version 3.0.529.0</td>
</tr>
<tr>
<td></td>
<td>• NVIDIA GRID version 461.33</td>
</tr>
<tr>
<td></td>
<td>• SQL Server CUs installed:</td>
</tr>
<tr>
<td></td>
<td>• SQL 2016_SP2: CU16</td>
</tr>
<tr>
<td></td>
<td>• SQL 2019: CU9</td>
</tr>
<tr>
<td></td>
<td>• KB4577586 update for the removal of Adobe Flash Player installed on all applicable images (Adobe Flash player is not enabled by default on all images).</td>
</tr>
</tbody>
</table>

**Note**

Amazon Root CAs have been added to the Trusted Root Certification Authorities certificate store on all AMIs. For more information, see [https://www.amazontrust.com/repository/#rootcas](https://www.amazontrust.com/repository/#rootcas).

**Windows Server 2016 and 2019 AMIs**

• Updated from default .NET framework versions to version 4.8.

**Windows Server 2012RTM/2012 R2 AMIs**

• EC2Config version 4.9.4326
• SSM Agent version 3.0.431.0
## AWS Windows AMI Version History

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021.02.10</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to February 9th, 2021</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.1224</td>
</tr>
<tr>
<td></td>
<td>• NVIDIA GRID version 461.09</td>
</tr>
</tbody>
</table>

Beginning in March 2021, AWS-provided Windows AMIs include Amazon Root CAs in the certificate store to minimize potential disruption from the upcoming S3 and CloudFront certificate migration, which is scheduled for March 23rd, 2021. For more information, see the following:

- [https://forums.aws.amazon.com/ann.jspa?annID=7541](https://forums.aws.amazon.com/ann.jspa?annID=7541)

Additionally, AWS will apply "update for Removal of Adobe Flash Player" (KB4577586) to all Windows AMIs in March to remove the built-in Adobe Flash player, which ended support on December 31st, 2020. If your use case requires the built-in Adobe Flash player, we recommend creating a custom image based on AMIs with version 2021.02.10 or earlier. For more information on the End of Support of Adobe Flash Player, see:


**EC2LaunchV2_Preview AMIs**

- EC2Launch v2 version 2.0.207

**New Windows AMIs**

- Windows_Server-2016-Japanese-Full-SQL_2019_Enterprise-2021.02.10
- Windows_Server-2016-Japanese-Full-SQL_2019_Standard-2021.02.10
- Windows_Server-2016-Japanese-Full-SQL_2019_Web-2021.02.10

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021.01.13</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to January 12th, 2021</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.1204</td>
</tr>
<tr>
<td></td>
<td>• AWS ENA version 2.2.2</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch v1 version 1.3.2003210</td>
</tr>
</tbody>
</table>

**Windows Server SAC/2019/2016 AMIs**

- SSM Agent version 3.0.431.0
## Monthly AMI updates for 2020 (to date)

For more information about Microsoft updates, see Description of Software Update Services and Windows Server Update Services changes in content for 2020.

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020.12.09</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to December 8th, 2020</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.1181</td>
</tr>
<tr>
<td></td>
<td>• All SQL Server Enterprise, Standard, and Web AMIs now include SQL Server installation media at C: \SQLServerSetup</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch v1 version 1.3.2003189</td>
</tr>
<tr>
<td></td>
<td>• Previous versions of Amazon-published Windows AMIs dated September 9th, 2020 and earlier were made private.</td>
</tr>
</tbody>
</table>

**Windows Server 2012/2012 R2 AMIs**

- EC2Config version 4.9.4279
- SSM Agent version 2.3.871.0

**EC2LaunchV2_Preview AMIs**

- EC2Launch v2 version 2.0.160

<table>
<thead>
<tr>
<th>2020.11.11</th>
<th><strong>All AMIs</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Windows security updates current to November 10th, 2020</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.1160</td>
</tr>
<tr>
<td></td>
<td>• NVIDIA GRID version 452.39</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch v2 Preview AMIs: EC2Launch version 2.0.153</td>
</tr>
<tr>
<td></td>
<td>• Previous versions of Amazon-published Windows AMIs dated August 12th, 2020 and earlier were made private.</td>
</tr>
</tbody>
</table>

**New Windows AMIs**

- Windows_Server-20H2-English-Core-Base-2020.11.11
- Windows_Server-20H2-English-Core-ContainersLatest-2020.11.11

<table>
<thead>
<tr>
<th>2020.10.14</th>
<th><strong>All AMIs</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Windows security updates current to October 13th, 2020</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.1140</td>
</tr>
<tr>
<td></td>
<td>• NVIDIA GRID version 452.39</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch v2 Preview AMIs: EC2Launch version 2.0.146</td>
</tr>
<tr>
<td></td>
<td>• AWS ENA version 2.2.1</td>
</tr>
</tbody>
</table>
Amazon Elastic Compute Cloud
User Guide for Windows Instances
AWS Windows AMI Version History

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020.9.25</td>
<td>A new version of Amazon Machine Images with SQL Server 2019 dated 2020.09.25 has been released. This release includes the same software components as the previous release dated 2020.09.09 but does not include CU7 for SQL 2019, which has recently been removed from public availability by Microsoft due to a known issue with reliability of the database snapshot feature. For more information, please see the following Microsoft blog post: <a href="https://techcommunity.microsoft.com/t5/sql-server/cumulative-update-7-for-sql-server-2019-rtm-removed/ba-p/1629317">https://techcommunity.microsoft.com/t5/sql-server/cumulative-update-7-for-sql-server-2019-rtm-removed/ba-p/1629317</a>.</td>
</tr>
</tbody>
</table>

**New Windows AMIs**

- Windows_Server-2016-English-Full-SQL_2019_Enterprise-2020.09.25
- Windows_Server-2016-English-Full-SQL_2019_Express-2020.09.25
- Windows_Server-2016-English-Full-SQL_2019_Web-2020.09.25

**EC2LaunchV2 Preview AMIs**


**2020.9.9**

**All AMIs**

- Windows security updates current to September 8th, 2020
- AWS PV drivers version 8.3.4
- AWS ENA version 2.2.0
- AWS Tools for Windows PowerShell version 3.15.1110
- SQL Server CU installed
  - SQL_2016_SP2: CU14
  - SQL_2019: CU7
- Previous versions of Amazon-published Windows AMIs dated June 10th, 2020 and earlier were made private.


- EC2Launch version 1.3.2003155
- SSM Agent version 2.3.1319.0

**EC2LaunchV2 Preview AMIs**

- EC2Launch v2 version 2.0.124
<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020.8.12</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to August 11th, 2020</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.1084</td>
</tr>
<tr>
<td></td>
<td>• G3 AMIs: NVIDIA GRID version 451.48</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch v2 Preview AMIs: EC2Launch version 2.0.104</td>
</tr>
<tr>
<td></td>
<td>• SQL CUs installed</td>
</tr>
<tr>
<td></td>
<td>• SQL_2019: CU6</td>
</tr>
<tr>
<td></td>
<td>• Previous versions of Amazon-published Windows AMIs dated May 13th, 2020 and earlier were made private.</td>
</tr>
<tr>
<td>2020.7.15</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to July 14th, 2020</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.1064</td>
</tr>
<tr>
<td></td>
<td>• ENA version 2.1.5</td>
</tr>
<tr>
<td></td>
<td>• SQL Server CUs installed</td>
</tr>
<tr>
<td></td>
<td>• SQL_2017: CU21</td>
</tr>
<tr>
<td></td>
<td>• SQL_2019: CU5</td>
</tr>
<tr>
<td></td>
<td>• Previous versions of Amazon-published Windows AMIs dated April 15th, 2020 and earlier were made private.</td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2020.7.01</td>
<td>A new version of Amazon Machine Images has been released. These images include EC2Launch v2 and serve as a functional preview of the new launch agent in advance of it being included by default on all Windows AMIs currently provided by AWS later this year. Note that some SSM documents and dependent services, such as EC2 Image Builder, may require updates to support EC2 Launch v2. These updates will follow in the coming weeks. These images are not recommended for use in production environments. You can read more about EC2Launch v2 at <a href="https://aws.amazon.com/about-aws/whats-new/2020/07/introducing-ec2-launch-v2-simplify-customizing-windows-instances/">https://aws.amazon.com/about-aws/whats-new/2020/07/introducing-ec2-launch-v2-simplify-customizing-windows-instances/</a> and Configure a Windows instance using EC2Launch v2 (p. 449). All current Windows Server AMIs will continue to be provided without changes to the current launch agent, either EC2Config (Server 2012 RTM or 2012 R2) or EC2Launch v1 (Server 2016 or later), for the next several months. In the near future, all Windows Server AMIs currently provided by AWS will be migrated to use EC2Launch v2 by default as part of the monthly release. EC2LaunchV2_Preview AMIs will be updated monthly and remain available until this migration occurs. New Windows AMIs</td>
</tr>
<tr>
<td></td>
<td>• EC2LaunchV2_Preview-Windows_Server-2004-English-Core-Base-2020.06.30</td>
</tr>
<tr>
<td></td>
<td>• EC2LaunchV2_Preview-Windows_Server-2019-English-Full-Base-2020.06.30</td>
</tr>
<tr>
<td></td>
<td>• EC2LaunchV2_Preview-Windows_Server-2019-English-Core-Base-2020.06.30</td>
</tr>
<tr>
<td></td>
<td>• EC2LaunchV2_Preview-Windows_Server-2016-English-Full-Base-2020.06.30</td>
</tr>
<tr>
<td></td>
<td>• EC2LaunchV2_Preview-Windows_Server-2016-English-Core-Base-2020.06.30</td>
</tr>
<tr>
<td></td>
<td>• EC2LaunchV2_Preview-Windows_Server-2012_R2_RTM-English-Full-Base-2020.06.30</td>
</tr>
<tr>
<td></td>
<td>• EC2LaunchV2_Preview-Windows_Server-2012_R2_RTM-English-Core-Base-2020.06.30</td>
</tr>
<tr>
<td></td>
<td>• EC2LaunchV2_Preview-Windows_Server-2012_RTM-English-Full-Base-2020.06.30</td>
</tr>
<tr>
<td></td>
<td>• EC2LaunchV2_Preview-Windows_Server-2019-English-Full-SQL_2019_Express-2020.06.30</td>
</tr>
<tr>
<td></td>
<td>• EC2LaunchV2_Preview-Windows_Server-2016-English-Full-SQL_2017_Express-2020.06.30</td>
</tr>
<tr>
<td>2020.6.10</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to June 9th, 2020</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.1034</td>
</tr>
<tr>
<td></td>
<td>• cfn-init version 1.4.33</td>
</tr>
<tr>
<td></td>
<td>• SQL CU installed: SQL_2016_SP2: CU13</td>
</tr>
<tr>
<td>2020.5.27</td>
<td><strong>New Windows AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2004-English-Core-Base-2020.05.27</td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2004-English-Core-ContainersLatest-2020.05.27</td>
</tr>
<tr>
<td>2020.5.13</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to May 12th, 2020</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.1013</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch version 1.3.2003150</td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>2020.4.15</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to April 14th, 2020</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.998</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.4222</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch version 1.3.2003040</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent version 2.3.842.0</td>
</tr>
<tr>
<td></td>
<td>• SQL Server CUs installed:</td>
</tr>
<tr>
<td></td>
<td>• SQL_2017: CU 20</td>
</tr>
<tr>
<td></td>
<td>• SQL_2019: CU 4</td>
</tr>
<tr>
<td>2020.3.18</td>
<td><strong>Windows Server 2019 AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>Resolves an intermittent issue discovered in the 2020.3.11 release in which the Background Intelligent Transfer Service (BITS) may not start within the expected time after initial OS boot, potentially resulting in timeouts, BITS errors in the event log, or failures of cmdlets involving BITS invoked quickly after the initial boot. Other Windows Server AMIs are not affected by this issue, and their latest version remains 2020.03.11.</td>
</tr>
<tr>
<td>2020.3.11</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to March 10th, 2020</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.969</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.4122</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch version 1.3.2002730</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent version 2.3.814.0</td>
</tr>
<tr>
<td></td>
<td>• SQL Server CUs installed:</td>
</tr>
<tr>
<td></td>
<td>• SQL_2016_SP2: CU 12</td>
</tr>
<tr>
<td></td>
<td>• SQL_2017: CU 19</td>
</tr>
<tr>
<td></td>
<td>• SQL_2019: CU 2 not applied due to known issue with SQL Agent</td>
</tr>
<tr>
<td></td>
<td>• Out of band security update (KB4551762) for server core 1909 and 1903 applied to mitigate CVE-2020-0796. Other Windows Server versions are not impacted by this issue. For details, see <a href="https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-0796">https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-0796</a></td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2020.2.12</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to February 11th, 2020</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.945</td>
</tr>
<tr>
<td></td>
<td>• Intel SRIOV driver updates</td>
</tr>
<tr>
<td></td>
<td>• 2019/1903/1909: version 2.1.185.0</td>
</tr>
<tr>
<td></td>
<td>• 2016/1809: version 2.1.186.0</td>
</tr>
<tr>
<td></td>
<td>• 2012 R2: version 1.2.199.0</td>
</tr>
<tr>
<td></td>
<td>• SQL Server CUs installed:</td>
</tr>
<tr>
<td></td>
<td>• SQL_2019: CU 1</td>
</tr>
<tr>
<td></td>
<td>• SQL_2017: CU 18</td>
</tr>
<tr>
<td></td>
<td>• SQL_2016_SP2: CU 11</td>
</tr>
<tr>
<td></td>
<td><strong>Microsoft Windows Server 2008 SP2 and Windows Server 2008 R2</strong></td>
</tr>
<tr>
<td></td>
<td>Windows Server 2008 SP2 and Window Server 2008 R2 reached End of Support (EOS) on 01/14/20 and will no longer receive regular security updates from Microsoft. AWS will no longer publish or distribute Windows Server 2008 SP2 or Windows Server 2008 R2 AMIs. Existing 2008 SP2/R2 instances and custom AMIs in your account are not impacted, and you can continue to use them after the EOS date. For more information about Microsoft End of Service on AWS, including upgrade and import options, as well as a full list of AMIs that are no longer published as of 01/14/2020, see <a href="#">End of Support (EOS) for Microsoft Products</a>.</td>
</tr>
<tr>
<td>2020.1.15</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to January 14, 2020</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.925</td>
</tr>
<tr>
<td></td>
<td>• ENA version 2.1.4</td>
</tr>
<tr>
<td></td>
<td><strong>Microsoft Windows Server 2008 SP2 and Windows Server 2008 R2</strong></td>
</tr>
<tr>
<td></td>
<td>Windows Server 2008 SP2 and Window Server 2008 R2 reached End of Support (EOS) on 01/14/20 and will no longer receive regular security updates from Microsoft. AWS will no longer publish or distribute Windows Server 2008 SP2 or Windows Server 2008 R2 AMIs. Existing 2008 SP2/R2 instances and custom AMIs in your account are not impacted, and you can continue to use them after the EOS date. For more information about Microsoft End of Service on AWS, including upgrade and import options, as well as a full list of AMIs that are no longer published as of 01/14/2020, see <a href="#">End of Support (EOS) for Microsoft Products</a>.</td>
</tr>
</tbody>
</table>

**Monthly AMI updates for 2019**

For more information about Microsoft updates, see [Description of Software Update Services and Windows Server Update Services changes in content for 2019](#).
<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2019.12.16</td>
<td><strong>All AMIs</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to December 10, 2019</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.903</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Microsoft Windows Server 2008 SP2 and Windows Server 2008 R2</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Microsoft will end mainstream support for Windows Server 2008 SP2 and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Windows Server 2008 R2 on January 14, 2020. On this date, AWS will no</td>
<td></td>
</tr>
<tr>
<td></td>
<td>longer publish or distribute Windows Server 2008 SP2 or Windows Server</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2008 R2 AMIs. Existing 2008 SP2/R2 instances and custom AMIs in your</td>
<td></td>
</tr>
<tr>
<td></td>
<td>account will not be impacted and you can continue to use them after</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the end-of-service (EOS) date.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For more information about Microsoft EOS on AWS, including upgrade and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>import options, along with a full list of AMIs that will no longer be</td>
<td></td>
</tr>
<tr>
<td></td>
<td>published or distributed on January 14, 2020, see</td>
<td></td>
</tr>
<tr>
<td></td>
<td>End of Support (EOS) for Microsoft Products.</td>
<td></td>
</tr>
<tr>
<td>2019.11.13</td>
<td><strong>All AMIs</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.876</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to November 12th, 2019</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• EC2 Config version 4.9.3865</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• EC2 Launch version 1.3.2002240</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• SSM Agent v2.3.722.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Previous versions of AMIs have been marked private.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>New Windows AMIs</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-1909-English-Core-Base-2019.11.13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-1909-English-Core-ContainersLatest-2019.11.13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2019_Enterprise-2019.11.13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2019_Express-2019.11.13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2019_Standard-2019.11.13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2019_Web-2019.11.13</td>
<td></td>
</tr>
<tr>
<td>2019.11.05</td>
<td><strong>New Windows AMIs</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>New SQL AMIs available:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2019_Enterprise-2019.11.05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2019_Express-2019.11.05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2019_Standard-2019.11.05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2019_Web-2019.11.05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2019-English-Full-SQL_2019_Enterprise-2019.11.05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2019-English-Full-SQL_2019_Express-2019.11.05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2019-English-Full-SQL_2019_Web-2019.11.05</td>
<td></td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>2019.10.09</td>
<td><strong>All AMIs</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.15.846</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows security updates current to October 8th, 2019</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows Defender platform updates current and update block via registry removed. For details, see <a href="https://support.microsoft.com/en-us/help/4513240/sfc-incorrectly-flags-windows-defender-ps-files-as-corrupted">https://support.microsoft.com/en-us/help/4513240/sfc-incorrectly-flags-windows-defender-ps-files-as-corrupted</a></td>
<td></td>
</tr>
<tr>
<td>2019.09.12</td>
<td><strong>New Windows AMIs</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>New ECS-optimized AMI available:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2019-English-Core-ECS_Optimized-2019.10.09</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>New Windows AMI</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• amzn2-ami-hvm-2.0.20190618-x86_64-gp2-mono</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.NET Core 2.2, Mono 5.18, and PowerShell 6.2 pre-installed to run your .NET applications on Amazon Linux 2 with Long Term Support (LTS)</td>
<td></td>
</tr>
</tbody>
</table>
## AWS Windows AMI Version History

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019.09.11</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>- AWS PV driver version 8.3.2</td>
</tr>
<tr>
<td></td>
<td>- AWS NVMe driver version 1.3.2</td>
</tr>
<tr>
<td></td>
<td>- AWS Tools for Windows PowerShell version 3.15.826</td>
</tr>
<tr>
<td></td>
<td>- NLA enabled on all OS 2012 RTM to 2019 AMIs</td>
</tr>
<tr>
<td></td>
<td>- Intel 82599 VF driver reverted to version 2.0.210.0 (Server 2016) or version 2.1.138.0 (Server 2019) due to customer reported issues. Engagement with Intel concerning these issues ongoing.</td>
</tr>
<tr>
<td></td>
<td>- Windows security updates current to September 10th, 2019</td>
</tr>
</tbody>
</table>

Previous versions of AMIs have been marked private.

### New Windows AMIs

New STIG-compliant AMIs available:

- Windows_Server-2012-R2-English-STIG-Full
- Windows_Server-2012-R2-English-STIG-Core
- Windows_Server-2016-English-STIG-Full
- Windows_Server-2016-English-STIG-Core
- Windows_Server-2019-English-STIG-Full
- Windows_Server-2019-English-STIG-Core

### Windows Server 2008 R2 SP1

Includes the following updates, which are required for Microsoft Extended Security (ESU) updates.

- KB4490628
- KB4474419
- KB4516655

### Windows Server 2008 SP2

Includes the following updates, which are required for Microsoft Extended Security (ESU) updates.

- KB4493730
- KB4474419
- KB4517134
<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Note</strong> NLA is now enabled on all 2012 RTM, 2012 R2, and 2016 AMIs to increase default RDP security posture. NLA remains enabled on 2019 AMIs.</td>
</tr>
<tr>
<td>2019.08.16</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>- EC2Config version 4.9.3519</td>
</tr>
<tr>
<td></td>
<td>- SSM Agent version 2.3.634.0</td>
</tr>
<tr>
<td></td>
<td>- AWS Tools for Windows PowerShell version 3.15.802</td>
</tr>
<tr>
<td></td>
<td>- Windows Defender platform update blocked via registry due to SFC failures introduced by update. Update will be re-enabled when new patch is available.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> Starting in September, NLA will be enabled on all 2012 RTM, 2012 R2, and 2016 AMIs to increase default RDP security posture.</td>
</tr>
<tr>
<td>2019.07.19</td>
<td><strong>New Windows AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>- Windows_Server-2016-English-Full-ECS_Optimized-2019.07.19</td>
</tr>
<tr>
<td>2019.07.12</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>- Microsoft security updates current to July 9th, 2019</td>
</tr>
<tr>
<td>2019.06.12</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>- Microsoft security updates current to June 11th, 2019</td>
</tr>
<tr>
<td></td>
<td>- AWS SDK version 3.15.756</td>
</tr>
<tr>
<td></td>
<td>- AWS PV driver version 8.2.7</td>
</tr>
<tr>
<td></td>
<td>- AWS NVMe driver version 1.3.1</td>
</tr>
<tr>
<td></td>
<td>- The following &quot;P3&quot; AMIs will be renamed as &quot;Tesla&quot; AMIs. These AMIs will support all GPU-backed AWS instances using the Tesla driver. P3 AMIs will no longer be updated after this release and will be removed as part of our regular cycle.</td>
</tr>
<tr>
<td></td>
<td>- Windows_Server-2012-R2_RTM-English-P3-2019.06.12 replaced with Windows_Server-2012-R2_RTM-English-Tesla-2019.06.12</td>
</tr>
<tr>
<td></td>
<td>- Windows_Server-2016-English-P3-2016.06.12 replaced with Windows_Server-2016-English-Tesla-2019.06.12</td>
</tr>
<tr>
<td></td>
<td><strong>New Windows AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>Previous versions of AMIs have been marked private.</td>
</tr>
<tr>
<td>2019.05.21</td>
<td><strong>Windows Server, version 1903</strong></td>
</tr>
<tr>
<td></td>
<td>- AMIs are now available</td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2019.05.15</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to May 14th, 2019</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.3429</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent version 2.3.542.0</td>
</tr>
<tr>
<td></td>
<td>• AWS SDK version 3.15.735</td>
</tr>
<tr>
<td>2019.04.26</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Fixed AMIs for Windows Server 2019 with SQL to address edge cases where</td>
</tr>
<tr>
<td></td>
<td>the first launch of an instance may result in Instance Impairment and</td>
</tr>
<tr>
<td></td>
<td>Windows displays the message &quot;Please wait for the User Profile Service&quot;</td>
</tr>
<tr>
<td>2019.04.21</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• AWS PV Driver rollback to version 8.2.6 from version 8.3.0</td>
</tr>
<tr>
<td>2019.04.10</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to April 9, 2019</td>
</tr>
<tr>
<td></td>
<td>• AWS SDK version 3.15.715</td>
</tr>
<tr>
<td></td>
<td>• AWS PV Driver version 8.3.0</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch version 1.3.2001360</td>
</tr>
<tr>
<td></td>
<td><strong>New Windows AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2012_SP4_Standard-2019.04.10</td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2014_SP3_Standard-2019.04.10</td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2014_SP3_Enterprise-2019.04.10</td>
</tr>
<tr>
<td>2019.03.13</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to March 12, 2019</td>
</tr>
<tr>
<td></td>
<td>• AWS SDK version 3.15.693</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch version 1.3.2001220</td>
</tr>
<tr>
<td></td>
<td>• NVIDIA Tesla driver version 412.29 for Deep Learning and P3 AMIs (<a href="https://nvidia.custhelp.com/app/answers/detail/a_id/4772">https://nvidia.custhelp.com/app/answers/detail/a_id/4772</a>)</td>
</tr>
<tr>
<td></td>
<td>Previous versions of AMIs have been marked private</td>
</tr>
</tbody>
</table>
### AWS Windows AMI Version History

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2019.02.13</strong></td>
<td>All AMIs</td>
</tr>
<tr>
<td></td>
<td>- Microsoft security updates current to February 12, 2019</td>
</tr>
<tr>
<td></td>
<td>- SSM Agent version 2.3.444.0</td>
</tr>
<tr>
<td></td>
<td>- AWS SDK version 3.15.666</td>
</tr>
<tr>
<td></td>
<td>- EC2Launch version 1.3.2001040</td>
</tr>
<tr>
<td></td>
<td>- EC2Config version 4.9.3289</td>
</tr>
<tr>
<td></td>
<td>- AWS PV driver 8.2.6</td>
</tr>
<tr>
<td></td>
<td>- EBS NVMe tool</td>
</tr>
</tbody>
</table>

SQL 2014 with Service Pack 2 and SQL 2016 with Service Pack 1 will no longer be updated after this release.

| **2019.02.09** | All AMIs |
|               | - Windows AMIs have been updated. New AMIs can be found with the following date versions: |
|               |   - November "2018.11.29" |
|               |   - December "2018.12.13" |
|               |   - January "2019.02.09" |
|               | - Previous versions of AMIs have been marked private |

| **2019.01.10** | All AMIs |
|               | - Microsoft security updates current to January 10, 2019 |
|               | - SSM Agent version 2.3.344.0 |
|               | - AWS SDK version 3.15.647 |
|               | - EC2Launch version 1.3.2000930 |
|               | - EC2Config version 4.9.3160 |

<table>
<thead>
<tr>
<th>All AMIs with SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Latest cumulative updates</td>
</tr>
</tbody>
</table>

### Monthly AMI updates for 2018

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

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2018.12.12</strong></td>
<td>All AMIs</td>
</tr>
<tr>
<td></td>
<td>- Microsoft security updates current to December 12, 2018</td>
</tr>
<tr>
<td></td>
<td>- SSM Agent version 2.3.274.0</td>
</tr>
<tr>
<td></td>
<td>- AWS SDK version 3.15.629</td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>• EC2Launch version 1.3.2000760</td>
<td></td>
</tr>
</tbody>
</table>

**New Windows AMIs**

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
</table>

**Updated Linux AMI**


2018.11.28

**All AMIs**

- SSM Agent version 2.3.235.0
- Changes in all power schemes to set the display to never turn off
<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018.11.20</td>
<td><strong>Windows Server-2016-English-Deep-Learning</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server-2016-English-Deep-Learning</strong></td>
</tr>
<tr>
<td></td>
<td>• TensorFlow version 1.12</td>
</tr>
<tr>
<td></td>
<td>• MXNet version 1.3</td>
</tr>
<tr>
<td></td>
<td>• NVIDIA version 392.05</td>
</tr>
<tr>
<td>2018.11.19</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to November 19, 2018</td>
</tr>
<tr>
<td></td>
<td>• AWS SDK version 3.15.602.0</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent version 2.3.193.0</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.3067</td>
</tr>
<tr>
<td></td>
<td>• Intel Chipset INF configurations to support new instance types</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server, version 1809</strong></td>
</tr>
<tr>
<td></td>
<td>• AMIs are now available.</td>
</tr>
<tr>
<td>2018.10.14</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to October 9, 2018</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.3.365.0</td>
</tr>
<tr>
<td></td>
<td>• CloudFormation version 1.4.31</td>
</tr>
<tr>
<td></td>
<td>• AWS PV Driver version 8.2.4</td>
</tr>
<tr>
<td></td>
<td>• AWS PCI Serial Driver version 1.0.0.0 (support for Windows 2008R2 and</td>
</tr>
<tr>
<td></td>
<td>2012 on Bare Metal instances</td>
</tr>
<tr>
<td></td>
<td>• ENA Driver version 1.5.0</td>
</tr>
<tr>
<td></td>
<td>**Microsoft Windows Server 2016 Datacenter and Standard Editions for Nano</td>
</tr>
<tr>
<td></td>
<td>Server**</td>
</tr>
<tr>
<td></td>
<td>Microsoft ended mainstream support for Windows Server 2016 Datacenter</td>
</tr>
<tr>
<td></td>
<td>and Standard Editions for Nano Server installation options as of April 10, 2018.</td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2018.09.15</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to September 12, 2018</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.3.343</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch version 1.3.2000430</td>
</tr>
<tr>
<td></td>
<td>• AWS NVMe Driver version 1.3.0</td>
</tr>
<tr>
<td></td>
<td>• EC2 WinUtil Driver version 2.0.0</td>
</tr>
<tr>
<td></td>
<td><strong>Microsoft Windows Server 2016 Base Nano</strong></td>
</tr>
<tr>
<td></td>
<td>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: <a href="https://docs.microsoft.com/en-us/windows-server/get-started/nano-in-semi-annual-channel">https://docs.microsoft.com/en-us/windows-server/get-started/nano-in-semi-annual-channel</a>.</td>
</tr>
<tr>
<td>2018.08.15</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to August 14, 2018</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell version 3.3.335</td>
</tr>
<tr>
<td></td>
<td>• AMIs now default to use Amazon’s NTP service at IP 169.254.169.123 for time synchronization. For more information, see Default network time protocol (NTP) settings for Amazon Windows AMIs (p. 561).</td>
</tr>
<tr>
<td></td>
<td><strong>Microsoft Windows Server 2016 Base Nano</strong></td>
</tr>
<tr>
<td></td>
<td>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: <a href="https://docs.microsoft.com/en-us/windows-server/get-started/nano-in-semi-annual-channel">https://docs.microsoft.com/en-us/windows-server/get-started/nano-in-semi-annual-channel</a>.</td>
</tr>
<tr>
<td>2018.07.11</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to July 10, 2018</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.2756</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.2.800.0</td>
</tr>
<tr>
<td>2018.06.22</td>
<td><strong>Windows Server 2008 R2</strong></td>
</tr>
<tr>
<td></td>
<td>• 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).</td>
</tr>
<tr>
<td>2018.06.13</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to June 12, 2018</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.2688</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.2.619.0</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell 3.3.283.0</td>
</tr>
<tr>
<td></td>
<td>• AWS NVMe driver 1.2.0</td>
</tr>
<tr>
<td></td>
<td>• AWS PV driver 8.2.3</td>
</tr>
</tbody>
</table>
## AWS Windows AMI Version History

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2018.05.09</strong></td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to May 9, 2018</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.2644</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.2.493.0</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell 3.3.270.0</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server, version 1709 and Windows Server, version 1803</strong></td>
</tr>
<tr>
<td></td>
<td>• AMIs are now available. For more information, see Windows Server version 1709 and 1803 AMIs for Amazon EC2.</td>
</tr>
<tr>
<td><strong>2018.04.11</strong></td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to April 10, 2018</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.2586</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.2.392.0</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell 3.3.256.0</td>
</tr>
<tr>
<td></td>
<td>• AWS CloudFormation templates 1.4.30</td>
</tr>
<tr>
<td></td>
<td>• Serial INF and Intel Chipset INF configurations to support new instance types</td>
</tr>
<tr>
<td></td>
<td><strong>SQL Server 2017</strong></td>
</tr>
<tr>
<td></td>
<td>• Cumulative update 5 (CU5)</td>
</tr>
<tr>
<td></td>
<td><strong>SQL Server 2016 SP1</strong></td>
</tr>
<tr>
<td></td>
<td>• Cumulative update 8 (CU8)</td>
</tr>
<tr>
<td><strong>2018.03.24</strong></td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to March 13, 2018</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.2565</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.2.355.0</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell 3.3.245.0</td>
</tr>
<tr>
<td></td>
<td>• AWS PV driver 8.2</td>
</tr>
<tr>
<td></td>
<td>• AWS ENA driver 1.2.3.0</td>
</tr>
<tr>
<td></td>
<td>• Amazon EC2 Hibernate Agent 1.0 (rollback from 2.1.0 in the 2018.03.16 AMI release)</td>
</tr>
<tr>
<td></td>
<td>• AWS EC2WinUtilDriver 1.0.1 (for troubleshooting)</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2016</strong></td>
</tr>
<tr>
<td></td>
<td>• EC2Launch 1.3.2000080</td>
</tr>
<tr>
<td><strong>2018.03.16</strong></td>
<td>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. 35).</td>
</tr>
</tbody>
</table>
**Amazon Elastic Compute Cloud**  
**User Guide for Windows Instances**  
**AWS Windows AMI Version History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018.03.06</td>
<td>All AMIs</td>
</tr>
<tr>
<td></td>
<td>• AWS PV driver 8.2.1</td>
</tr>
<tr>
<td>2018.02.23</td>
<td>All AMIs</td>
</tr>
<tr>
<td></td>
<td>• AWS PV driver 7.4.6 (rollback from 8.2 in the 2018.02.13 AMI release)</td>
</tr>
<tr>
<td>2018.02.13</td>
<td>All AMIs</td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to February 13, 2018</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.2400</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.2.160.0</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell 3.3.225.1</td>
</tr>
<tr>
<td></td>
<td>• AWS PV driver 8.2</td>
</tr>
<tr>
<td></td>
<td>• AWS ENA driver 1.2.3.0</td>
</tr>
<tr>
<td></td>
<td>• AWS NVMe driver 1.0.0.146</td>
</tr>
<tr>
<td></td>
<td>• Amazon EC2 HibernateAgent 1.0.0</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2016</strong></td>
</tr>
<tr>
<td></td>
<td>• EC2Launch 1.3.740</td>
</tr>
<tr>
<td>2018.01.12</td>
<td>All AMIs</td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to January 9, 2018</td>
</tr>
<tr>
<td>2018.01.05</td>
<td>All AMIs</td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to January 2018</td>
</tr>
<tr>
<td></td>
<td>• Registry settings to enable mitigations for the Spectre and Meltdown exploits</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell 3.3.215</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.2262</td>
</tr>
</tbody>
</table>

**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](#).

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017.12.13</td>
<td>All AMIs</td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to December 12, 2017</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.2218</td>
</tr>
<tr>
<td></td>
<td>• AWS CloudFormation templates 1.4.27</td>
</tr>
<tr>
<td></td>
<td>• AWS NVMe driver 1.02</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.2.93.0</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell 3.3.201</td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2017.11.29</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Removed components for Volume Shadow Copy Service (VSS) included in</td>
</tr>
<tr>
<td></td>
<td>2017.11.18 and 2017.11.19 due to a compatibility issue with Windows</td>
</tr>
<tr>
<td></td>
<td>Backup.</td>
</tr>
<tr>
<td>2017.11.19</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• EC2 Hibernate Agent 1.0 (supports hibernation for Spot Instances)</td>
</tr>
<tr>
<td>2017.11.18</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to November 14, 2017</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.2218</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.2.64.0</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell 3.3.182</td>
</tr>
<tr>
<td></td>
<td>• Elastic Network Adapter (ENA) driver 1.08 (rollback from 1.2.2 in the</td>
</tr>
<tr>
<td></td>
<td>2017.10.13 AMI release)</td>
</tr>
<tr>
<td></td>
<td>• Query for the latest Windows AMI using Systems Manager Parameter Store</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2016</strong></td>
</tr>
<tr>
<td></td>
<td>• EC2Launch 1.3.640</td>
</tr>
<tr>
<td>2017.10.13</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to October 11, 2017</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.2188</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.2.30.0</td>
</tr>
<tr>
<td></td>
<td>• AWS CloudFormation templates 1.4.24</td>
</tr>
<tr>
<td></td>
<td>• Elastic Network Adapter (ENA) driver 1.2.2. (Windows Server 2008 R2</td>
</tr>
<tr>
<td></td>
<td>through Windows Server 2016)</td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td><strong>2017.10.04</strong></td>
<td><strong>Microsoft SQL Server</strong></td>
</tr>
<tr>
<td></td>
<td>Windows Server 2016 with Microsoft SQL Server 2017 AMIs are now public in all regions.</td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2017_Enterprise-2017.10.04</td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2017_Standard-2017.10.04</td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2017_WEB-2017.10.04</td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2017_Express-2017.10.04</td>
</tr>
<tr>
<td></td>
<td>Microsoft SQL Server 2017 supports the following features:</td>
</tr>
<tr>
<td></td>
<td>• Machine Learning Services with Python (ML and AI) and R language support</td>
</tr>
<tr>
<td></td>
<td>• Automatic database tuning</td>
</tr>
<tr>
<td></td>
<td>• Clusterless Availability Groups</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• Windows-Linux cross-OS migrations</td>
</tr>
<tr>
<td></td>
<td>• Resumable online index rebuild</td>
</tr>
<tr>
<td></td>
<td>• Improved adaptive query processing</td>
</tr>
<tr>
<td></td>
<td>• Graph data support</td>
</tr>
<tr>
<td><strong>2017.09.13</strong></td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to September 13, 2017</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.2106</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.0.952.0</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell 3.3.143</td>
</tr>
<tr>
<td></td>
<td>• AWS CloudFormation templates 1.4.21</td>
</tr>
<tr>
<td><strong>2017.08.09</strong></td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to August 9, 2017</td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.2016</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.0.879.0</td>
</tr>
</tbody>
</table>

**Windows Server 2012 R2**

• Due to an internal error, these AMIs were released with an older version of AWS Tools for Windows PowerShell, 3.3.58.0.
### AWS Windows AMI Version History

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2017.07.13</strong></td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>- Microsoft security updates current to July 13, 2017</td>
</tr>
<tr>
<td></td>
<td>- SSM Agent 2.0.847.0</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2016</strong></td>
</tr>
<tr>
<td></td>
<td>- Intel SRIOV Driver 2.0.210.0</td>
</tr>
<tr>
<td><strong>2017.06.14</strong></td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>- Microsoft security updates current to June 14, 2017</td>
</tr>
<tr>
<td></td>
<td>- Updates for .NET Framework 4.7 installed from Windows Update</td>
</tr>
<tr>
<td></td>
<td>- Microsoft updates to address the &quot;privilege not held&quot; error using the PowerShell Stop-Computer cmdlet. For more information, see <a href="#">Privilege not held error</a> on the Microsoft site.</td>
</tr>
<tr>
<td></td>
<td>- EC2Config version 4.9.1900</td>
</tr>
<tr>
<td></td>
<td>- SSM Agent 2.0.805.0</td>
</tr>
<tr>
<td></td>
<td>- AWS Tools for Windows PowerShell 3.3.99.0</td>
</tr>
<tr>
<td></td>
<td>- Internet Explorer 11 for the desktop is the default, instead of the immersive Internet Explorer</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2016</strong></td>
</tr>
<tr>
<td></td>
<td>- EC2Launch 1.3.610</td>
</tr>
<tr>
<td><strong>2017.05.30</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>2017.05.22</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>2017.05.10</strong></td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>- Microsoft security updates current to May 9, 2017</td>
</tr>
<tr>
<td></td>
<td>- AWS PV Driver v7.4.6</td>
</tr>
<tr>
<td></td>
<td>- AWS Tools for Windows PowerShell 3.3.83.0</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2016</strong></td>
</tr>
<tr>
<td></td>
<td>- SSM Agent 2.0.767</td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2017.04.12</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to April 11, 2017</td>
</tr>
<tr>
<td></td>
<td>• AWS Tools for Windows PowerShell 3.3.71.0</td>
</tr>
<tr>
<td></td>
<td>• AWS CloudFormation templates 1.4.18</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2003 to Windows Server 2012</strong></td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.9.1775</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.0.761.0</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2016</strong></td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.0.730.0</td>
</tr>
<tr>
<td>2017.03.15</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to March 14, 2017</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• Current AWS CloudFormation templates</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2003 to Windows Server 2012</strong></td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.7.1631</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.0.682.0</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2016</strong></td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.0.706.0</td>
</tr>
<tr>
<td></td>
<td>• EC2Launch v1.3.540</td>
</tr>
<tr>
<td>2017.02.21</td>
<td>Microsoft recently <strong>announced</strong> 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.</td>
</tr>
<tr>
<td></td>
<td>Amazon Web Services did not release updated Windows Server AMIs in February.</td>
</tr>
<tr>
<td>2017.01.11</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to January 10, 2017</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• Current AWS CloudFormation templates</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2003 to Windows Server 2012</strong></td>
</tr>
<tr>
<td></td>
<td>• EC2Config version 4.2.1442</td>
</tr>
<tr>
<td></td>
<td>• SSM Agent 2.0.599.0</td>
</tr>
</tbody>
</table>
## 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.

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016.12.14</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to December 13, 2016</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td><strong>Windows Server 2003 to Windows Server 2012</strong></td>
<td>• Released EC2Config version 4.1.1396</td>
</tr>
<tr>
<td></td>
<td>• Elastic Network Adapter (ENA) driver 1.0.9.0 (Windows Server 2008 R2 only)</td>
</tr>
<tr>
<td><strong>Windows Server 2016</strong></td>
<td>New AMIs available in all regions:</td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Core-Base</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server</strong></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2008-R2_SP1-English-64Bit-SQL_2012_SP3_edition-2016.12.14</td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2012-R2_RTM-English-64Bit-SQL_2016_SP1_edition-2016.12.14</td>
</tr>
<tr>
<td></td>
<td>• Windows_Server-2016-English-Full-SQL_2016_SP1_edition-2016.12.14</td>
</tr>
</tbody>
</table>

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:

- Row-level security
- Dynamic Data Masking
- Change Data Capture
- Database snapshot
- Column store
- Partitioning
- Compression
<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016.11.23</td>
<td><strong>Windows Server 2003 to Windows Server 2012</strong></td>
</tr>
<tr>
<td></td>
<td>• Released EC2Config version 4.1.1378</td>
</tr>
<tr>
<td></td>
<td>• The AMIs released this month, and going forward, use the EC2Config service to process boot-time</td>
</tr>
<tr>
<td></td>
<td>configurations and SSM Agent to process AWS Systems Manager Run Command and Config requests.</td>
</tr>
<tr>
<td></td>
<td>EC2Config no longer processes requests for Systems Manager Run Command and State Manager. The</td>
</tr>
<tr>
<td></td>
<td>latest EC2Config installer installs SSM Agent side-by-side with the EC2Config service.</td>
</tr>
<tr>
<td></td>
<td>For more information, see Configure a Windows instance using the EC2Config service (p. 493).</td>
</tr>
<tr>
<td>2016.11.09</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to November 8 2016</td>
</tr>
<tr>
<td></td>
<td>• Released AWS PV driver, version 7.4.3.0 for Windows 2008 R2 and later</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td>2016.10.18</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to October 12, 2016</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2016</strong></td>
</tr>
<tr>
<td></td>
<td>• Released AMIs for Windows Server 2016. These AMIs include significant changes. For example,</td>
</tr>
<tr>
<td></td>
<td>they don't include the EC2Config service. For more information, see Changes in</td>
</tr>
<tr>
<td></td>
<td>Windows Server 2016 and later AMIs (p. 35).</td>
</tr>
<tr>
<td>2016.9.14</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to September 13, 2016</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• Renamed AMI Windows_Server-2012-RTM-Japanese-64Bit-SQL_2008_R3_SP2_Standard to Windows_Server-</td>
</tr>
<tr>
<td></td>
<td>2012-RTM-Japanese-64Bit-SQL_2008_R2_SP3_Standard</td>
</tr>
<tr>
<td>2016.8.26</td>
<td>**All Windows Server 2008 R2 AMIs dated 2016.08.11 were updated to fix a known issue. New AMIs</td>
</tr>
<tr>
<td></td>
<td>are dated 2016.08.25.</td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| 2016.8.11 | **All AMIs**  
  - 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 |
  - 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 | **All AMIs**  
  - 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 | **All AMIs**  
  - 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 | **All AMIs**  
  - Microsoft security updates current to June 2016  
  - Current AWS Tools for Windows PowerShell  
  - EC2Config service version 3.17.1032 |
|          | **Microsoft SQL Server**  
  - 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. |
<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016.5.11</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to May 2016</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.16.930</td>
</tr>
<tr>
<td></td>
<td>• MS15-011 Active Directory patch installed</td>
</tr>
<tr>
<td></td>
<td><strong>Windows Server 2012 R2</strong></td>
</tr>
<tr>
<td></td>
<td>• Intel SRIOV Driver 1.0.16.1</td>
</tr>
<tr>
<td>2016.4.13</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to April 2016</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.15.880</td>
</tr>
<tr>
<td>2016.3.9</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to March 2016</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.14.786</td>
</tr>
<tr>
<td>2016.2.10</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to February 2016</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.13.727</td>
</tr>
<tr>
<td>2016.1.25</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to January 2016</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.12.649</td>
</tr>
<tr>
<td>2016.1.5</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015.12.15</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to December 2015</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2015.11.11</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to November 2015</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.11.521</td>
</tr>
<tr>
<td></td>
<td>• CFN Agent updated to latest version</td>
</tr>
<tr>
<td>2015.10.26</td>
<td>Corrected boot volume sizes of base AMIs to be 30GB instead of 35GB</td>
</tr>
<tr>
<td>2015.10.14</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to October 2015</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.10.442</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• Updated SQL Service Packs to latest versions for all SQL variants</td>
</tr>
<tr>
<td></td>
<td>• Removed old entries in Event Logs</td>
</tr>
<tr>
<td></td>
<td>• AMI Names have been changed to reflect the latest service pack.</td>
</tr>
<tr>
<td></td>
<td>For example, the latest AMI with Server 2012 and SQL 2014 Standard is</td>
</tr>
<tr>
<td>2015.9.9</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to September 2015</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.9.359</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• Current AWS CloudFormation helper scripts</td>
</tr>
<tr>
<td>2015.8.18</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to August 2015</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.8.294</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td><strong>Only AMIs with Windows Server 2012 and Windows Server 2012 R2</strong></td>
</tr>
<tr>
<td></td>
<td>• AWS PV Driver 7.3.2</td>
</tr>
<tr>
<td>2015.7.21</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to July 2015</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.7.308</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• Modified AMI descriptions of SQL images for consistency</td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
</tr>
<tr>
<td>2015.6.10</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to June 2015</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.6.269</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• Current AWS CloudFormation helper scripts</td>
</tr>
<tr>
<td></td>
<td><strong>Only AMIs with Windows Server 2012 R2</strong></td>
</tr>
<tr>
<td></td>
<td>• AWS PV Driver 7.3.1</td>
</tr>
<tr>
<td>2015.5.13</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to May 2015</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.5.228</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td>2015.04.15</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to April 2015</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.3.174</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td>2015.03.11</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to March 2015</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.2.97</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td><strong>Only AMIs with Windows Server 2012 R2</strong></td>
</tr>
<tr>
<td></td>
<td>• AWS PV Driver 7.3.0</td>
</tr>
<tr>
<td>2015.02.11</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to February 2015</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 3.0.54</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• Current AWS CloudFormation helper scripts</td>
</tr>
<tr>
<td>2015.01.14</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to January 2015</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.3.313</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• Current AWS CloudFormation helper scripts</td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014.12.10</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to December 2014</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.2.12</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td>2014.11.19</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to November 2014</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.2.11</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td>2014.10.15</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to October 2014</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.2.10</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td><strong>Only AMIs with Windows Server 2012 R2</strong></td>
</tr>
<tr>
<td></td>
<td>• AWS PV Driver 7.2.4.1 (resolves the issues with Plug and Play Cleanup, which is now enabled by default)</td>
</tr>
<tr>
<td>2014.09.10</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to September 2014</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.2.8</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td><strong>Only AMIs with Windows Server 2012 R2</strong></td>
</tr>
<tr>
<td></td>
<td>• Disable Plug and Play Cleanup (see Important information)</td>
</tr>
<tr>
<td></td>
<td>• AWS PV Driver 7.2.2.1 (resolves issues with the uninstaller)</td>
</tr>
<tr>
<td>2014.08.13</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to August 2014</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.2.7</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td><strong>Only AMIs with Windows Server 2012 R2</strong></td>
</tr>
<tr>
<td></td>
<td>• AWS PV Driver 7.2.2.1 (improves disk performance, resolves issues with reconnecting multiple network interfaces and lost network settings)</td>
</tr>
</tbody>
</table>
## AWS Windows AMI Version History

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014.07.10</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to July 2014</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.2.5</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td>2014.06.12</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to June 2014</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.2.4</td>
</tr>
<tr>
<td></td>
<td>• Removed NVIDIA drivers (except for Windows Server 2012 R2 AMIs)</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td>2014.05.14</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to May 2014</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.2.2</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• AWS CloudFormation helper scripts version 1.4.0</td>
</tr>
<tr>
<td>2014.04.09</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to April 2014</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• Current AWS CloudFormation helper scripts</td>
</tr>
<tr>
<td>2014.03.12</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to March 2014</td>
</tr>
<tr>
<td>2014.02.12</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to February 2014</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.2.1</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• KB2634328</td>
</tr>
<tr>
<td></td>
<td>• Remove the BCDEdit useplatformclock value</td>
</tr>
<tr>
<td></td>
<td><strong>Only AMIs with Microsoft SQL Server</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft SQL Server 2012 SP1 cumulative update package 8</td>
</tr>
<tr>
<td></td>
<td>• Microsoft SQL Server 2008 R2 cumulative update package 10</td>
</tr>
</tbody>
</table>

## Monthly AMI updates for 2013

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013.11.13</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to November 2013</td>
</tr>
<tr>
<td>Release</td>
<td>Changes</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.1.19</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• Configure NTP to synchronize the time once a day (the default is every seven days)</td>
</tr>
<tr>
<td></td>
<td><strong>Only AMIs with Windows Server 2012</strong></td>
</tr>
<tr>
<td></td>
<td>• Clean up the WinSXS folder using the following command: dism /online /cleanup-image /StartComponentCleanup</td>
</tr>
<tr>
<td>2013.09.11</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to September 2013</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.1.18</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• AWS CloudFormation helper scripts version 1.3.15</td>
</tr>
<tr>
<td>2013.07.10</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to July 2013</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.1.16</td>
</tr>
<tr>
<td></td>
<td>• Expanded the root volume to 50 GB</td>
</tr>
<tr>
<td></td>
<td>• Set the page file to 512 MB, expanding to 8 GB as needed</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td>2013.06.12</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to June 2013</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td><strong>Only AMIs with Microsoft SQL Server</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft SQL Server 2012 SP1 with cumulative update package 4</td>
</tr>
<tr>
<td>2013.05.15</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to May 2013</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.1.15</td>
</tr>
<tr>
<td></td>
<td>• All instance store volumes attached by default</td>
</tr>
<tr>
<td></td>
<td>• Remote PowerShell enabled by default</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td>2013.04.14</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to April 2013</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• AWS CloudFormation helper scripts version 1.3.14</td>
</tr>
</tbody>
</table>
## AWS Windows AMI Version History

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013.03.14</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to March 2013</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.1.14</td>
</tr>
<tr>
<td></td>
<td>• Citrix Agent with CPU heartbeat fix</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• AWS CloudFormation helper scripts version 1.3.11</td>
</tr>
<tr>
<td>2013.02.22</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to February 2013</td>
</tr>
<tr>
<td></td>
<td>• KB2800213</td>
</tr>
<tr>
<td></td>
<td>• Windows PowerShell 3.0 upgrade</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.1.13</td>
</tr>
<tr>
<td></td>
<td>• Citrix Agent with time fix</td>
</tr>
<tr>
<td></td>
<td>• Citrix PV drivers dated 2011.07.19</td>
</tr>
<tr>
<td></td>
<td>• Current AWS Tools for Windows PowerShell</td>
</tr>
<tr>
<td></td>
<td>• AWS CloudFormation helper scripts version 1.3.8</td>
</tr>
<tr>
<td></td>
<td><strong>Only AMIs with Microsoft SQL Server</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft SQL Server 2012 cumulative update package 5</td>
</tr>
</tbody>
</table>

### Monthly AMI updates for 2012

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012.12.12</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to December 2012</td>
</tr>
<tr>
<td></td>
<td>• Set the ActiveTimeBias registry value to 0</td>
</tr>
<tr>
<td></td>
<td>• Disable IPv6 for the network adapter</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.1.9</td>
</tr>
<tr>
<td></td>
<td>• Add AWS Tools for Windows PowerShell and set the policy to allow import-module</td>
</tr>
<tr>
<td>2012.11.15</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to November 2012</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.1.7</td>
</tr>
<tr>
<td>2012.10.10</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to October 2012</td>
</tr>
<tr>
<td>2012.08.15</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to August 2012</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.1.2</td>
</tr>
</tbody>
</table>
## AWS Windows AMI Version History

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012.07.11</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• KB2545227</td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to July 2012</td>
</tr>
<tr>
<td>2012.06.12</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to June 2012</td>
</tr>
<tr>
<td></td>
<td>• Set page file to 4 GB</td>
</tr>
<tr>
<td></td>
<td>• Remove installed language packs</td>
</tr>
<tr>
<td></td>
<td>• Set performance option to &quot;Adjust for best performance&quot;</td>
</tr>
<tr>
<td></td>
<td>• Set the screen saver to no longer display the logon screen on resume</td>
</tr>
<tr>
<td></td>
<td>• Remove previous RedHat driver versions using pnputil</td>
</tr>
<tr>
<td></td>
<td>• Remove duplicate bootloaders and set bootstatuspolicy to ignoreallfailures using bcdedit</td>
</tr>
<tr>
<td>2012.05.10</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to May 2012</td>
</tr>
<tr>
<td></td>
<td>• EC2Config service version 2.1.0</td>
</tr>
<tr>
<td>2012.04.11</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to April 2012</td>
</tr>
<tr>
<td></td>
<td>• KB2582281</td>
</tr>
<tr>
<td></td>
<td>• Current version of EC2Config</td>
</tr>
<tr>
<td></td>
<td>• System time in UTC instead of GMT</td>
</tr>
<tr>
<td>2012.03.13</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to March 2012</td>
</tr>
<tr>
<td>2012.02.24</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to February 2012</td>
</tr>
<tr>
<td></td>
<td>• Standardize AMI names and descriptions</td>
</tr>
<tr>
<td>2012.01.12</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to January 2012</td>
</tr>
<tr>
<td></td>
<td>• RedHat PV driver version 1.3.10</td>
</tr>
</tbody>
</table>

### Monthly AMI updates for 2011 and earlier

<table>
<thead>
<tr>
<th>Release</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011.09.11</td>
<td><strong>All AMIs</strong></td>
</tr>
<tr>
<td></td>
<td>• Microsoft security updates current to September 2011</td>
</tr>
</tbody>
</table>
Find 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 Find a Linux AMI in the Amazon EC2 User Guide for Linux Instances.

Contents
- Find a Windows AMI using the Amazon EC2 console (p. 97)
- Find an AMI using the AWS Tools for Windows PowerShell (p. 98)
- Find an AMI using the AWS CLI (p. 98)
- Find the latest Windows AMI using Systems Manager (p. 99)
- Use a Systems Manager parameter to find an AMI (p. 99)

Find a Windows AMI using the Amazon EC2 console

You can find Windows AMIs using the Amazon EC2 console. You can select from the list of AMIs when you use the launch wizard to launch an instance, or you can search through all available AMIs using the Images page. AMI IDs are unique to each AWS Region.
To find a Windows AMI using the launch wizard

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region in which to launch your instances. You can select any Region that's available to you, regardless of your location.
3. From the console dashboard, choose Launch instance.
4. On the Quick Start tab, select from one of the commonly used AMIs in the list. If you don’t see the AMI that you need, select the My AMIs, AWS Marketplace, or Community AMIs tab to find additional AMIs. For more information, see Step 1: Choose an Amazon Machine Image (AMI) (p. 392).

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. 393). If you're not ready to launch the instance now, make note of the AMI ID for later.

Find 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.

Find 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.

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

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

```bash
--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](#).

**Find the latest Windows AMI using Systems Manager**

Amazon EC2 provides AWS Systems Manager public parameters for AWS-maintained public AMIs that you can use when launching instances.

The Amazon EC2 AMI public parameters are available from the following path:

```
/aws/service/ami-windows-latest
```

You can view a list of all Windows AMIs in the current AWS Region by using the following command in the AWS CLI.

```bash
aws ssm get-parameters-by-path --path /aws/service/ami-windows-latest --query "Parameters[].Name"
```

For more information, see [Using public parameters](#) in the *AWS Systems Manager User Guide* and [Query for the Latest Windows AMI Using AWS Systems Manager Parameter Store](#).

**Use a Systems Manager parameter to find an AMI**

When you launch an instance using the EC2 launch wizard in the console, you can either select an AMI from the list, or you can select an AWS Systems Manager parameter that points to an AMI ID. If you use automation code to launch your instances, you can specify the Systems Manager parameter instead of the AMI ID.

A Systems Manager parameter is a customer-defined key-value pair that you can create in Systems Manager Parameter Store. The Parameter Store provides a central store to externalize your application configuration values. For more information, see *AWS Systems Manager Parameter Store* in the *AWS Systems Manager User Guide*.

When you create a parameter that points to an AMI ID, make sure that you specify the data type as `aws:ec2:image`. This data type ensures that when the parameter is created or modified, the parameter value is validated as an AMI ID. For more information, see *Native parameter support for Amazon Machine Image IDs* in the *AWS Systems Manager User Guide*.
Use cases

By using Systems Manager parameters to point to AMI IDs, you can make it easier for your users to select the correct AMI when launching instances, and you can simplify the maintenance of automation code.

Easier for users

If you require instances to be launched using a specific AMI, and if that AMI is updated regularly, we recommend that you require your users to select a Systems Manager parameter to find the AMI. By requiring your users to select a Systems Manager parameter, you can ensure that the latest AMI is used to launch instances.

For example, every month in your organization you might create a new version of your AMI that has the latest operating system and application patches. You also require your users to launch instances using the latest version of your AMI. To ensure that your users use the latest version, you can create a Systems Manager parameter (for example, golden-ami) that points to the correct AMI ID. Each time a new version of the AMI is created, you update the AMI ID value in the parameter so that it always points to the latest AMI. Your users don't need to know about the periodic updates to the AMI, because they continue to select the same Systems Manager parameter every time. By having users select a Systems Manager parameter, you make it easier for them to select the correct AMI for an instance launch.

Simplify automation code maintenance

If you use automation code to launch your instances, you can specify the Systems Manager parameter instead of the AMI ID. If a new version of the AMI is created, you change the AMI ID value in the parameter so that it points to the latest AMI. The automation code that references the parameter doesn't need to be modified every time a new version of the AMI is created. This greatly simplifies maintenance of automation and helps drive down deployment costs.

Note
Running instances are not affected when you change the AMI ID to which the Systems Manager parameter points.

Launch an instance using a Systems Manager parameter

You can launch an instance using the console or the AWS CLI. Instead of specifying an AMI ID, you can specify an AWS Systems Manager parameter that points to an AMI ID.

To find a Windows AMI using a Systems Manager parameter (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region in which to launch your instances. You can select any Region that's available to you, regardless of your location.
3. From the console dashboard, choose Launch instance.
4. Choose Search by Systems Manager parameter (at top right).
5. For Systems Manager parameter, select a parameter. The corresponding AMI ID appears next to Currently resolves to.
6. Choose Search. The AMIs that match the AMI ID appear in the list.
7. Select the AMI from the list, and choose Select.
For more information about launching an instance from an AMI using the launch wizard, see Step 1: Choose an Amazon Machine Image (AMI) (p. 392).

To launch an instance using an AWS Systems Manager parameter instead of an AMI ID (AWS CLI)

The following example uses the Systems Manager parameter `golden-ami` to launch an `m5.xlarge` instance. The parameter points to an AMI ID.

To specify the parameter in the command, use the following syntax: `resolve:ssm:parameter-name`, where `resolve:ssm` is the standard prefix and `parameter-name` is the unique parameter name. Note that the parameter name is case-sensitive. Backslashes for the parameter name are only necessary when the parameter is part of a hierarchy, for example, `/amis/production/golden-ami`. You can omit the backslash if the parameter is not part of a hierarchy.

In this example, the `--count` and `--security-group` parameters are not included. For `--count`, the default is 1. If you have a default VPC and a default security group, they are used.

```
aws ec2 run-instances
   --image-id resolve:ssm:/golden-ami
   --instance-type m5.xlarge
   ...
```

To launch an instance using a specific version of an AWS Systems Manager parameter (AWS CLI)

Systems Manager parameters have version support. Each iteration of a parameter is assigned a unique version number. You can reference the version of the parameter as follows `resolve:ssm:parameter-name:version`, where `version` is the unique version number. By default, the latest version of the parameter is used when no version is specified.

The following example uses version 2 of the parameter.

In this example, the `--count` and `--security-group` parameters are not included. For `--count`, the default is 1. If you have a default VPC and a default security group, they are used.

```
aws ec2 run-instances
   --image-id resolve:ssm:/golden-ami:2
   --instance-type m5.xlarge
   ...
```

To launch an instance using a public parameter provided by AWS

Amazon EC2 provides Systems Manager public parameters for public AMIs provided by AWS. For example, the public parameter `/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2` is available in all Regions and always points to the latest version of the Amazon Linux 2 AMI in the Region.

```
aws ec2 run-instances
   --image-id resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2
   --instance-type m5.xlarge
   ...
```

Permissions

If you use Systems Manager parameters that point to AMI IDs in the launch instance wizard, you must add `ssm:DescribeParameters` and `ssm:GetParameters` to your IAM policy. `ssm:DescribeParameters` grants your IAM users the permission to view and select Systems Manager parameters. `ssm:GetParameters` grants your IAM users the permission to get the values of the Systems Manager parameters. You can also restrict access to specific Systems Manager parameters. For more information, see Use the EC2 launch wizard (p. 1106).
Shared AMIs

A shared AMI is an AMI that a developer created and made available for others to use. One of the easiest ways to get started with Amazon EC2 is to use a shared AMI that has the components you need and then add custom content. You can also create your own AMIs and share them with others.

You use a shared AMI at your own risk. Amazon can't vouch for the integrity or security of AMIs shared by other Amazon EC2 users. Therefore, you should treat shared AMIs as you would any foreign code that you might consider deploying in your own data center and perform the appropriate due diligence. We recommend that you get an AMI from a trusted source.

Amazon's public images have an aliased owner, which appears as `amazon` in the account field. This enables you to find AMIs from Amazon easily. Other users can't alias their AMIs.

For information about creating an AMI, see Create a custom Windows AMI. For information about building, delivering, and maintaining your applications on the AWS Marketplace, see the AWS Marketplace Documentation.

Contents

- Find shared AMIs (p. 102)
- Make an AMI public (p. 104)
- Share an AMI with specific AWS accounts (p. 106)
- Use bookmarks (p. 108)
- Best Practices for shared Windows AMIs (p. 108)

Find shared AMIs

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

AMIs are a regional resource. Therefore, when searching for a shared AMI (public or private), you must search for it from within the Region from which it is being shared. To make an AMI available in a different Region, copy the AMI to the Region and then share it. For more information, see Copy an AMI (p. 113).

Topics

- Find a shared AMI (console) (p. 102)
- Find a shared AMI (Tools for Windows PowerShell) (p. 103)
- Find a shared AMI (AWS CLI) (p. 103)

Find 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.

Find 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" }
```

Find 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"
```

Make an AMI public

Amazon EC2 enables you to share your AMIs with other AWS accounts. You can allow all AWS accounts to use the AMI to launch instances (by making the AMI public), or only allow a few specific accounts to use the AMI to launch instances (see Share an AMI with specific AWS accounts (p. 106)). You are not billed when your AMI is used by other AWS accounts to launch instances; only the accounts launching instances using the AMI are billed for the instances they launch.

AMIs with encrypted volumes cannot be made public.

AMIs are a regional resource. Therefore, sharing an AMI makes it available in that Region. To make an AMI available in a different Region, copy the AMI to the Region and then share it. For more information, see Copy an AMI (p. 113).

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

Topics

- Share an AMI with all AWS accounts (console) (p. 105)
- Share an AMI with all AWS accounts (Tools for Windows PowerShell) (p. 105)
- Share an AMI with all AWS accounts (AWS CLI) (p. 105)
Share 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.

Share 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.

   ```powershell
   PS C:\> Edit-EC2ImageAttribute -ImageId ami-0abcdef1234567890 -Attribute launchPermission -OperationType add -UserGroup all
   ```

2. To verify the launch permissions of the AMI, use the following Get-EC2ImageAttribute command.

   ```powershell
   PS C:\> Get-EC2ImageAttribute -ImageId ami-0abcdef1234567890 -Attribute launchPermission
   ```

3. (Optional) To make the AMI private again, remove the all group from its launch permissions.
   Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

   ```powershell
   PS C:\> Edit-EC2ImageAttribute -ImageId ami-0abcdef1234567890 -Attribute launchPermission -OperationType remove -UserGroup all
   ```

Share 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.

   ```bash
   aws ec2 modify-image-attribute \\
   --image-id ami-0abcdef1234567890 \\
   --launch-permission "Add=[{Group=all}]"
   ```

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

   ```bash
   aws ec2 describe-image-attribute \\
   --image-id ami-0abcdef1234567890 \\
   --attribute launchPermission
   ```

3. (Optional) To make the AMI private again, remove the all group from its launch permissions.

   Note that the owner of the AMI always has launch permissions and is therefore unaffected by this command.

   ```bash
   aws ec2 modify-image-attribute \\
   --image-id ami-0abcdef1234567890 \\
   --launch-permission "Remove=[{Group=all}]"
   ```

Share an AMI with specific AWS accounts

You can share an AMI with specific AWS accounts without making the AMI public. All you need is the AWS account IDs. You can only share AMIs that have unencrypted volumes and volumes that are encrypted with a customer managed key. If you share an AMI with encrypted volumes, you must also share any customer managed keys used to encrypt them. For more information, see Share an Amazon EBS snapshot (p. 1234). You cannot share an AMI that has volumes that are encrypted with an AWS managed key.

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 Copy an AMI (p. 113).

There is no limit to the number of AWS accounts with which an AMI can be shared. User-defined tags that you attach to a shared AMI are available only to your AWS account and not to the other accounts that the AMI is shared with.

Share 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.

   Note
   You do not need to share the Amazon EBS snapshots that an AMI references in order to share the AMI. Only the AMI itself needs to be shared; the system automatically provides
the instance access to the referenced Amazon EBS snapshots for the launch. However, you do need to share any KMS keys used to encrypt snapshots that the AMI references. For more information, see Share an Amazon EBS snapshot (p. 1234).

5. Choose Save when you are done.

6. (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 Find shared AMIs (p. 102).

**Share 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-0abcdef1234567890 -Attribute launchPermission -OperationType add -UserId "123456789012"
```

**Note**
You do not need to share the Amazon EBS snapshots that an AMI references in order to share the AMI. Only the AMI itself needs to be shared; the system automatically provides the instance access to the referenced Amazon EBS snapshots for the launch. However, you do need to share any KMS keys used to encrypt snapshots that the AMI references. For more information, see Share an Amazon EBS snapshot (p. 1234).

**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-0abcdef1234567890 -Attribute launchPermission -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-0abcdef1234567890 -Attribute launchPermission
```

**Share an AMI (AWS CLI)**

Use the modify-image-attribute command (AWS CLI) to share an AMI as shown in the following examples.

**To grant explicit launch permissions**

The following command grants launch permissions for the specified AMI to the specified AWS account.

```
aws ec2 modify-image-attribute \\
  --image-id ami-0abcdef1234567890 \\
  --launch-permission "Add=[{UserId=123456789012}]"
```
Note
You do not need to share the Amazon EBS snapshots that an AMI references in order to share the AMI. Only the AMI itself needs to be shared; the system automatically provides the instance access to the referenced Amazon EBS snapshots for the launch. However, you do need to share any KMS keys used to encrypt snapshots that the AMI references. For more information, see Share an Amazon EBS snapshot (p. 1234).

To remove launch permissions for an account

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

```bash
aws ec2 modify-image-attribute \
  --image-id ami-0abcdef1234567890 \
  --launch-permission "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.

```bash
aws ec2 reset-image-attribute \
  --image-id ami-0abcdef1234567890 \
  --attribute launchPermission
```

Use 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:

   ```plaintext
   ```

   For example, this URL launches an instance from the ami-0abcdef1234567890 AMI in the us-east-1 Region:

   ```plaintext
   https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:ami=ami-0abcdef1234567890
   ```

2. Distribute the link to users who want to use your AMI.
3. To use a bookmark, choose the link or copy and paste it into your browser. The launch wizard opens, with the AMI already selected.

Best Practices 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 Install the latest version of EC2Config (p. 494).
• 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

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. 102).

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.

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 in AWS Marketplace. AWS Marketplace supports AMIs backed by Amazon EBS.

Contents
• Sell your AMI (p. 110)
• Find a paid AMI (p. 110)
• Purchase a paid AMI (p. 111)
• Get the product code for your instance (p. 112)
• Use paid support (p. 112)
Sell 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 the AWS Marketplace, see Selling in AWS Marketplace.

Find 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.

Find 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.

Find 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.

Find 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
------------------ -------------
product_code        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.
Find a paid AMI using the AWS CLI

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

```powershell
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:

```json
"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.

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

Purchase 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.

Purchase a paid AMI using the console

You can purchase a paid AMI by using the Amazon EC2 launch wizard. For more information, see Launch an AWS Marketplace instance (p. 412).

Subscribe 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 Launch an AWS Marketplace instance (p. 412).
Get 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. 579).

To retrieve a product code, use the following command:

```powershell
```

If the instance has a product code, Amazon EC2 returns it.

Use 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)
  ```powershell
  aws ec2 modify-image-attribute --image-id ami_id --product-codes "product_code"
  ```

- **Edit-EC2ImageAttribute** (AWS Tools for Windows PowerShell)
  ```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 products in the AWS Marketplace Buyer Guide.

Manage 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/](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.

**AMI lifecycle**

**Topics**
- Create an AMI (p. 113)
- Copy an AMI (p. 113)
- Store and restore an AMI using S3 (p. 119)
- Deprecate an AMI (p. 124)
- Automate the EBS-backed AMI lifecycle (p. 127)

**Create an AMI**

For information about how to create a Windows AMI, see *Create a custom Windows AMI* (p. 37).

For information about how to create a Linux AMI, see *Create an Amazon EBS-backed Linux AMI* and *Create an instance store-backed Linux AMI*.

**Copy an AMI**

You can copy an Amazon Machine Image (AMI) within or across AWS Regions using the AWS Management Console, the AWS Command Line Interface or SDKs, or the Amazon EC2 API, all of which support the **CopyImage** action. You can copy both Amazon EBS-backed AMIs and instance-store-backed AMIs. You can copy AMIs with encrypted snapshots and also change encryption status during the copy process.

Copying a source AMI results in an identical but distinct target AMI with its own unique identifier. You can change or deregister the source AMI with no effect on the target AMI. The reverse is also true.

With an Amazon EBS-backed AMI, each of its backing snapshots is copied to an identical but distinct target snapshot. If you copy an AMI to a new Region, the snapshots are complete (non-incremental) copies. If you encrypt unencrypted backing snapshots or encrypt them to a new KMS key, the snapshots are complete (non-incremental) copies. Subsequent copy operations of an AMI result in incremental copies of the backing snapshots.
There are no charges for copying an AMI. However, standard storage and data transfer rates apply. If you copy an EBS-backed AMI, you will incur charges for the storage of any additional EBS snapshots.

Considerations

- You can use IAM policies to grant or deny users permissions to copy AMIs. Resource-level permissions specified for the CopyImage action apply only to the new AMI. You cannot specify resource-level permissions for the source AMI.
- 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.
- If you are using an AWS Marketplace AMI, or an AMI that was directly or indirectly derived from an AWS Marketplace AMI, you cannot copy it across accounts. Instead, launch an EC2 instance using the AWS Marketplace AMI and then create an AMI from the instance. For more information, see Create a custom Windows AMI (p. 37).

Contents

- Permissions for copying an instance store-backed AMI (p. 114)
- Cross-Region copying (p. 115)
- Cross-account copying (p. 116)
- Encryption and copying (p. 116)
- Copy an AMI (p. 117)
- Stop a pending AMI copy operation (p. 118)

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.

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

**Note**
The `s3:CreateBucket` permission is only needed the first time that the IAM user copies an instance store-backed AMI to an individual Region. After that, the Amazon S3 bucket that is already created in the Region is used to store all future AMIs that you copy to that Region.

**Cross-Region copying**

Copying an AMI across geographically diverse Regions provides the following benefits:

- **Consistent global deployment:** Copying an AMI from one Region to another enables you to launch consistent instances in different Regions based on the same AMI.
- **Scalability:** You can more easily design and build global applications that meet the needs of your users, regardless of their location.
- **Performance:** You can increase performance by distributing your application, as well as locating critical components of your application in closer proximity to your users. You can also take advantage of Region-specific features, such as instance types or other AWS services.
- **High availability:** You can design and deploy applications across AWS Regions, to increase availability.

The following diagram shows the relations among a source AMI and two copied AMIs in different Regions, as well as the EC2 instances launched from each. When you launch an instance from an AMI, it resides in the same Region where the AMI resides. If you make changes to the source AMI and want those changes to be reflected in the AMIs in the target Regions, you must recopy the source AMI to the target Regions.

When you first copy an instance store-backed AMI to a Region, we create an Amazon S3 bucket for the AMIs copied to that Region. All instance store-backed AMIs that you copy to that Region are stored in this
bucket. The bucket names have the following format: amis-for-account-in-region-hash. For example: amis-for-123456789012-in-us-east-2-yhjmxvp6.

**Prerequisite**

Prior to copying an AMI, you must ensure that the contents of the source AMI are updated to support running in a different Region. For example, you should update any database connection strings or similar application configuration data to point to the appropriate resources. Otherwise, instances launched from the new AMI in the destination Region may still use the resources from the source Region, which can impact performance and cost.

**Limits**

- Destination Regions are limited to 100 concurrent AMI copies.

**Cross-account copying**

You can share an AMI with another AWS account. Sharing an AMI does not affect the ownership of the AMI. The owning account is charged for the storage in the Region. For more information, see [Share an AMI with specific AWS accounts](p. 106).

If you copy an AMI that has been shared with your account, you are the owner of the target AMI in your account. The owner of the source AMI is charged standard Amazon EBS or Amazon S3 transfer fees, and you are charged for the storage of the target AMI in the destination Region.

**Resource Permissions**

To copy an AMI that was shared with you from another account, the owner of the source AMI must grant you read permissions for the storage that backs the AMI, either the associated EBS snapshot (for an Amazon EBS-backed AMI) or an associated S3 bucket (for an instance store-backed AMI). If the shared AMI has encrypted snapshots, the owner must share the key or keys with you as well.

**Encryption and copying**

The following table shows encryption support for various AMI-copying scenarios. While it is possible to copy an unencrypted snapshot to yield an encrypted snapshot, you cannot copy an encrypted snapshot to yield an unencrypted one.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Unencrypted-to-unencrypted</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Encrypted-to-encrypted</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>Unencrypted-to-encrypted</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>Encrypted-to-unencrypted</td>
<td>No</td>
</tr>
</tbody>
</table>

**Note**

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

By default (i.e., without specifying encryption parameters), the backing snapshot of an AMI is copied with its original encryption status. Copying an AMI backed by an unencrypted snapshot results in an identical target snapshot that is also unencrypted. If the source AMI is backed by an encrypted snapshot, copying...
it results in an identical target snapshot that is encrypted by the same AWS KMS key. Copying an AMI backed by multiple snapshots preserves, by default, the source encryption status in each target snapshot.

If you specify encryption parameters while copying an AMI, you can encrypt or re-encrypt its backing snapshots. The following example shows a non-default case that supplies encryption parameters to the CopyImage action in order to change the target AMI's encryption state.

**Copy an unencrypted source AMI to an encrypted target AMI**

In this scenario, an AMI backed by an unencrypted root snapshot is copied to an AMI with an encrypted root snapshot. The CopyImage action is invoked with two encryption parameters, including a customer managed key. As a result, the encryption status of the root snapshot changes, so that the target AMI is backed by a root snapshot containing the same data as the source snapshot, but encrypted using the specified key. You incur storage costs for the snapshots in both AMIs, as well as charges for any instances you launch from either AMI.

**Note**

Enabling encryption by default (p. 1331) has the same effect as setting the Encrypted parameter to true for all snapshots in the AMI.

Set the Encrypted parameter encrypts the single snapshot for this instance. If you do not specify the KmsKeyId parameter, the default customer managed key is used to encrypt the snapshot copy.

For more information about copying AMIs with encrypted snapshots, see Use encryption with EBS-backed AMIs (p. 127).

**Copy 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 Create a custom Windows AMI (p. 37) and Finding an AMI.
To copy an AMI using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the console navigation bar, select the Region that contains the AMI. In the navigation pane, choose Images, AMIs to display the list of AMIs available to you in the Region.
3. Select the AMI to copy and choose Actions, Copy AMI.
4. In the Copy AMI dialog box, specify the following information and then choose Copy AMI:
   - **Destination region**: The Region in which to copy the AMI.
   - **Name**: A name for the new AMI. You can include operating system information in the name, as we do not provide this information when displaying details about the AMI.
   - **Description**: By default, the description includes information about the source AMI so that you can distinguish a copy from its original. You can change this description as needed.
   - **Encryption**: Select this field to encrypt the target snapshots, or to re-encrypt them using a different key. If you have enabled encryption by default (p. 1331), the Encryption option is set and cannot be unset.
   - **KMS Key**: The KMS key to use 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.

**Stop 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 *Access Amazon EC2* (p. 3).

- `deregister-image` (AWS CLI)
- `Unregister-EC2Image` (AWS Tools for Windows PowerShell)

**Store and restore an AMI using S3**

You can store an Amazon Machine Image (AMI) in an Amazon S3 bucket, copy the AMI to another S3 bucket, and then restore it from the S3 bucket. By storing and restoring an AMI using S3 buckets, you can copy AMIs from one AWS partition to another, for example, from the main commercial partition to the AWS GovCloud (US) partition. You can also make archival copies of AMIs by storing them in an S3 bucket.

The supported APIs for storing and restoring an AMI using S3 are `CreateStoreImageTask`, `DescribeStoreImageTasks`, and `CreateRestoreImageTask`.

`CopyImage` is the recommended API to use for copying AMIs *within* an AWS partition. However, `CopyImage` can't copy an AMI to another partition.

**Warning**

Ensure that you comply with all applicable laws and business requirements when moving data between AWS partitions or AWS Regions, including, but not limited to, any applicable government regulations and data residency requirements.

**Topics**

- Use cases (p. 119)
- How the AMI store and restore APIs work (p. 120)
- Limitations (p. 122)
- Costs (p. 122)
- Securing your AMIs (p. 122)
- Permissions for storing and restoring AMIs using S3 (p. 122)
- Work with the AMI store and restore APIs (p. 123)

**Use cases**

**Use the store and restore APIs to do the following:**

- Copy an AMI from one AWS partition to another AWS partition (p. 119)
- Make archival copies of AMIs (p. 120)

**Copy an AMI from one AWS partition to another AWS partition**

By storing and restoring an AMI using S3 buckets, you can copy an AMI from one AWS partition to another, or from one AWS Region to another. In the following example, you copy an AMI from the main commercial partition to the AWS GovCloud (US) partition, specifically from the `us-east-2` Region to the `us-gov-east-1` Region.
To copy an AMI from one partition to another, follow these steps:

- Store the AMI in an S3 bucket in the current Region by using `CreateStoreImageTask`. In this example, the S3 bucket is located in `us-east-2`. For an example command, see *Store an AMI in an S3 bucket* (p. 123).

- Monitor the progress of the store task by using `DescribeStoreImageTasks`. The object becomes visible in the S3 bucket when the task is completed. For an example command, see *Describe the progress of an AMI store task* (p. 123).

- Copy the stored AMI object to an S3 bucket in the target partition using a procedure of your choice. In this example, the S3 bucket is located in `us-gov-east-1`.

  **Note**

  Because you need different AWS credentials for each partition, you can't copy an S3 object directly from one partition to another. The process for copying an S3 object across partitions is outside the scope of this documentation. We provide the following copy processes as examples, but you must use the copy process that meets your security requirements.

  - To copy one AMI across partitions, the copy process could be as straightforward as the following: **Download the object** from the source bucket to an intermediate host (for example, an EC2 instance or a laptop), and then **upload the object** from the intermediate host to the source bucket. For each stage of the process, use the AWS credentials for the partition.

  - For more sustained usage, consider developing an application that manages the copies, potentially using S3 **multipart downloads and uploads**.

- Restore the AMI from the S3 bucket in the target partition by using `CreateRestoreImageTask`. In this example, the S3 bucket is located in `us-gov-east-1`. For an example command, see *Restore an AMI from an S3 bucket* (p. 124).

- Monitor the progress of the restore task by describing the AMI to check when its state becomes available. You can also monitor the progress percentages of the snapshots that make up the restored AMI by describing the snapshots.

**Make archival copies of AMIs**

You can make archival copies of AMIs by storing them in an S3 bucket. For an example command, see *Store an AMI in an S3 bucket* (p. 123).

The AMI is packed into a single object in S3, and all of the AMI metadata (excluding sharing information) is preserved as part of the stored AMI. The AMI data is compressed as part of the storage process. AMIs that contain data that can easily be compressed will result in smaller objects in S3. To reduce costs, you can use less expensive S3 storage tiers. For more information, see Amazon S3 Storage Classes and Amazon S3 pricing

**How the AMI store and restore APIs work**

To store and restore an AMI using S3, you use the following APIs:

- `CreateStoreImageTask` – Stores the AMI in an S3 bucket
- `DescribeStoreImageTasks` – Provides the progress of the AMI store task
- `CreateRestoreImageTask` – Restores the AMI from an S3 bucket

**How the APIs work**

- `CreateStoreImageTask (p. 121)`
- `DescribeStoreImageTasks (p. 121)`
- `CreateRestoreImageTask (p. 121)`
CreateStoreImageTask

The CreateStoreImageTask (p. 123) API stores an AMI as a single object in an S3 bucket.

The API creates a task that reads all of the data from the AMI and its snapshots, and then uses an S3 multipart upload to store the data in an S3 object. The API takes all of the components of the AMI, including most of the non-Region-specific AMI metadata, and all the EBS snapshots contained in the AMI, and packs them into a single object in S3. The data is compressed as part of the upload process to reduce the amount of space used in S3, so the object in S3 might be smaller than the sum of the sizes of the snapshots in the AMI.

If there are AMI and snapshot tags visible to the account calling this API, they are preserved.

The object in S3 has the same ID as the AMI, but with a .bin extension. The following data is also stored as S3 metadata tags on the S3 object: AMI name, AMI description, AMI registration date, AMI owner account, and a timestamp for the store operation.

The time it takes to complete the task depends on the size of the AMI. It also depends on how many other tasks are in progress because tasks are queued. You can track the progress of the task by calling the DescribeStoreImageTasks (p. 123) API.

The sum of the sizes of all the AMIs in progress is limited to 600 GB of EBS snapshot data per account. Further task creation will be rejected until the tasks in progress are less than the limit. For example, if an AMI with 100 GB of snapshot data and another AMI with 200 GB of snapshot data are currently being stored, another request will be accepted, because the total in progress is 300 GB, which is less than the limit. But if a single AMI with 800 GB of snapshot data is currently being stored, further tasks are rejected until the task is completed.

DescribeStoreImageTasks

The DescribeStoreImageTasks (p. 123) API describes the progress of the AMI store tasks. You can describe tasks for specified AMIs. If you don’t specify AMIs, you get a paginated list of all of the store image tasks that have been processed in the last 31 days.

For each AMI task, the response indicates if the task is InProgress, Completed, or Failed. For tasks InProgress, the response shows an estimated progress as a percentage.

Tasks are listed in reverse chronological order.

Currently, only tasks from the previous month can be viewed.

CreateRestoreImageTask

The CreateRestoreImageTask (p. 124) API starts a task that restores an AMI from an S3 object that was previously created by using a CreateStoreImageTask (p. 123) request.

The restore task can be performed in the same or a different Region in which the store task was performed.

The S3 bucket from which the AMI object will be restored must be in the same Region in which the restore task is requested. The AMI will be restored in this Region.

The AMI is restored with its metadata, such as the name, description, and block device mappings corresponding to the values of the stored AMI. The name must be unique for AMIs in the Region for this account. If you do not provide a name, the new AMI gets the same name as the original AMI. The AMI gets a new AMI ID that is generated at the time of the restore process.

The time it takes to complete the AMI restoration task depends on the size of the AMI. It also depends on how many other tasks are in progress because tasks are queued. You can view the progress of the task by describing the AMI (describe-images) or its EBS snapshots (describe-snapshots). If the task fails, the AMI and snapshots are moved to a failed state.
The sum of the sizes of all of the AMIs in progress is limited to 300 GB (based on the size after restoration) of EBS snapshot data per account. Further task creation will be rejected until the tasks in progress are less than the limit.

Limitations

- Only EBS-backed AMIs can be stored using these APIs.
- Paravirtual (PV) AMIs are not supported.
- The size of an AMI (before compression) that can be stored is limited to the size limit of a single S3 object, which is 1 TB.
- Quota on store image (p. 123) requests: 600 GB of storage work (snapshot data) in progress.
- Quota on restore image (p. 124) requests: 300 GB of restore work (snapshot data) in progress.
- For the duration of the store task, the snapshots must not be deleted and the IAM principal doing the store must have access to the snapshots, otherwise the store process will fail.
- You can’t create multiple copies of an AMI in the same S3 bucket.
- An AMI that is stored in an S3 bucket can’t be restored with its original AMI ID. You can mitigate this by using AMI aliasing.
- Currently the store and restore APIs are only supported by using the AWS Command Line Interface, AWS SDKs, and Amazon EC2 API. You can’t store and restore an AMI using the Amazon EC2 console.

Costs

When you store and restore AMIs using S3, you are charged for the services that are used by the store and restore APIs, and for data transfer. The APIs use S3 and the EBS Direct API (used internally by these APIs to access the snapshot data). For more information, see Amazon S3 pricing and Amazon EBS pricing.

Securing your AMIs

To use the store and restore APIs, the S3 bucket and the AMI must be in the same Region. It is important to ensure that the S3 bucket is configured with sufficient security to secure the content of the AMI and that the security is maintained for as long as the AMI objects remain in the bucket. If this can’t be done, use of these APIs is not recommended. Ensure that public access to the S3 bucket is not allowed. We recommend enabling Server Side Encryption for the S3 buckets in which you store the AMIs, although it’s not required.

For information about how to set the appropriate security settings for your S3 buckets, review the following security topics:

- Blocking public access to your Amazon S3 storage
- Setting default server-side encryption behavior for Amazon S3 buckets
- What S3 bucket policy should I use to comply with the AWS Config rule s3-bucket-ssl-requests-only?
- Enabling Amazon S3 server access logging

When the AMI snapshots are copied to the S3 object, the data is then copied over TLS connections. You can store AMIs with encrypted snapshots, but the snapshots are decrypted as part of the store process.

Permissions for storing and restoring AMIs using S3

If your IAM principals will store or restore AMIs using S3, you need to grant them the required permissions.

The following example policy includes all of the actions that are required to allow an IAM principal to carry out the store and restore tasks.
You can also craft policies so that IAM principals can only access named resources. For more example policies, see Access management for AWS resources in the IAM User Guide.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "s3:DeleteObject",
            "s3:GetObject",
            "s3:ListBucket",
            "s3:PutObject",
            "s3:AbortMultipartUpload",
            "ebs:CompleteSnapshot",
            "ebs:GetSnapshotBlock",
            "ebs:ListChangedBlocks",
            "ebs:ListSnapshotBlocks",
            "ebs:PutSnapshotBlock",
            "ebs:StartSnapshot",
            "ec2:CreateStoreImageTask",
            "ec2:DescribeStoreImageTasks",
            "ec2:CreateRestoreImageTask",
            "ec2:GetEbsEncryptionByDefault",
            "ec2:DescribeTags"
         ],
         "Resource": "*"
      }
   ]
}
```

Work with the AMI store and restore APIs

**Topics**

- Store an AMI in an S3 bucket (p. 123)
- Describe the progress of an AMI store task (p. 123)
- Restore an AMI from an S3 bucket (p. 124)

**Store an AMI in an S3 bucket**

**To store an AMI (AWS CLI)**

Use the `create-store-image-task` command. Specify the ID of the AMI and the name of the S3 bucket in which to store the AMI.

```
aws ec2 create-store-image-task \  
   --image-id ami-1234567890abcdef0 \  
   --bucket myamibucket
```

**Expected output**

```
{
   "ObjectKey": "ami-1234567890abcdef0.bin"
}
```

**Describe the progress of an AMI store task**

**To describe the progress of an AMI store task (AWS CLI)**
Use the `describe-store-image-tasks` command.

```bash
aws ec2 describe-store-image-tasks
```

Expected output

```
{
  "AmiId": "ami-1234567890abcdef0",
  "Bucket": "myamibucket",
  "ProgressPercentage": 17,
  "S3ObjectKey": "ami-1234567890abcdef0.bin",
  "StoreTaskState": "InProgress",
  "StoreTaskFailureReason": null,
  "TaskStartTime": "2021-01-01T01:01:01.001Z"
}
```

### Restore an AMI from an S3 bucket

**To restore an AMI (AWS CLI)**

Use the `create-restore-image-task` command. Using the values for `S3ObjectKey` and `Bucket` from the `describe-store-image-tasks` output, specify the object key of the AMI and the name of the S3 bucket to which the AMI was copied. Also specify a name for the restored AMI. The name must be unique for AMIs in the Region for this account.

**Note**
The restored AMI gets a new AMI ID.

```bash
aws ec2 create-restore-image-task \\   --object-key ami-1234567890abcdef0.bin \\   --bucket myamibucket \\   --name "New AMI Name"
```

Expected output

```
{
  "ImageId": "ami-0eab20fe36f83e1a8"
}
```

---

### Deprecate an AMI

You can deprecate an AMI to indicate that it is out of date and should not be used. You can also specify a future deprecation date for an AMI, indicating when the AMI will be out of date. For example, you might deprecate an AMI that is no longer actively maintained, or you might deprecate an AMI that has been superseded by a newer version. By default, deprecated AMIs do not appear in AMI listings, preventing new users from using out-of-date AMIs. However, existing users and launch services, such as launch templates and Auto Scaling groups, can continue to use a deprecated AMI by specifying its ID. To delete the AMI so that users and services cannot use it, you must `deregister (p. 51)` it.

After an AMI is deprecated:

- For AMI users, the deprecated AMI does not appear in `DescribelImages` API calls unless you specify its ID or specify that deprecated AMIs must appear. AMI owners continue to see deprecated AMIs in `DescribelImages` API calls.
- For AMI users, the deprecated AMI is not available to select via the EC2 console. For example, a deprecated AMI does not appear in the AMI catalog in the launch instance wizard. AMI owners continue to see deprecated AMIs in the EC2 console.
For AMI users, if you know the ID of a deprecated AMI, you can continue to launch instances using the deprecated AMI by using the API, CLI, or the SDKs. Launch services, such as launch templates and Auto Scaling groups, can continue to reference deprecated AMIs. EC2 instances that were launched using an AMI that is subsequently deprecated are not affected, and can be stopped, started, and rebooted.

You can deprecate both private and public AMIs. You can also create Amazon Data Lifecycle Manager EBS-backed AMI policies to automate the deprecation of EBS-backed AMIs. For more information, see Automate AMI lifecycles (p. 1284).

Topics
- Costs (p. 125)
- Limitations (p. 122)
- Deprecate an AMI (p. 125)
- Describe deprecated AMIs (p. 126)
- Cancel the deprecation of an AMI (p. 127)

Costs
When you deprecate an AMI, the AMI is not deleted. The AMI owner continues to pay for the AMI's snapshots. To stop paying for the snapshots, the AMI owner must delete the AMI by deregistering (p. 51) it.

Limitations
- To deprecate an AMI, you must be the owner of the AMI.
- You can't use the EC2 console to deprecate an AMI or to cancel the deprecation of an AMI.

Deprecate an AMI
You can deprecate an AMI on a specific date and time. You must be the AMI owner to perform this procedure.

To deprecate an AMI on a specific date (AWS CLI)
Use the `enable-image-deprecation` command. Specify the ID of the AMI and the date and time on which to deprecate the AMI. If you specify a value for seconds, Amazon EC2 rounds the seconds to the nearest minute.

```bash
aws ec2 enable-image-deprecation
    --image-id ami-1234567890abcdef0
    --deprecate-at "2021-10-15T13:17:12.000Z"
```

Expected output

```json
{
    "RequestID": "59dbff89-35bd-4eac-99ed-be587EXAMPLE",
    "Return": "true"
}
```
Describe deprecated AMIs

When you describe all AMIs using the describe-images command, the results are different depending on whether you are an AMI user or the AMI owner.

- If you are an AMI user:

  By default, when you describe all AMIs using the describe-images command, deprecated AMIs that are not owned by you, but which are shared with you, do not appear in the results. To include deprecated AMIs in the results, you must specify the --include-deprecated true parameter. The default value for --include-deprecated is false. If you omit this parameter, deprecated AMIs do not appear in the results.

- If you are the AMI owner:

  When you describe all AMIs using the describe-images command, all the AMIs that you own, including deprecated AMIs, appear in the results. You do not need to specify the --include-deprecated true parameter. Furthermore, you cannot exclude deprecated AMIs that you own from the results by using --include-deprecated false.

If an AMI is deprecated, the DeprecationTime field appears in the results.

Note
A deprecated AMI is an AMI whose deprecation date is in the past. If you have set the deprecation date to a date in the future, the AMI is not yet deprecated.

To include all deprecated AMIs when describing all AMIs (AWS CLI)

Use the describe-images command and specify the --include-deprecated parameter with a value of true to include all deprecated AMIs that are not owned by you in the results.

```bash
aws ec2 describe-images \
  --region us-east-1 \
  --owners 123456example \
  --include-deprecated true
```

To describe the deprecation date of an AMI (AWS CLI)

Use the describe-images command and specify the ID of the AMI.

Note that if you specify --include-deprecated false together with the AMI ID, the deprecated AMI will be returned in the results.

```bash
aws ec2 describe-images \
  --region us-east-1 \
  --image-ids ami-1234567890EXAMPLE
```

Expected output

The DeprecationTime field displays the date on which the AMI is set to be deprecated. If the AMI is not set to be deprecated, the DeprecationTime field does not appear in the output.

```json
{
  "Images": [
    {
      "VirtualizationType": "hvm",
      "Description": "Provided by Red Hat, Inc.",
      "PlatformDetails": "Red Hat Enterprise Linux",
      "EnaSupport": true,
      "Hypervisor": "xen",
      "DeprecationTime": "2023-01-01T00:00:00Z"
    }
  ]
}
```
Automate the EBS-backed AMI lifecycle

You can use Amazon Data Lifecycle Manager to automate the creation, retention, copy, deprecation, and deregistration of Amazon EBS-backed AMIs and their backing snapshots. For more information, see Amazon Data Lifecycle Manager (p. 1272).

Use encryption with EBS-backed AMIs

AMIs that are backed by Amazon EBS snapshots can take advantage of Amazon EBS encryption. Snapshots of both data and root volumes can be encrypted and attached to an AMI. You can launch...
instances and copy images with full EBS encryption support included. Encryption parameters for these operations are supported in all Regions where AWS KMS is available.

EC2 instances with encrypted EBS volumes are launched from AMIs in the same way as other instances. In addition, when you launch an instance from an AMI backed by unencrypted EBS snapshots, you can encrypt some or all of the volumes during launch.

Like EBS volumes, snapshots in AMIs can be encrypted by either your default AWS KMS key, or to a customer managed key that you specify. You must in all cases have permission to use the selected KMS key.

AMIs with encrypted snapshots can be shared across AWS accounts. For more information, see Shared AMIs (p. 102).

Encryption with EBS-backed AMIs topics
- Instance-launching scenarios (p. 128)
- Image-copying scenarios (p. 131)

Instance-launching scenarios

Amazon EC2 instances are launched from AMIs using the RunInstances action with parameters supplied through block device mapping, either by means of the AWS Management Console or directly using the Amazon EC2 API or CLI. For more information about block device mapping, see Block device mapping. For examples of controlling block device mapping from the AWS CLI, see Launch, List, and Terminate EC2 Instances.

By default, without explicit encryption parameters, a RunInstances action maintains the existing encryption state of an AMI's source snapshots while restoring EBS volumes from them. If Encryption by default (p. 1331) is enabled, all volumes created from the AMI (whether from encrypted or unencrypted snapshots) will be encrypted. If encryption by default is not enabled, then the instance maintains the encryption state of the AMI.

You can also launch an instance and simultaneously apply a new encryption state to the resulting volumes by supplying encryption parameters. Consequently, the following behaviors are observed:

Launch with no encryption parameters
- An unencrypted snapshot is restored to an unencrypted volume, unless encryption by default is enabled, in which case all the newly created volumes will be encrypted.
- An encrypted snapshot that you own is restored to a volume that is encrypted to the same KMS key.
- An encrypted snapshot that you do not own (for example, the AMI is shared with you) is restored to a volume that is encrypted by your AWS account's default KMS key.

The default behaviors can be overridden by supplying encryption parameters. The available parameters are Encrypted and KmsKeyId. Setting only the Encrypted parameter results in the following:

Instance launch behaviors with Encrypted set, but no KmsKeyId specified
- An unencrypted snapshot is restored to an EBS volume that is encrypted by your AWS account's default KMS key.
- An encrypted snapshot that you own is restored to an EBS volume encrypted by the same KMS key. (In other words, the Encrypted parameter has no effect.)
- An encrypted snapshot that you do not own (i.e., the AMI is shared with you) is restored to a volume that is encrypted by your AWS account's default KMS key. (In other words, the Encrypted parameter has no effect.)
Setting both the `Encrypted` and `KmsKeyId` parameters allows you to specify a non-default KMS key for an encryption operation. The following behaviors result:

**Instance with both `Encrypted` and `KmsKeyId` set**

- An unencrypted snapshot is restored to an EBS volume encrypted by the specified KMS key.
- An encrypted snapshot is restored to an EBS volume encrypted not to the original KMS key, but instead to the specified KMS key.

Submitting a `KmsKeyId` without also setting the `Encrypted` parameter results in an error.

The following sections provide examples of launching instances from AMIs using non-default encryption parameters. In each of these scenarios, parameters supplied to the `RunInstances` action result in a change of encryption state during restoration of a volume from a snapshot.

For information about using the console to launch an instance from an AMI, see [Launch your instance](p. 390).

### Encrypt a volume during launch

In this example, an AMI backed by an unencrypted snapshot is used to launch an EC2 instance with an encrypted EBS volume.

The `Encrypted` parameter alone results in the volume for this instance being encrypted. Providing a `KmsKeyId` parameter is optional. If no KMS key ID is specified, the AWS account's default KMS key is used to encrypt the volume. To encrypt the volume to a different KMS key that you own, supply the `KmsKeyId` parameter.

### Re-encrypt a volume during launch

In this example, an AMI backed by an encrypted snapshot is used to launch an EC2 instance with an EBS volume encrypted by a new KMS key.
If you own the AMI and supply no encryption parameters, the resulting instance has a volume encrypted by the same KMS key as the snapshot. If the AMI is shared rather than owned by you, and you supply no encryption parameters, the volume is encrypted by your default KMS key. With encryption parameters supplied as shown, the volume is encrypted by the specified KMS key.

**Change encryption state of multiple volumes during launch**

In this more complex example, an AMI backed by multiple snapshots (each with its own encryption state) is used to launch an EC2 instance with a newly encrypted volume and a re-encrypted volume.
In this scenario, the `RunInstances` action is supplied with encryption parameters for each of the source snapshots. When all possible encryption parameters are specified, the resulting instance is the same regardless of whether you own the AMI.

### Image-copying scenarios

Amazon EC2 AMIs are copied using the `CopyImage` action, either through the AWS Management Console or directly using the Amazon EC2 API or CLI.

By default, without explicit encryption parameters, a `CopyImage` action maintains the existing encryption state of an AMI’s source snapshots during copy. You can also copy an AMI and simultaneously apply a new encryption state to its associated EBS snapshots by supplying encryption parameters. Consequently, the following behaviors are observed:

#### Copy with no encryption parameters

- An unencrypted snapshot is copied to another unencrypted snapshot, unless encryption by default is enabled, in which case all the newly created snapshots will be encrypted.
- An encrypted snapshot that you own is copied to a snapshot encrypted with the same KMS key.
- An encrypted snapshot that you do not own (that is, the AMI is shared with you) is copied to a snapshot that is encrypted by your AWS account’s default KMS key.
All of these default behaviors can be overridden by supplying encryption parameters. The available parameters are Encrypted and KmsKeyId. Setting only the Encrypted parameter results in the following:

**Copy-image behaviors with Encrypted set, but no KmsKeyId specified**

- An unencrypted snapshot is copied to a snapshot encrypted by the AWS account's default KMS key.
- An encrypted snapshot is copied to a snapshot encrypted by the same KMS key. (In other words, the Encrypted parameter has no effect.)
- An encrypted snapshot that you do not own (i.e., the AMI is shared with you) is copied to a volume that is encrypted by your AWS account’s default KMS key. (In other words, the Encrypted parameter has no effect.)

Setting both the Encrypted and KmsKeyId parameters allows you to specify a customer managed KMS key for an encryption operation. The following behaviors result:

**Copy-image behaviors with both Encrypted and KmsKeyId set**

- An unencrypted snapshot is copied to a snapshot encrypted by the specified KMS key.
- An encrypted snapshot is copied to a snapshot encrypted not to the original KMS key, but instead to the specified KMS key.

Submitting a KmsKeyId without also setting the Encrypted parameter results in an error.

The following section provides an example of copying an AMI using non-default encryption parameters, resulting in a change of encryption state.

For detailed instructions using the console, see Copy an AMI (p. 113).

### Encrypt an unencrypted image during copy

In this scenario, an AMI backed by an unencrypted root snapshot is copied to an AMI with an encrypted root snapshot. The CopyImage action is invoked with two encryption parameters, including a customer managed key. As a result, the encryption status of the root snapshot changes, so that the target AMI is backed by a root snapshot containing the same data as the source snapshot, but encrypted using the specified key. You incur storage costs for the snapshots in both AMIs, as well as charges for any instances you launch from either AMI.

**Note**

Enabling encryption by default (p. 1331) has the same effect as setting the Encrypted parameter to true for all snapshots in the AMI.
Setting the `Encrypted` parameter encrypts the single snapshot for this instance. If you do not specify the `KmsKeyId` parameter, the default customer managed key is used to encrypt the snapshot copy.

**Note**

You can also copy an image with multiple snapshots and configure the encryption state of each individually.

---

### Understand AMI billing information

There are many Amazon Machine Images (AMIs) to choose from when launching your instances, and they support a variety of operating system platforms and features. To understand how the AMI you choose when launching your instance affects the bottom line on your AWS bill, you can research the associated operating system platform and billing information. Do this before you launch any On-Demand or Spot Instances, or purchase a Reserved Instance.

Here are two examples of how researching your AMI in advance can help you choose the AMI that best suits your needs:

- For Spot Instances, you can use the AMI **Platform details** to confirm that the AMI is supported for Spot Instances.
- When purchasing a Reserved Instance, you can make sure that you select the operating system platform (**Platform**) that maps to the AMI **Platform details**.

For more information about instance pricing, see [Amazon EC2 pricing](https://aws.amazon.com/ec2/pricing/).

**Contents**

- AMI billing information fields (p. 134)
- Finding AMI billing and usage details (p. 135)
- Verify AMI charges on your bill (p. 136)
AMI billing information fields

The following fields provide billing information associated with an AMI:

Platform details

The platform details associated with the billing code of the AMI. For example, Red Hat Enterprise Linux.

Usage operation

The operation of the Amazon EC2 instance and the billing code that is associated with the AMI. For example, RunInstances:0010. **Usage operation** corresponds to the lineitem/Operation column on your AWS Cost and Usage Report (CUR) and in the AWS Price List API.

You can view these fields on the Instances or AMIs page in the Amazon EC2 console, or in the response that is returned by the describe-images command.

Sample data: usage operation by platform

The following table lists some of the platform details and usage operation values that can be displayed on the Instances or AMIs pages in the Amazon EC2 console, or in the response that is returned by the describe-images command.

<table>
<thead>
<tr>
<th>Platform details</th>
<th>Usage operation **</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux/UNIX</td>
<td>RunInstances</td>
</tr>
<tr>
<td>Red Hat BYOL Linux</td>
<td>RunInstances:00g0</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux</td>
<td>RunInstances:0010</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux with HA</td>
<td>RunInstances:1010</td>
</tr>
<tr>
<td>SQL Server Enterprise</td>
<td>RunInstances:0100</td>
</tr>
<tr>
<td>SQL Server Standard</td>
<td>RunInstances:0004</td>
</tr>
<tr>
<td>SQL Server Web</td>
<td>RunInstances:0200</td>
</tr>
<tr>
<td>SUSE Linux</td>
<td>RunInstances:000g</td>
</tr>
<tr>
<td>Windows</td>
<td>RunInstances:0002</td>
</tr>
<tr>
<td>Windows BYOL</td>
<td>RunInstances:0800</td>
</tr>
<tr>
<td>Windows with SQL Server Enterprise</td>
<td>RunInstances:0102</td>
</tr>
<tr>
<td>Windows with SQL Server Standard</td>
<td>RunInstances:0006</td>
</tr>
<tr>
<td>Windows with SQL Server Web</td>
<td>RunInstances:0202</td>
</tr>
</tbody>
</table>

* If two software licenses are associated with an AMI, the Platform details field shows both.

** If you are running Spot Instances, the lineitem/Operation on your AWS Cost and Usage Report might be different from the Usage operation value that is listed here. For example, if lineitem/Operation displays RunInstances:0010:SV006, it means that Amazon EC2 is running Red Hat Enterprise Linux Spot Instance-hour in US East (Virginia) in VPC Zone #6.
Finding AMI billing and usage details

In the Amazon EC2 console, you can view the AMI billing information from the AMIs page or from the Instances page. You can also find billing information using the AWS CLI or the instance metadata service.

The following fields can help you verify AMI charges on your bill:

- Platform details
- Usage operation
- AMI ID

Find AMI billing information (console)

Follow these steps to view AMI billing information in the Amazon EC2 console:

**Look up AMI billing information from the AMIs page**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose AMIs, and then select an AMI.
3. On the Details tab, check the values for Platform details and Usage operation.

**Look up AMI billing information from the Instances page**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances, and then select an instance.
3. On the Details tab (or the Description tab if you are using the prior version of the console), check the values for Platform details and Usage operation.

Find AMI billing information (AWS CLI)

To find the AMI billing information using the AWS CLI, you need to know the AMI ID. If you don't know the AMI ID, you can get it from the instance using the describe-instances command.

**To find the AMI ID**

If you know the instance ID, you can get the AMI ID for the instance by using the `describe-instances` command.

```
aws ec2 describe-instances --instance-ids i-123456789abcde123
```

In the output, the AMI ID is specified in the ImageId field.

```
"Instances": [
  {
    "AmiLaunchIndex": 0,
    "ImageId": "ami-0123456789EXAMPLE",
    "InstanceId": "i-123456789abcde123",
    ...
  }
]
```

**To find the AMI billing information**
If you know the AMI ID, you can use the `describe-images` command to get the AMI platform and usage operation details.

```bash
$ aws ec2 describe-images --image-ids ami-0123456789EXAMPLE
```

The following example output shows the `PlatformDetails` and `UsageOperation` fields. In this example, the ami-0123456789EXAMPLE platform is Red Hat Enterprise Linux and the usage operation and billing code is RunInstances:0010.

```
{
   "Images": [ 
       {
           "VirtualizationType": "hvm",
           "Description": "Provided by Red Hat, Inc.",
           "Hypervisor": "xen",
           "EnaSupport": true,
           "SriovNetSupport": "simple",
           "ImageId": "ami-0123456789EXAMPLE",
           "State": "available",
           "BlockDeviceMappings": [ 
               {
                   "DeviceName": "/dev/sda1",
                   "Ebs": {
                       "SnapshotId": "snap-111223344aaabb",
                       "DeleteOnTermination": true,
                       "VolumeType": "gp2",
                       "VolumeSize": 10,
                       "Encrypted": false
                   }
               }
           ],
           "Architecture": "x86_64",
           "ImageLocation": "123456789012/RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2",
           "RootDeviceType": "ebs",
           "OwnerId": "123456789012",
           "PlatformDetails": "Red Hat Enterprise Linux",
           "UsageOperation": "RunInstances:0010",
           "RootDeviceName": "/dev/sda1",
           "CreationDate": "2019-05-10T13:17:12.000Z",
           "Public": true,
           "ImageType": "machine",
           "Name": "RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2"
       }
   ]
}
```

### Verify AMI charges on your bill

To ensure that you're not incurring unplanned costs, you can verify that the billing information for an instance in your AWS Cost and Usage Report (CUR) matches the billing information that's associated with the AMI that you used to launch the instance.

To verify the billing information, find the instance ID in your CUR and check the corresponding value in the `lineitem/Operation` column. That value should match the value for `Usage operation` that's associated with the AMI.

For example, the AMI `ami-0123456789EXAMPLE` has the following billing information:

- **Platform details** = Red Hat Enterprise Linux
- **Usage operation** = RunInstances:0010
If you launched an instance using this AMI, you can find the instance ID in your CUR, and check the corresponding value in the `lineitem/Operation` column. In this example, the value should be `RunInstances:0010`. 
Amazon EC2 instances

If you’re new to Amazon EC2, see the following topics to get started:

- What is Amazon EC2? (p. 1)
- Set up to use Amazon EC2 (p. 5)
- Tutorial: Get started with Amazon EC2 Windows instances (p. 9)
- Instance lifecycle (p. 386)

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. 141).

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. 239).

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

AWS Systems Manager enables you to remotely and securely manage the configuration of your Amazon EC2 instances, and your on-premises instances and virtual machines (VMs) in hybrid environments, including VMs from other cloud providers. For more information, see the AWS Systems Manager User Guide.

Amazon EC2 Windows instances

The following is an introduction to key components of Amazon EC2 and how a Windows instance compares to running Windows Server on premises.

Instances and AMIs

An Amazon Machine Image (AMI) is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch 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.

### Differences between Windows Server and Windows instances

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.

<table>
<thead>
<tr>
<th>Windows Server</th>
<th>Amazon EC2 Windows Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources and capacity are physically limited.</td>
<td>Resources and capacity are scalable.</td>
</tr>
<tr>
<td>You pay for the infrastructure, even if you don’t use it.</td>
<td>You pay for the usage of the infrastructure. We stop charging you for the instance as soon as you stop or terminate it.</td>
</tr>
<tr>
<td>Occupies physical space and must be maintained on a regular basis.</td>
<td>Doesn't occupy physical space and does not require regular maintenance.</td>
</tr>
<tr>
<td>Windows Server</td>
<td>Amazon EC2 Windows Instance</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>Starts with push of the power button (known as cold booting).</td>
<td>Starts with the launch of the instance.</td>
</tr>
<tr>
<td>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).</td>
<td>You can keep the server running, or stop and restart it (during which the instance is moved to a new host computer).</td>
</tr>
<tr>
<td>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.</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

![State Diagram for Windows Server](image)

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.

![State Diagram for Amazon EC2 Windows Instance](image)
Design your applications to run on 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.

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.

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 is grouped in an instance family 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. 142)
- Hardware specifications (p. 145)
- Instances built on the Nitro System (p. 146)
- Networking and storage features (p. 147)
- Instance limits (p. 150)
- General purpose instances (p. 150)
- Compute optimized instances (p. 193)
- Memory optimized instances (p. 199)
- Storage optimized instances (p. 210)
- Windows accelerated computing instances (p. 216)
- Find an Amazon EC2 instance type (p. 230)
- Change the instance type (p. 231)
- Get recommendations for an instance type (p. 236)

Available instance types

Amazon EC2 provides a wide selection of instance types optimized for different use cases. To determine which instance types meet your requirements, such as supported Regions, compute resources, or storage resources, see Find an Amazon EC2 instance type (p. 230).

Current generation instances

For the best performance, we recommend that you use the following instance types when you launch new instances. For more information, see Amazon EC2 Instance Types.

<table>
<thead>
<tr>
<th>Type</th>
<th>Sizes</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4</td>
<td>c4.large</td>
<td>Compute optimized (p. 193)</td>
</tr>
<tr>
<td>C5</td>
<td>c5.large</td>
<td>Compute optimized (p. 193)</td>
</tr>
<tr>
<td>C5a</td>
<td>c5a.large</td>
<td>Compute optimized (p. 193)</td>
</tr>
<tr>
<td>C5ad</td>
<td>c5ad.large</td>
<td>Compute optimized (p. 193)</td>
</tr>
<tr>
<td>C5d</td>
<td>c5d.large</td>
<td>Compute optimized (p. 193)</td>
</tr>
<tr>
<td>C5n</td>
<td>c5n.large</td>
<td>Compute optimized (p. 193)</td>
</tr>
<tr>
<td>Type</td>
<td>Sizes</td>
<td>Use case</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>D2</td>
<td>d2.xlarge</td>
<td>Storage optimized (p. 210)</td>
</tr>
<tr>
<td>D3</td>
<td>d3.xlarge</td>
<td>Storage optimized (p. 210)</td>
</tr>
<tr>
<td>D3en</td>
<td>d3en.xlarge</td>
<td>Storage optimized (p. 210)</td>
</tr>
<tr>
<td>F1</td>
<td>f1.2xlarge</td>
<td>Accelerated computing (p. 216)</td>
</tr>
<tr>
<td>G3</td>
<td>g3s.xlarge</td>
<td>Accelerated computing (p. 216)</td>
</tr>
<tr>
<td>G4ad</td>
<td>g4ad.xlarge</td>
<td>Accelerated computing (p. 216)</td>
</tr>
<tr>
<td>G4dn</td>
<td>g4dn.xlarge</td>
<td>Accelerated computing (p. 216)</td>
</tr>
<tr>
<td>H1</td>
<td>h1.2xlarge</td>
<td>Storage optimized (p. 210)</td>
</tr>
<tr>
<td>I3</td>
<td>i3.large</td>
<td>Storage optimized (p. 210)</td>
</tr>
<tr>
<td>I3en</td>
<td>i3en.xlarge</td>
<td>Storage optimized (p. 210)</td>
</tr>
<tr>
<td>M4</td>
<td>m4.large</td>
<td>General purpose (p. 150)</td>
</tr>
<tr>
<td>M5</td>
<td>m5.large</td>
<td>General purpose (p. 150)</td>
</tr>
<tr>
<td>M5a</td>
<td>m5a.large</td>
<td>General purpose (p. 150)</td>
</tr>
<tr>
<td>M5ad</td>
<td>m5ad.large</td>
<td>General purpose (p. 150)</td>
</tr>
<tr>
<td>M5d</td>
<td>m5d.large</td>
<td>General purpose (p. 150)</td>
</tr>
<tr>
<td>M5dn</td>
<td>m5dn.large</td>
<td>General purpose (p. 150)</td>
</tr>
<tr>
<td>M5n</td>
<td>m5n.large</td>
<td>General purpose (p. 150)</td>
</tr>
<tr>
<td>M5zn</td>
<td>m5zn.large</td>
<td>General purpose (p. 150)</td>
</tr>
<tr>
<td>M6i</td>
<td>m6i.large</td>
<td>General purpose (p. 150)</td>
</tr>
<tr>
<td>Type</td>
<td>Sizes</td>
<td>Use case</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>P2</td>
<td>p2.xlarge</td>
<td>Accelerated computing (p. 216)</td>
</tr>
<tr>
<td>P3</td>
<td>p3.2xlarge</td>
<td>Accelerated computing (p. 216)</td>
</tr>
<tr>
<td>P3dn</td>
<td>p3dn.24xlarge</td>
<td>Accelerated computing (p. 216)</td>
</tr>
<tr>
<td>R4</td>
<td>r4.large</td>
<td>Memory optimized (p. 199)</td>
</tr>
<tr>
<td>R5</td>
<td>r5.large</td>
<td>Memory optimized (p. 199)</td>
</tr>
<tr>
<td>R5a</td>
<td>r5a.large</td>
<td>Memory optimized (p. 199)</td>
</tr>
<tr>
<td>R5ad</td>
<td>r5ad.large</td>
<td>Memory optimized (p. 199)</td>
</tr>
<tr>
<td>R5b</td>
<td>r5b.large</td>
<td>Memory optimized (p. 199)</td>
</tr>
<tr>
<td>R5d</td>
<td>r5d.large</td>
<td>Memory optimized (p. 199)</td>
</tr>
<tr>
<td>R5dn</td>
<td>r5dn.large</td>
<td>Memory optimized (p. 199)</td>
</tr>
<tr>
<td>R5n</td>
<td>r5n.large</td>
<td>Memory optimized (p. 199)</td>
</tr>
<tr>
<td>T2</td>
<td>t2.nano</td>
<td>General purpose (p. 150)</td>
</tr>
<tr>
<td>T3</td>
<td>t3.nano</td>
<td>General purpose (p. 150)</td>
</tr>
<tr>
<td>T3a</td>
<td>t3a.nano</td>
<td>General purpose (p. 150)</td>
</tr>
<tr>
<td>High memory (u-*)</td>
<td>u-6tb1.56xlarge</td>
<td>Memory optimized (p. 199)</td>
</tr>
<tr>
<td>X1</td>
<td>x1.16xlarge</td>
<td>Memory optimized (p. 199)</td>
</tr>
<tr>
<td>X1e</td>
<td>x1e.xlarge</td>
<td>Memory optimized (p. 199)</td>
</tr>
<tr>
<td>z1d</td>
<td>z1d.large</td>
<td>Memory optimized (p. 199)</td>
</tr>
</tbody>
</table>
Previous generation instances

Amazon Web Services offers previous generation instance types for users who have optimized their applications around them and have yet to upgrade. We encourage you to use current generation instance types to get the best performance, but we continue to support the following previous generation instance types. For more information about which current generation instance type would be a suitable upgrade, see Previous Generation Instances.

<table>
<thead>
<tr>
<th>Type</th>
<th>Sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>c1.medium</td>
</tr>
<tr>
<td>C3</td>
<td>c3.large</td>
</tr>
<tr>
<td>G2</td>
<td>g2.2xlarge</td>
</tr>
<tr>
<td>I2</td>
<td>i2.xlarge</td>
</tr>
<tr>
<td>M1</td>
<td>m1.small</td>
</tr>
<tr>
<td>M2</td>
<td>m2.xlarge</td>
</tr>
<tr>
<td>M3</td>
<td>m3.medium</td>
</tr>
<tr>
<td>R3</td>
<td>r3.large</td>
</tr>
<tr>
<td>T1</td>
<td>t1.micro</td>
</tr>
</tbody>
</table>

Hardware specifications

For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instance Types.

To determine which instance type best meets your needs, we recommend that you launch an instance and use your own benchmark application. Because you pay by the instance second, it’s convenient and inexpensive to test multiple instance types before making a decision.

If your needs change, even after you make a decision, you can resize your instance later. For more information, see Change the instance type (p. 231).

Note

Amazon EC2 instances typically run on 64-bit virtual Intel processors as specified in the instance type product pages. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instance Types. However, confusion may result from industry naming conventions for 64-bit CPUs. Chip manufacturer Advanced Micro Devices (AMD) introduced the first commercially successful 64-bit architecture based on the Intel x86 instruction set. Consequently, the architecture is widely referred to as AMD64 regardless of the chip manufacturer. Windows and several Linux distributions follow this practice. This explains why the internal system information on an Ubuntu or Windows EC2 instance displays the CPU architecture as AMD64 even though the instances are running on Intel hardware.

Processor features

Intel processor features

Amazon EC2 instances that run on Intel processors may include the following features. Not all of the following processor features are supported by all instance types. For detailed information about which features are available for each instance type, see Amazon EC2 Instance Types.
• **Intel AES New Instructions (AES-NI)** — Intel AES-NI encryption instruction set improves upon the original Advanced Encryption Standard (AES) algorithm to provide faster data protection and greater security. All current generation EC2 instances support this processor feature.

• **Intel Advanced Vector Extensions (Intel AVX, Intel AVX2, and Intel AVX-512)** — Intel AVX and Intel AVX2 are 256-bit, and Intel AVX-512 is a 512-bit instruction set extension designed for applications that are Floating Point (FP) intensive. Intel AVX instructions improve performance for applications like image and audio/video processing, scientific simulations, financial analytics, and 3D modeling and analysis. These features are only available on instances launched with HVM AMIs.

• **Intel Turbo Boost Technology** — Intel Turbo Boost Technology processors automatically run cores faster than the base operating frequency.

• **Intel Deep Learning Boost (Intel DL Boost)** — Accelerates AI deep learning use cases. The 2nd Gen Intel Xeon Scalable processors extend Intel AVX-512 with a new Vector Neural Network Instruction (VNNI/INT8) that significantly increases deep learning inference performance over previous generation Intel Xeon Scalable processors (with FP32) for image recognition/segmentation, object detection, speech recognition, language translation, recommendation systems, reinforcement learning, and more. VNNI may not be compatible with all Linux distributions.

The following instances support VNNI: M5n, R5n, M5dn, M5zn, R5b, R5dn, D3, and D3en. C5 and C5d instances support VNNI for only 12xlarge, 24xlarge, and metal instances.

### Instances built on the Nitro System

The Nitro System is a collection of AWS-built hardware and software components that enable high performance, high availability, and high security. For more information, see [AWS Nitro System](https://aws.amazon.com/nitro).

The Nitro System provides bare metal capabilities that eliminate virtualization overhead and support workloads that require full access to host hardware. Bare metal 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

#### Nitro components

The following components are part of the Nitro System:

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

#### Instance types

The following instances are built on the Nitro System:

- **Virtualized**: C5, C5a, C5ad, C5d, C5n, D3, D3en, G4, I3en, M5, M5a, M5ad, M5d, M5dn, M5n, M5zn, M6i, p3dn.24xlarge, R5, R5a, R5ad, R5b, R5d, R5dn, R5n, T3, T3a, high memory (u*), and z1d
• **Bare metal**: c5.metal, c5d.metal, c5n.metal, i3.metal, i3en.metal, m5.metal, m5d.metal, m5dn.metal, m5zn.metal, r5.metal, r5b.metal, r5d.metal, r5dn.metal, r5n.metal, u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, u-18tb1.metal, u-24tb1.metal, and z1d.metal

**Learn more**

For more information, see the following videos:

• AWS re:Invent 2017: The Amazon EC2 Nitro System Architecture
• AWS re:Invent 2017: Amazon EC2 Bare Metal Instances
• AWS re:Invent 2019: Powering next-gen Amazon EC2: Deep dive into the Nitro system
• AWS re:Inforce 2019: Security Benefits of the Nitro Architecture

**Networking and storage features**

When you select an instance type, this determines the networking and storage features that are available. To describe an instance type, use the `describe-instance-types` command.

**Networking features**

• IPv6 is supported on all current generation instance types and the C3, R3, and I2 previous generation instance types.

• To maximize the networking and bandwidth performance of your instance type, you can do the following:
  • Launch supported instance types into a cluster placement group to optimize your instances for high performance computing (HPC) applications. Instances in a common cluster placement group can benefit from high-bandwidth, low-latency networking. For more information, see Placement groups (p. 975).
  • 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. 960).
  • Current generation instance types that are enabled for enhanced networking have the following networking performance attributes:
    • Traffic within the same Region over private IPv4 or IPv6 can support 5 Gbps for single-flow traffic and up to 25 Gbps for multi-flow traffic (depending on the instance type).
    • Traffic to and from Amazon S3 buckets within the same Region over the public IP address space or through a VPC endpoint can use all available instance aggregate bandwidth.
  • The maximum transmission unit (MTU) supported varies across instance types. All Amazon EC2 instance types support standard Ethernet V2 1500 MTU frames. All current generation instances support 9001 MTU, or jumbo frames, and some previous generation instances support them as well. For more information, see Network maximum transmission unit (MTU) for your EC2 instance (p. 987).

**Storage features**

• Some instance types support EBS volumes and instance store volumes, while other instance types support only EBS volumes. Some instance types that support instance store volumes use solid state drives (SSD) to deliver very high random I/O performance. Some instance types support NVMe instance store volumes. Some instance types support NVMe EBS volumes. For more information, see Amazon EBS and NVMe on Windows instances (p. 1343) and NVMe SSD volumes (p. 1403).
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. 1344).

### Summary of networking and storage features

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

<table>
<thead>
<tr>
<th>Instance Type</th>
<th>EBS only</th>
<th>NVMe EBS</th>
<th>Instance store</th>
<th>Placement group</th>
<th>Enhanced networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Intel 82599 VF</td>
</tr>
<tr>
<td>C5</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>C5a</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>C5ad</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>C5d</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>C5n</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>D2</td>
<td>No</td>
<td>No</td>
<td>HDD</td>
<td>Yes</td>
<td>Intel 82599 VF</td>
</tr>
<tr>
<td>D3</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>D3en</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>F1</td>
<td>No</td>
<td>No</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>G3</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>G4ad</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>G4dn</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>H1</td>
<td>No</td>
<td>No</td>
<td>HDD *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>I3</td>
<td>No</td>
<td>No</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>I3en</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>M4</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>m4.16xlarge: ENA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>All other sizes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Intel 82599 VF</td>
</tr>
<tr>
<td>M5</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>M5a</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>M5ad</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>M5d</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>M5dn</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>M5n</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
</tbody>
</table>
### Networking and storage features

<table>
<thead>
<tr>
<th>Instance type</th>
<th>EBS only</th>
<th>NVMe EBS</th>
<th>Instance store</th>
<th>Placement group</th>
<th>Enhanced networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>M5zn</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>M6i</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>P2</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>P3</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>P3dn</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>R4</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>R5</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>R5a</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>R5ad</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>R5b</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>R5d</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>R5dn</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>R5n</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>T2</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>T3</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>ENA</td>
</tr>
<tr>
<td>T3a</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>ENA</td>
</tr>
<tr>
<td>High memory (u-*)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Virtualized: Yes Bare metal: No</td>
<td>ENA</td>
</tr>
<tr>
<td>X1</td>
<td>No</td>
<td>No</td>
<td>SSD *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>X1e</td>
<td>No</td>
<td>No</td>
<td>SSD *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
<tr>
<td>z1d</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
<td>ENA</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Instance store</th>
<th>Placement group</th>
<th>Enhanced networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>C3</td>
<td>SSD</td>
<td>Yes</td>
</tr>
<tr>
<td>G2</td>
<td>SSD</td>
<td>Yes</td>
</tr>
<tr>
<td>I2</td>
<td>SSD</td>
<td>Yes</td>
</tr>
<tr>
<td>M3</td>
<td>SSD</td>
<td>No</td>
</tr>
<tr>
<td>R3</td>
<td>SSD</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Instance limits

There is a limit on the total number of instances that you can launch in a region, and there are additional limits on some instance types.

For more information about the default limits, see How many instances can I run in Amazon EC2?

For more information about viewing your current limits or requesting an increase in your current limits, see Amazon EC2 service quotas (p. 1463).

General purpose instances

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

M5 and M5a instances

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

- Small and midsize databases
- Data processing tasks that require additional memory
- Caching fleets
- Backend servers for SAP, Microsoft SharePoint, cluster computing, and other enterprise applications

For more information, see Amazon EC2 M5 Instances.

Bare metal instances, such as m5.metal, provide your applications with direct access to physical resources of the host server, such as processors and memory.

M5zn

These instances are ideal for applications that benefit from extremely high single-thread performance, high throughput, and low latency networking. They are well-suited for the following:

- Gaming
- High performance computing
- Simulation modeling

For more information, see Amazon EC2 M5 Instances.

Bare metal instances, such as m5zn.metal, provide your applications with direct access to physical resources of the host server, such as processors and memory.

M6i instances

These instances are well suited for general-purpose workloads such as the following:

- Application servers and web servers
- Microservices
- High performance computing
- App development
- Small and midsize databases
- Caching fleets
For more information, see Amazon EC2 M6i Instances.

**T2, T3, and T3a 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. 160). They are well-suited for the following:

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

For more information, see Amazon EC2 T2 Instances and Amazon EC2 T3 Instances.

**Contents**

- Hardware specifications (p. 151)
- Instance performance (p. 154)
- Network performance (p. 154)
- SSD I/O performance (p. 157)
- Instance features (p. 158)
- Release notes (p. 159)
- Burstable performance instances (p. 160)

**Hardware specifications**

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

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Memory (GiB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>m4.large</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>m4.xlarge</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>m4.2xlarge</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>m4.4xlarge</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>m4.10xlarge</td>
<td>40</td>
<td>160</td>
</tr>
<tr>
<td>m4.16xlarge</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>m5.large</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>m5.xlarge</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>m5.2xlarge</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>m5.4xlarge</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>m5.8xlarge</td>
<td>32</td>
<td>128</td>
</tr>
<tr>
<td>m5.12xlarge</td>
<td>48</td>
<td>192</td>
</tr>
<tr>
<td>m5.16xlarge</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>Instance type</td>
<td>Default vCPUs</td>
<td>Memory (GiB)</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>m5.24xlarge</td>
<td>96</td>
<td>384</td>
</tr>
<tr>
<td>m5.metal</td>
<td>96</td>
<td>384</td>
</tr>
<tr>
<td>m5a.large</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>m5a.xlarge</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>m5a.2xlarge</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>m5a.4xlarge</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>m5a.8xlarge</td>
<td>32</td>
<td>128</td>
</tr>
<tr>
<td>m5a.12xlarge</td>
<td>48</td>
<td>192</td>
</tr>
<tr>
<td>m5a.16xlarge</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>m5a.24xlarge</td>
<td>96</td>
<td>384</td>
</tr>
<tr>
<td>m5ad.large</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>m5ad.xlarge</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>m5ad.2xlarge</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>m5ad.4xlarge</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>m5ad.8xlarge</td>
<td>32</td>
<td>128</td>
</tr>
<tr>
<td>m5ad.12xlarge</td>
<td>48</td>
<td>192</td>
</tr>
<tr>
<td>m5ad.16xlarge</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>m5ad.24xlarge</td>
<td>96</td>
<td>384</td>
</tr>
<tr>
<td>m5d.large</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>m5d.xlarge</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>m5d.2xlarge</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>m5d.4xlarge</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>m5d.8xlarge</td>
<td>32</td>
<td>128</td>
</tr>
<tr>
<td>m5d.12xlarge</td>
<td>48</td>
<td>192</td>
</tr>
<tr>
<td>m5d.16xlarge</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>m5d.24xlarge</td>
<td>96</td>
<td>384</td>
</tr>
<tr>
<td>m5d.metal</td>
<td>96</td>
<td>384</td>
</tr>
<tr>
<td>m5dn.large</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>m5dn.xlarge</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>m5dn.2xlarge</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>m5dn.4xlarge</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>Instance type</td>
<td>Default vCPUs</td>
<td>Memory (GiB)</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>m5dn.8xlarge</td>
<td>32</td>
<td>128</td>
</tr>
<tr>
<td>m5dn.12xlarge</td>
<td>48</td>
<td>192</td>
</tr>
<tr>
<td>m5dn.16xlarge</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>m5dn.24xlarge</td>
<td>96</td>
<td>384</td>
</tr>
<tr>
<td>m5dn.metal</td>
<td>96</td>
<td>384</td>
</tr>
<tr>
<td>m5n.large</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>m5n.xlarge</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>m5n.2xlarge</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>m5n.4xlarge</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>m5n.8xlarge</td>
<td>32</td>
<td>128</td>
</tr>
<tr>
<td>m5n.12xlarge</td>
<td>48</td>
<td>192</td>
</tr>
<tr>
<td>m5n.16xlarge</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>m5n.24xlarge</td>
<td>96</td>
<td>384</td>
</tr>
<tr>
<td>m5n.metal</td>
<td>96</td>
<td>384</td>
</tr>
<tr>
<td>m5zn.large</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>m5zn.xlarge</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>m5zn.2xlarge</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>m5zn.3xlarge</td>
<td>12</td>
<td>48</td>
</tr>
<tr>
<td>m5zn.6xlarge</td>
<td>24</td>
<td>96</td>
</tr>
<tr>
<td>m5zn.12xlarge</td>
<td>48</td>
<td>192</td>
</tr>
<tr>
<td>m5zn.metal</td>
<td>48</td>
<td>192</td>
</tr>
<tr>
<td>m6i.large</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>m6i.xlarge</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>m6i.2xlarge</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>m6i.4xlarge</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>m6i.8xlarge</td>
<td>32</td>
<td>128</td>
</tr>
<tr>
<td>m6i.12xlarge</td>
<td>48</td>
<td>192</td>
</tr>
<tr>
<td>m6i.16xlarge</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>m6i.24xlarge</td>
<td>96</td>
<td>384</td>
</tr>
<tr>
<td>m6i.32xlarge</td>
<td>128</td>
<td>512</td>
</tr>
<tr>
<td>t2.nano</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>Instance type</td>
<td>Default vCPUs</td>
<td>Memory (GiB)</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>t2.micro</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>t2.small</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>t2.medium</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>t2.large</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>t2.xlarge</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>t2.2xlarge</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>t3.nano</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>t3.micro</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>t3.small</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>t3.medium</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>t3.large</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>t3.xlarge</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>t3.2xlarge</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>t3a.nano</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>t3a.micro</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>t3a.small</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>t3a.medium</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>t3a.large</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>t3a.xlarge</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>t3a.2xlarge</td>
<td>8</td>
<td>32</td>
</tr>
</tbody>
</table>

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 Optimize CPU options (p. 540).

**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. 1344).

**Network performance**

You can enable enhanced networking on supported instance types to provide lower latencies, lower network jitter, and higher packet-per-second (PPS) performance. Most applications do not consistently need a high level of network performance, but can benefit from access to increased bandwidth when they send or receive data. For more information, see Enhanced networking on Windows (p. 960).
The following is a summary of network performance for general purpose instances that support enhanced networking.

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Network performance</th>
<th>Enhanced networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>t2.nano</td>
<td>t2.micro</td>
<td>t2.small</td>
</tr>
<tr>
<td>t3.nano</td>
<td>t3.micro</td>
<td>t3.small</td>
</tr>
<tr>
<td>m4.large</td>
<td></td>
<td>Moderate</td>
</tr>
<tr>
<td>m4.xlarge</td>
<td>m4.2xlarge</td>
<td>m4.4xlarge</td>
</tr>
<tr>
<td>m5.4xlarge and smaller</td>
<td>m5a.8xlarge and smaller</td>
<td>m5ad.8xlarge and smaller</td>
</tr>
<tr>
<td>m4.10xlarge</td>
<td></td>
<td>10 Gbps</td>
</tr>
<tr>
<td>m5.8xlarge</td>
<td>m5a.12xlarge</td>
<td>m5ad.12xlarge</td>
</tr>
<tr>
<td>m5.12xlarge</td>
<td>m5a.16xlarge</td>
<td>m5ad.16xlarge</td>
</tr>
<tr>
<td>m6i.4xlarge and smaller</td>
<td></td>
<td>Up to 12.5 Gbps †</td>
</tr>
<tr>
<td>m6i.8xlarge</td>
<td></td>
<td>12.5 Gbps</td>
</tr>
<tr>
<td>m6i.12xlarge</td>
<td></td>
<td>18.75 Gbps</td>
</tr>
<tr>
<td>m5.16xlarge</td>
<td>m5a.24xlarge</td>
<td>m5ad.24xlarge</td>
</tr>
<tr>
<td>m5dn.4xlarge and smaller</td>
<td>m5n.4xlarge and smaller</td>
<td>m5zn.3xlarge and smaller</td>
</tr>
<tr>
<td>m4.16xlarge</td>
<td>m5.24xlarge</td>
<td>m5.metal</td>
</tr>
<tr>
<td>m6i.24xlarge</td>
<td></td>
<td>37.5 Gbps</td>
</tr>
</tbody>
</table>
### General purpose

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Network performance</th>
<th>Enhanced networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>m5dn.12xlarge</td>
<td>50 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>m5n.12xlarge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m5zn.6xlarge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m6i.32xlarge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m5dn.16xlarge</td>
<td>75 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>m5n.16xlarge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m5dn.24xlarge</td>
<td>100 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>m5n.24xlarge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m5zn.12xlarge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m5zn.metal</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

† These instances have a baseline bandwidth and can use a network I/O credit mechanism to burst beyond their baseline bandwidth on a best effort basis. For more information, see instance network bandwidth (p. 959).

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Baseline bandwidth (Gbps)</th>
<th>Burst bandwidth (Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>m5.large</td>
<td>.75</td>
<td>10</td>
</tr>
<tr>
<td>m5.xlarge</td>
<td>1.25</td>
<td>10</td>
</tr>
<tr>
<td>m5.2xlarge</td>
<td>2.5</td>
<td>10</td>
</tr>
<tr>
<td>m5.4xlarge</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>m5a.large</td>
<td>.75</td>
<td>10</td>
</tr>
<tr>
<td>m5a.xlarge</td>
<td>1.25</td>
<td>10</td>
</tr>
<tr>
<td>m5a.2xlarge</td>
<td>2.5</td>
<td>10</td>
</tr>
<tr>
<td>m5a.4xlarge</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>m5ad.large</td>
<td>.75</td>
<td>10</td>
</tr>
<tr>
<td>m5ad.xlarge</td>
<td>1.25</td>
<td>10</td>
</tr>
<tr>
<td>m5ad.2xlarge</td>
<td>2.5</td>
<td>10</td>
</tr>
<tr>
<td>m5ad.4xlarge</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>m5d.large</td>
<td>.75</td>
<td>10</td>
</tr>
<tr>
<td>m5d.xlarge</td>
<td>1.25</td>
<td>10</td>
</tr>
<tr>
<td>m5d.2xlarge</td>
<td>2.5</td>
<td>10</td>
</tr>
<tr>
<td>m5d.4xlarge</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>m5dn.1.xlarge</td>
<td>2.1</td>
<td>25</td>
</tr>
<tr>
<td>m5dn.xlarge</td>
<td>4.1</td>
<td>25</td>
</tr>
<tr>
<td>m5dn.2xlarge</td>
<td>8.125</td>
<td>25</td>
</tr>
<tr>
<td>m5dn.4xlarge</td>
<td>16.25</td>
<td>25</td>
</tr>
</tbody>
</table>
### Instance type

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Baseline bandwidth (Gbps)</th>
<th>Burst bandwidth (Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>m5n.large</td>
<td>2.1</td>
<td>25</td>
</tr>
<tr>
<td>m5n.xlarge</td>
<td>4.1</td>
<td>25</td>
</tr>
<tr>
<td>m5n.2xlarge</td>
<td>8.125</td>
<td>25</td>
</tr>
<tr>
<td>m5n.4xlarge</td>
<td>16.25</td>
<td>25</td>
</tr>
<tr>
<td>m5zn.large</td>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>m5zn.xlarge</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>m5zn.2xlarge</td>
<td>10</td>
<td>25</td>
</tr>
<tr>
<td>m5zn.3xlarge</td>
<td>15</td>
<td>25</td>
</tr>
<tr>
<td>m6i.large</td>
<td>.781</td>
<td>12.5</td>
</tr>
<tr>
<td>m6i.xlarge</td>
<td>1.562</td>
<td>12.5</td>
</tr>
<tr>
<td>m6i.2xlarge</td>
<td>3.125</td>
<td>12.5</td>
</tr>
<tr>
<td>m6i.4xlarge</td>
<td>6.25</td>
<td>12.5</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Instance Size</th>
<th>100% Random Read IOPS</th>
<th>Write IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>m5ad.large *</td>
<td>30,000</td>
<td>15,000</td>
</tr>
<tr>
<td>m5ad.xlarge *</td>
<td>59,000</td>
<td>29,000</td>
</tr>
<tr>
<td>m5ad.2xlarge *</td>
<td>117,000</td>
<td>57,000</td>
</tr>
<tr>
<td>m5ad.4xlarge *</td>
<td>234,000</td>
<td>114,000</td>
</tr>
<tr>
<td>m5ad.8xlarge</td>
<td>466,666</td>
<td>233,333</td>
</tr>
<tr>
<td>m5ad.12xlarge</td>
<td>700,000</td>
<td>340,000</td>
</tr>
<tr>
<td>m5ad.16xlarge</td>
<td>933,333</td>
<td>466,666</td>
</tr>
<tr>
<td>m5ad.24xlarge</td>
<td>1,400,000</td>
<td>680,000</td>
</tr>
<tr>
<td>m5d.large *</td>
<td>30,000</td>
<td>15,000</td>
</tr>
<tr>
<td>m5d.xlarge *</td>
<td>59,000</td>
<td>29,000</td>
</tr>
<tr>
<td>m5d.2xlarge *</td>
<td>117,000</td>
<td>57,000</td>
</tr>
<tr>
<td>m5d.4xlarge *</td>
<td>234,000</td>
<td>114,000</td>
</tr>
<tr>
<td>m5d.8xlarge</td>
<td>466,666</td>
<td>233,333</td>
</tr>
</tbody>
</table>
### General purpose Instance Size

<table>
<thead>
<tr>
<th>Instance Size</th>
<th>100% Random Read IOPS</th>
<th>Write IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>m5d.12xlarge</td>
<td>700,000</td>
<td>340,000</td>
</tr>
<tr>
<td>m5d.16xlarge</td>
<td>933,333</td>
<td>466,666</td>
</tr>
<tr>
<td>m5d.24xlarge</td>
<td>1,400,000</td>
<td>680,000</td>
</tr>
<tr>
<td>m5d.metal</td>
<td>1,400,000</td>
<td>680,000</td>
</tr>
<tr>
<td>m5dn.large*</td>
<td>30,000</td>
<td>15,000</td>
</tr>
<tr>
<td>m5dn.xlarge*</td>
<td>59,000</td>
<td>29,000</td>
</tr>
<tr>
<td>m5dn.2xlarge*</td>
<td>117,000</td>
<td>57,000</td>
</tr>
<tr>
<td>m5dn.4xlarge*</td>
<td>234,000</td>
<td>114,000</td>
</tr>
<tr>
<td>m5dn.8xlarge</td>
<td>466,666</td>
<td>233,333</td>
</tr>
<tr>
<td>m5dn.12xlarge</td>
<td>700,000</td>
<td>340,000</td>
</tr>
<tr>
<td>m5dn.16xlarge</td>
<td>933,333</td>
<td>466,666</td>
</tr>
<tr>
<td>m5dn.24xlarge</td>
<td>1,400,000</td>
<td>680,000</td>
</tr>
<tr>
<td>m5dn.metal</td>
<td>1,400,000</td>
<td>680,000</td>
</tr>
</tbody>
</table>

* 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. 1404).

### Instance features

The following is a summary of features for general purpose instances:
### Amazon Elastic Compute Cloud
**User Guide for Windows Instances**
**General purpose**

<table>
<thead>
<tr>
<th></th>
<th>EBS only</th>
<th>NVMe EBS</th>
<th>Instance store</th>
<th>Placement group</th>
</tr>
</thead>
<tbody>
<tr>
<td>M4</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>M5</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>M5a</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>M5ad</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
</tr>
<tr>
<td>M5d</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
</tr>
<tr>
<td>M5dn</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
</tr>
<tr>
<td>M5n</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>M5zn</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>M6i</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>T2</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>T3</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>T3a</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

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

For more information, see the following:

- Amazon EBS and NVMe on Windows instances (p. 1343)
- Amazon EC2 instance store (p. 1392)
- Placement groups (p. 975)

### Release notes

- M5, M5d, and T3 instances feature a 3.1 GHz Intel Xeon Platinum 8000 series processor from either the first generation (Skylake-SP) or second generation (Cascade Lake).
- M5a, M5ad, and T3a instances feature a 2.5 GHz AMD EPYC 7000 series processor.
- M5zn instances are powered by Intel Cascade Lake CPUs that deliver all-core turbo frequency of up to 4.5 GHz and up to 100 Gbps network bandwidth.
- M6i instances feature third generation Intel Xeon Scalable processors (Ice Lake) and support the Intel Advanced Vector Extensions 512 (Intel AVX-512) instruction set.
- Instances built on the Nitro System (p. 146), M4, t2.large and larger, t3.large and larger, and t3a.large and larger instance types require 64-bit HVM AMIs. They have high-memory, and require a 64-bit operating system to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. In addition, you must use an HVM AMI to take advantage of enhanced networking.
- Instances built on the Nitro System (p. 146) have the following requirements:
  - NVMe drivers (p. 1343) must be installed
  - Elastic Network Adapter (ENA) drivers (p. 961) must be installed

The current AWS Windows AMIs (p. 27) meet these requirements.
• To get the best performance from your M6i instances, ensure that they have ENA driver version 2.2.3 or later. Using an ENA driver earlier than version 2.0.0 with these instances causes network interface attachment failures. The following AMIs have a compatible ENA driver.
  • AWS Windows AMI from May 2021 or later
  • Instances built on the Nitro System support a maximum of 28 attachments, including network interfaces, EBS volumes, and NVMe instance store volumes. For more information, see Nitro System volume limits (p. 1407).
  • Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
  • To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support.
  • Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
  • There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see How many instances can I run in Amazon EC2? in the Amazon EC2 FAQ.

**Burstable performance instances**

Many general purpose workloads are on average not busy, and do not require a high level of sustained CPU performance. The following graph illustrates the CPU utilization for many common workloads that customers run in the AWS Cloud today.

![Many common workloads look like this](image)

These low-to-moderate CPU utilization workloads lead to wastage of CPU cycles and, as a result, you pay for more than you use. To overcome this, you can leverage the low-cost burstable general purpose instances, which are the T instances.

The T instance family provides a baseline CPU performance with the ability to burst above the baseline at any time for as long as required. The baseline CPU is defined to meet the needs of the majority of general purpose workloads, including large-scale micro-services, web servers, small and medium
databases, data logging, code repositories, virtual desktops, development and test environments, and business-critical applications. The T instances offer a balance of compute, memory, and network resources, and provide you with the most cost-effective way to run a broad spectrum of general purpose applications that have a low-to-moderate CPU usage. They can save you up to 15% in costs when compared to M instances, and can lead to even more cost savings with smaller, more economical instance sizes, offering as low as 2 vCPUs and 0.5 GiB of memory. The smaller T instance sizes, such as nano, micro, small, and medium, are well suited for workloads that need a small amount of memory and do not expect high CPU usage.

**EC2 burstable instance types**

The EC2 burstable instances consist of T3a and T3 instance types, and the previous generation T2 instance types.

The T4g instance types are the latest generation of burstable instances. They provide the best price for performance, and provide you with the lowest cost of all the EC2 instance types. The T4g instance types are powered by Arm-based AWS Graviton2 processors with extensive ecosystem support from operating systems vendors, independent software vendors, and popular AWS services and applications.

The following table summarizes the key differences between the burstable instance types.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Processor family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latest generation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T4g</td>
<td>Lowest cost EC2 instance type with up to 40% higher price/performance and 20% lower costs vs T3</td>
<td>AWS Graviton2 processors with Arm Neoverse N1 cores</td>
</tr>
<tr>
<td>T3a</td>
<td>Lowest cost x86-based instances with 10% lower costs vs T3 instances</td>
<td>AMD 1st gen EPYC processors</td>
</tr>
<tr>
<td>T3</td>
<td>Best peak price/performance for x86 workloads with up to 30% lower price/performance vs previous generation T2 instances</td>
<td>Intel Xeon Scalable (Skylake, Cascade Lake processors)</td>
</tr>
</tbody>
</table>

| Previous generation |                                                                             |                                    |
|---------------------|------------------------------------------------------------------------------|                                    |
| T2                  | Previous generation burstable instances                                      | Intel Xeon processors              |

For more information about instance pricing and additional specifications, see Amazon EC2 Pricing and Amazon EC2 Instance Types.

If your account is less than 12 months old, you can use a t2.micro instance for free (or a t3.micro instance in Regions where t2.micro is unavailable) within certain usage limits. For more information, see AWS Free Tier.

**Supported purchasing options for T instances**

- On-Demand Instances
- Reserved Instances
- Dedicated Instances (T3 only)
- Spot Instances
T instances are not supported on a Dedicated Host. For more information, see Instance purchasing options (p. 239).

Contents

• Best practices (p. 162)
• Key concepts and definitions for burstable performance instances (p. 162)
• Unlimited mode for burstable performance instances (p. 168)
• Standard mode for burstable performance instances (p. 175)
• Work with burstable performance instances (p. 184)
• Monitor your CPU credits (p. 189)

Best practices

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

• 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 t3.micro or larger instance size for many use cases. As the memory and CPU requirements of your workload grow over time, you have the flexibility with the T instances to scale to larger instance sizes of the same instance type, or to select another instance type.
• Enable AWS Compute Optimizer for your account and review the Compute Optimizer recommendations for your workload. Compute Optimizer can help assess whether instances should be upsized to improve performance or downsized for cost savings.
• For additional requirements, see Release notes (p. 159).

Key concepts and definitions for burstable performance instances

Traditional Amazon EC2 instance types provide fixed CPU resources, while burstable performance instances provide a baseline level of CPU utilization with the ability to burst CPU utilization above the baseline level. This ensures that you pay only for baseline CPU plus any additional burst CPU usage resulting in lower compute costs. The baseline utilization and ability to burst are governed by CPU credits. Burstable performance instances are the only instance types that use credits for CPU usage.

Each burstable performance instance continuously earns credit when it stays below the CPU baseline, and continuously spends credits when it bursts above the baseline. The amount of credits earned or spent depends on the CPU utilization of the instance:

• If the CPU utilization is below baseline, then credits earned are greater than credits spent.
• If the CPU utilization is equal to baseline, then credits earned are equal to credits spent.
• If the CPU utilization is higher than baseline, then credits spent are higher than credits earned.

When the credits earned are greater than credits spent, then the difference is called accrued credits, which can be used later to burst above baseline CPU utilization. Similarly, when the credits spent are more than credits earned, then the instance behavior depends on the credit configuration mode—Standard mode or Unlimited mode.

In Standard mode, when credits spent are more than credits earned, the instance uses the accrued credits to burst above baseline CPU utilization. If there are no accrued credits remaining, then the instance gradually comes down to baseline CPU utilization and cannot burst above baseline until it accrues more credits.

In Unlimited mode, if the instance bursts above baseline CPU utilization, then the instance first uses the accrued credits to burst. If there are no accrued credits remaining, then the instance spends surplus
credits to burst. When its CPU utilization falls below the baseline, it uses the CPU credits that it earns to pay down the surplus credits that it spent earlier. The ability to earn CPU credits to pay down surplus credits enables Amazon EC2 to average the CPU utilization of an instance over a 24-hour period. If the average CPU usage over a 24-hour period exceeds the baseline, the instance is billed for the additional usage at a flat additional rate per vCPU-hour.

Key concepts and definitions

The following key concepts and definitions are applicable to burstable performance instances.

CPU utilization

CPU utilization is the percentage of allocated EC2 compute units that are currently in use on the instance. This metric measures the percentage of allocated CPU cycles that are being utilized on an instance. The CPU Utilization CloudWatch metric shows CPU usage per instance and not CPU usage per core. The baseline CPU specification of an instance is also based on the CPU usage per instance. To measure CPU utilization using the AWS Management Console or the AWS CLI, see Get statistics for a specific instance (p. 854).

CPU credit

A unit of vCPU-time.

Examples:

1 CPU credit = 1 vCPU * 100% utilization * 1 minute.
1 CPU credit = 1 vCPU * 50% utilization * 2 minutes
1 CPU credit = 2 vCPU * 25% utilization * 2 minutes

Baseline utilization

The baseline utilization is the level at which the CPU can be utilized for a net credit balance of zero, when the number CPU credits being earned matches the number of CPU credits being used. Baseline utilization is also known as the baseline. Baseline utilization is expressed as a percentage of vCPU utilization, which is calculated as follows: Baseline utilization % = (number of credits earned/number of vCPUs)/60 minutes

Earned credits

Credits earned continuously by an instance when it is running.

Number of credits earned per hour = % baseline utilization * number of vCPUs * 60 minutes

Example:

A t3.nano with 2 vCPUs and a baseline utilization of 5% earns 6 credits per hour, calculated as follows:

2 vCPUs * 5% baseline * 60 minutes = 6 credits per hour

Spent or used credits

Credits used continuously by an instance when it is running.
CPU credits spent per minute = Number of vCPUs * CPU utilization * 1 minute

**Accrued credits**

Unspent CPU credits when an instance uses fewer credits than is required for baseline utilization. In other words, accrued credits = (Earned credits – Used credits) below baseline.

Example:

If a t3.nano is running at 2% CPU utilization, which is below its baseline of 5% for an hour, the accrued credits is calculated as follows:

Accrued CPU credits = (Earned credits per hour – Used credits per hour) = 6 – 2 vCPUs * 2% CPU utilization * 60 minutes = 6 – 2.4 = 3.6 accrued credits per hour

**Credit accrual limit**

Depends on the instance size but in general is equal to the number of maximum credits earned in 24 hours.

Example:

For t3.nano, the credit accrual limit = 24 * 6 = 144 credits

**Launch credits**

Only applicable for T2 instances configured for Standard mode. Launch credits are a limited number of CPU credits that are allocated to a new T2 instance so that, when launched in Standard mode, it can burst above the baseline.

**Surplus credits**

Credits that are spent by an instance after it depletes its accrued credit balance. The surplus credits are designed for burstable instances to sustain high performance for an extended period of time, and are only used in Unlimited mode. The surplus credits balance is used to determine how many credits were used by the instance for bursting in Unlimited mode.

**Standard mode**

Credit configuration mode, which allows an instance to burst above the baseline by spending credits it has accrued in its credit balance.

**Unlimited mode**

Credit configuration mode, which allows an instance to burst above the baseline by sustaining high CPU utilization for any period of time whenever required. The hourly instance price automatically covers all CPU usage spikes if the average CPU utilization of the instance is at or below the baseline over a rolling 24-hour period or the instance lifetime, whichever is shorter. If the instance runs at higher CPU utilization for a prolonged period, it can do so for a flat additional rate per vCPU-hour.

The following table summarizes the key credit differences between the burstable instance types.

<table>
<thead>
<tr>
<th>Type</th>
<th>Type of CPU credits supported</th>
<th>Credit configuration modes</th>
<th>Accrued CPU credits lifespan between instance starts and stops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latest generation</td>
<td>Earned credits, Accrued credits, Spent credits, Surplus credits (Unlimited mode only)</td>
<td>Standard, Unlimited (default)</td>
<td>7 days (credits persist for 7 days after an instance stops)</td>
</tr>
<tr>
<td>T4g</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Amazon Elastic Compute Cloud

**User Guide for Windows Instances**

**General purpose**

<table>
<thead>
<tr>
<th>Type</th>
<th>Type of CPU credits supported</th>
<th>Credit configuration modes</th>
<th>Accrued CPU credits lifespan between instance starts and stops</th>
</tr>
</thead>
<tbody>
<tr>
<td>T3a</td>
<td>Earned credits, Accrued credits, Spent credits, Surplus credits (Unlimited mode only)</td>
<td>Standard, Unlimited (default)</td>
<td>7 days (credits persist for 7 days after an instance stops)</td>
</tr>
<tr>
<td>T3</td>
<td>Earned credits, Accrued credits, Spent credits, Surplus credits (Unlimited mode only)</td>
<td>Standard, Unlimited (default)</td>
<td>7 days (credits persist for 7 days after an instance stops)</td>
</tr>
</tbody>
</table>

**Previous generation**

| T2   | Earned credits, Accrued credits, Spent credits, Launch credits (Standard mode only), Surplus credits (Unlimited mode only) | Standard (default), Unlimited | 0 days (credits are lost when an instance stops) |

**Earn CPU credits**

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

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

The following table lists the burstable performance instance types, the rate at which CPU credits are earned per hour, the maximum number of earned CPU credits that an instance can accrue, the number of vCPUs per instance, and the baseline utilization as a percentage of a full core (using a single vCPU).

<table>
<thead>
<tr>
<th>Instance type</th>
<th>CPU credits earned per hour</th>
<th>Maximum earned credits that can be accrued*</th>
<th>vCPUs***</th>
<th>Baseline utilization per vCPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t2.nano</td>
<td>3</td>
<td>72</td>
<td>1</td>
<td>5%</td>
</tr>
<tr>
<td>t2.micro</td>
<td>6</td>
<td>144</td>
<td>1</td>
<td>10%</td>
</tr>
<tr>
<td>t2.small</td>
<td>12</td>
<td>288</td>
<td>1</td>
<td>20%</td>
</tr>
<tr>
<td>t2.medium</td>
<td>24</td>
<td>576</td>
<td>2</td>
<td>20%**</td>
</tr>
<tr>
<td>t2.large</td>
<td>36</td>
<td>864</td>
<td>2</td>
<td>30%**</td>
</tr>
<tr>
<td>t2.xlarge</td>
<td>54</td>
<td>1296</td>
<td>4</td>
<td>22.5%**</td>
</tr>
</tbody>
</table>
## Instance type

<table>
<thead>
<tr>
<th>Instance type</th>
<th>CPU credits earned per hour</th>
<th>Maximum earned credits that can be accrued*</th>
<th>vCPUs***</th>
<th>Baseline utilization per vCPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>t2.2xlarge</td>
<td>81.6</td>
<td>1958.4</td>
<td>8</td>
<td>17%**</td>
</tr>
<tr>
<td>T3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t3.nano</td>
<td>6</td>
<td>144</td>
<td>2</td>
<td>5%**</td>
</tr>
<tr>
<td>t3.micro</td>
<td>12</td>
<td>288</td>
<td>2</td>
<td>10%**</td>
</tr>
<tr>
<td>t3.small</td>
<td>24</td>
<td>576</td>
<td>2</td>
<td>20%**</td>
</tr>
<tr>
<td>t3.medium</td>
<td>24</td>
<td>576</td>
<td>2</td>
<td>20%**</td>
</tr>
<tr>
<td>t3.large</td>
<td>36</td>
<td>864</td>
<td>2</td>
<td>30%**</td>
</tr>
<tr>
<td>t3.xlarge</td>
<td>96</td>
<td>2304</td>
<td>4</td>
<td>40%**</td>
</tr>
<tr>
<td>t3.2xlarge</td>
<td>192</td>
<td>4608</td>
<td>8</td>
<td>40%**</td>
</tr>
<tr>
<td>T3a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t3a.nano</td>
<td>6</td>
<td>144</td>
<td>2</td>
<td>5%**</td>
</tr>
<tr>
<td>t3a.micro</td>
<td>12</td>
<td>288</td>
<td>2</td>
<td>10%**</td>
</tr>
<tr>
<td>t3a.small</td>
<td>24</td>
<td>576</td>
<td>2</td>
<td>20%**</td>
</tr>
<tr>
<td>t3a.medium</td>
<td>24</td>
<td>576</td>
<td>2</td>
<td>20%**</td>
</tr>
<tr>
<td>t3a.large</td>
<td>36</td>
<td>864</td>
<td>2</td>
<td>30%**</td>
</tr>
<tr>
<td>t3a.xlarge</td>
<td>96</td>
<td>2304</td>
<td>4</td>
<td>40%**</td>
</tr>
<tr>
<td>t3a.2xlarge</td>
<td>192</td>
<td>4608</td>
<td>8</td>
<td>40%**</td>
</tr>
</tbody>
</table>

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

** The percentage baseline utilization in the table is per vCPU. In CloudWatch, CPU utilization is shown per vCPU. For example, the CPU utilization for a `t3.large` instance operating at the baseline level is shown as 30% in CloudWatch CPU metrics. For information about how to calculate the baseline utilization, see Baseline utilization (p. 167).

*** Each vCPU is a thread of either an Intel Xeon core or an AMD EPYC core, except for T2 instances.

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

T2 Standard instances also earn launch credits. Launch credits do not count towards the CPU credit balance limit. If a T2 instance has not spent its launch credits, and remains idle over a 24-hour period while accruing earned credits, its CPU credit balance appears as over the limit. For more information, see Launch credits (p. 176).

T3a and 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 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 T3a and 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 more information, see CPUCreditBalance in the CloudWatch metrics table (p. 190).

**Baseline utilization**

The baseline utilization is the level at which the CPU can be utilized for a net credit balance of zero, when the number of CPU credits being earned matches the number of CPU credits being used. Baseline utilization is also known as the baseline.

Baseline utilization is expressed as a percentage of vCPU utilization, which is calculated as follows:

$$\frac{\text{number of credits earned/number of vCPUs}}{60 \text{ minutes}} = \% \text{ baseline utilization}$$

For example, a t3.nano instance, with 2 vCPUs, earns 6 credits per hour, resulting in a baseline utilization of 5% , which is calculated as follows:

$$\frac{6 \text{ credits earned/2 vCPUs}}{60 \text{ minutes}} = 5\% \text{ baseline utilization}$$
A t3.xlarge instance, with 4 vCPUs, earns 96 credits per hour, resulting in a baseline utilization of 40% \( \frac{(96/4)/60} \).

The following graph provides an example of a t3.large with an average CPU utilization below the baseline.

Unlimited mode for burstable performance instances

A burstable performance instance configured as unlimited can sustain high CPU utilization for any period of time whenever required. The hourly instance price automatically covers all CPU usage spikes if the average CPU utilization of the instance is at or below the baseline over a rolling 24-hour period or the instance lifetime, whichever is shorter.

For the vast majority of general-purpose workloads, instances configured as unlimited provide ample performance without any additional charges. If the instance runs at higher CPU utilization for a prolonged period, it can do so for a flat additional rate per vCPU-hour. For information about pricing, see Amazon EC2 pricing and T2/T3/T4 Unlimited Mode Pricing.

If you use a t2.micro or t3.micro instance under the AWS Free Tier offer and use it in unlimited mode, charges might apply if your average utilization over a rolling 24-hour period exceeds the baseline utilization (p. 167) of the instance.

T3a and T3 instances launch as unlimited by default. If the average CPU usage over a 24-hour period exceeds the baseline, you incur charges for surplus credits. If you launch Spot Instances as unlimited and plan to use them immediately and for a short duration, with no idle time for accruing CPU credits, you incur charges for surplus credits. We recommend that you launch your Spot Instances in standard (p. 175) mode to avoid paying higher costs. For more information, see Surplus credits can incur charges (p. 171) and Burstable performance instances (p. 328).

Contents
- Unlimited mode concepts (p. 169)
- How Unlimited burstable performance instances work (p. 169)
- When to use unlimited mode versus fixed CPU (p. 169)
- Surplus credits can incur charges (p. 171)
- No launch credits for T2 Unlimited instances (p. 171)
Unlimited mode concepts

The unlimited mode is a credit configuration option for burstable performance instances. It can be enabled or disabled at any time for a running or stopped instance. You can set unlimited as the default credit option at the account level per AWS Region, per burstable performance instance family, so that all new burstable performance instances in the account launch using the default credit option.

How Unlimited burstable performance instances work

If a burstable performance instance configured as unlimited depletes its CPU credit balance, it can spend surplus credits to burst beyond the baseline (p. 167). When its CPU utilization falls below the baseline, it uses the CPU credits that it earns to pay down the surplus credits that it spent earlier. The ability to earn CPU credits to pay down surplus credits enables Amazon EC2 to average the CPU utilization of an instance over a 24-hour period. If the average CPU usage over a 24-hour period exceeds the baseline, the instance is billed for the additional usage at a flat additional rate per vCPU-hour.

The following graph shows the CPU usage of a t3.large. The baseline CPU utilization for a t3.large is 30%. If the instance runs at 30% CPU utilization or less on average over a 24-hour period, there is no additional charge because the cost is already covered by the instance hourly price. However, if the instance runs at 40% CPU utilization on average over a 24-hour period, as shown in the graph, the instance is billed for the additional 10% CPU usage at a flat additional rate per vCPU-hour.

For more information about the baseline utilization per vCPU for each instance type and how many credits each instance type earns, see the credit table (p. 165).

When to use unlimited mode versus fixed CPU

When determining whether you should use a burstable performance instance in unlimited mode, such as T3, or a fixed performance instance, such as M5, you need to determine the breakeven CPU usage. The breakeven CPU usage for a burstable performance instance is the point at which a burstable performance instance costs the same as a fixed performance instance. The breakeven CPU usage helps you determine the following:
• If the average CPU usage over a 24-hour period is at or below the breakeven CPU usage, use a burstable performance instance in **unlimited** mode so that you can benefit from the lower price of a burstable performance instance while getting the same performance as a fixed performance instance.

• If the average CPU usage over a 24-hour period is above the breakeven CPU usage, the burstable performance instance will cost more than the equivalently-sized fixed performance instance. If a T3 instance continuously bursts at 100% CPU, you end up paying approximately 1.5 times the price of an equivalently-sized M5 instance.

The following graph shows the breakeven CPU usage point where a t3.large costs the same as an m5.large. The breakeven CPU usage point for a t3.large is 42.5%. If the average CPU usage is at 42.5%, the cost of running the t3.large is the same as an m5.large, and is more expensive if the average CPU usage is above 42.5%. If the workload needs less than 42.5% average CPU usage, you can benefit from the lower price of the t3.large while getting the same performance as an m5.large.

The following table shows how to calculate the breakeven CPU usage threshold so that you can determine when it’s less expensive to use a burstable performance instance in **unlimited** mode or a fixed performance instance. The columns in the table are labeled A through K.

<table>
<thead>
<tr>
<th>Instance type</th>
<th>vCPUs</th>
<th>T3 price*/hour</th>
<th>M5 price*/hour</th>
<th>Price difference</th>
<th>Charge per vCPU for surplus credits</th>
<th>Charge per vCPU minute</th>
<th>Additional burst minutes available per vCPU</th>
<th>Additional CPU % available</th>
<th>Breakeven CPU %</th>
</tr>
</thead>
<tbody>
<tr>
<td>t3.large</td>
<td>2</td>
<td>$0.0835</td>
<td>$0.096</td>
<td>$0.0125</td>
<td>30%</td>
<td>$0.05</td>
<td>15</td>
<td>12.5%</td>
<td>42.5%</td>
</tr>
</tbody>
</table>

* Price is based on us-east-1 and Linux OS.

The table provides the following information:

• Column A shows the instance type, t3.large.
• Column B shows the number of vCPUs for the t3.large.
• Column C shows the price of a t3.large per hour.
• Column D shows the price of an m5.large per hour.
• Column E shows the price difference between the t3.large and the m5.large.
- Column F shows the baseline utilization per vCPU of the \texttt{t3.large}, which is 30%. At the baseline, the hourly cost of the instance covers the cost of the CPU usage.
- Column G shows the flat additional rate per vCPU-hour that an instance is charged if it bursts at 100% CPU after it has depleted its earned credits.
- Column H shows the flat additional rate per vCPU-minute that an instance is charged if it bursts at 100% CPU after it has depleted its earned credits.
- Column I shows the number of additional minutes that the \texttt{t3.large} can burst per hour at 100% CPU while paying the same price per hour as an \texttt{m5.large}.
- Column J shows the additional CPU usage (in %) over baseline that the instance can burst while paying the same price per hour as an \texttt{m5.large}.
- Column K shows the breakeven CPU usage (in %) that the \texttt{t3.large} can burst without paying more than the \texttt{m5.large}. Anything above this, and the \texttt{t3.large} costs more than the \texttt{m5.large}.

The following table shows the breakeven CPU usage (in %) for T3 instance types compared to the similarly-sized M5 instance types.

<table>
<thead>
<tr>
<th>T3 instance type</th>
<th>Breakeven CPU usage (in %) for T3 compared to M5</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{t3.large}</td>
<td>42.5%</td>
</tr>
<tr>
<td>\texttt{t3.xlarge}</td>
<td>52.5%</td>
</tr>
<tr>
<td>\texttt{t3.2xlarge}</td>
<td>52.5%</td>
</tr>
</tbody>
</table>

**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. 165) in a 24-hour period (for example, a \texttt{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. For information about the rate, see \textit{T2/T3/T4g Unlimited Mode Pricing}.

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

- The spent surplus credits exceed the maximum number of credits (p. 165) 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 \texttt{CPUSurplusCreditBalance}. Surplus credits that are charged are tracked by the CloudWatch metric \texttt{CPUSurplusCreditsCharged}. For more information, see \textit{Additional CloudWatch metrics for burstable performance instances (p. 190)}.

**No launch credits for T2 Unlimited instances**

T2 Standard instances receive launch credits (p. 176), 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.

T3a and T3 instances never receive launch credits because they support Unlimited mode. The Unlimited mode credit configuration enables T4g, T3a and T3 instances to use as much CPU as needed to burst beyond baseline and for as long as needed.

Enable unlimited mode

You can switch from unlimited to standard, and from standard to unlimited, at any time on a running or stopped instance. For more information, see Launch a burstable performance instance as Unlimited or Standard (p. 185) and Modify the credit specification of a burstable performance instance (p. 188).

You can set unlimited as the default credit option at the account level per AWS Region, per burstable performance instance family, so that all new burstable performance instances in the account launch using the default credit option. For more information, see Set the default credit specification for the account (p. 189).

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 View the credit specification of a burstable performance instance (p. 187) and View the default credit specification (p. 189).

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.

Monitor credit usage

To see if your instance is spending more credits than the baseline provides, you can use CloudWatch metrics to track usage, and you can set up hourly alarms to be notified of credit usage. For more information, see Monitor your CPU credits (p. 189).

Unlimited mode examples

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

Examples

- Example 1: Explain credit use with T3 Unlimited (p. 172)
- Example 2: Explain credit use with T2 Unlimited (p. 174)

Example 1: Explain credit use with T3 Unlimited

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

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

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

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

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

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

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

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

P7 – For the last 24 hours in this example, the instance is idle and CPU utilization is 0%. During this time, the instance earns 144 credits, which it uses to pay down the CPUSurplusCreditBalance.
Example 2: Explain credit use with T2 Unlimited

In this example, you see the CPU utilization of a \texttt{t2.nano} instance launched as \texttt{unlimited}, and how it spends \textit{earned} and \textit{surplus} credits to sustain CPU utilization.

A \texttt{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 \texttt{CPUCreditBalance}), it can spend \textit{surplus} CPU credits—that it has \textit{not yet earned}—to burst for as long as it needs. Because a \texttt{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 \texttt{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, \texttt{CPUCreditUsage} is at 0, and the \texttt{CPUCreditBalance} value remains at its maximum of 72.

2 – At 23:40, as CPU utilization increases, the instance spends CPU credits and the \texttt{CPUCreditBalance} value decreases.

3 – At around 00:47, the instance depletes its entire \texttt{CPUCreditBalance}, and starts to spend surplus credits to sustain high CPU utilization.

4 – Surplus credits are spent until 01:55, when the \texttt{CPUSurplusCreditBalance} value reaches 72 CPU credits. This is equal to the maximum a \texttt{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 \texttt{CPUSurplusCreditBalance}. After the \texttt{CPUSurplusCreditBalance} value reduces to 0, the instance starts to accrue earned credits in its \texttt{CPUCreditBalance} at 0.25 credits every 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 vCPU-hours x $0.096/vCPU-hour = $0.04032, rounded to $0.04.

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

<table>
<thead>
<tr>
<th>Amazon Elastic Compute Cloud running Windows</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.0081 per On Demand Windows t2.nano instance Hour</td>
<td>720.000 Hrs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Amazon Elastic Compute Cloud T2 CPU Credits</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.006 per vCPU-Hour of T2 CPU credits</td>
<td>0.420 vCPU-Hours</td>
</tr>
</tbody>
</table>

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

Standard mode for burstable performance instances

A burstable performance instance configured as standard is suited to workloads with an average CPU utilization that is consistently below the baseline CPU utilization of the instance. To burst above the baseline, the instance spends credits that it has accrued in its CPU credit balance. If the instance is running low on accrued credits, CPU utilization is gradually lowered to the baseline level, so that the instance does not experience a sharp performance drop-off when its accrued CPU credit balance is depleted. For more information, see Key concepts and definitions for burstable performance instances (p. 162).

Contents

- Standard mode concepts (p. 175)
  - How standard burstable performance instances work (p. 176)
  - Launch credits (p. 176)
  - Launch credit limits (p. 176)
  - Differences between launch credits and earned credits (p. 177)
- Standard mode examples (p. 177)
  - Example 1: Explain credit use with T3 Standard (p. 177)
  - Example 2: Explain credit use with T2 Standard (p. 179)
    - Period 1: 1 – 24 hours (p. 179)
    - Period 2: 25 – 36 hours (p. 180)
    - Period 3: 37 – 61 hours (p. 180)
    - Period 4: 62 – 72 hours (p. 181)
    - Period 5: 73 – 75 hours (p. 182)
    - Period 6: 76 – 90 hours (p. 183)
    - Period 7: 91 – 96 hours (p. 184)

Standard mode concepts

The standard mode is a configuration option for burstable performance instances. It can be enabled or disabled at any time for a running or stopped instance. You can set standard as the default credit option at the account level per AWS Region, per burstable performance instance family, so that all new burstable performance instances in the account launch using the default credit option.
How standard burstable performance instances work

When a burstable performance instance configured as standard is in a running state, it continuously earns (at a millisecond-level resolution) a set rate of earned credits per hour. For T2 Standard, when the instance is stopped, it loses all its accrued credits, and its credit balance is reset to zero. When it is restarted, it receives a new set of launch credits, and begins to accrue earned credits. For T3a and T3 Standard instances, the CPU credit balance persists for seven days after the instance stops and the credits are lost thereafter. If you start the instance within seven days, no credits are lost.

T2 Standard instances receive two types of CPU credits: earned credits and launch credits. When a T2 Standard instance is in a running state, it continuously earns (at a millisecond-level resolution) a set rate of earned credits per hour. At start, it has not yet earned credits for a good startup experience; therefore, to provide a good startup experience, it receives launch credits at start, which it spends first while it accrues earned credits.

T3a and T3 instances do not receive launch credits because they support Unlimited mode. The Unlimited mode credit configuration enables T4g, T3a and T3 instances to use as much CPU as needed to burst beyond baseline and for as long as needed.

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. 165).

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

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Launch credits</th>
<th>vCPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1.micro</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>t2.nano</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>t2.micro</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>t2.small</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>t2.medium</td>
<td>60</td>
<td>2</td>
</tr>
<tr>
<td>t2.large</td>
<td>60</td>
<td>2</td>
</tr>
<tr>
<td>t2.xlarge</td>
<td>120</td>
<td>4</td>
</tr>
<tr>
<td>t2.2xlarge</td>
<td>240</td>
<td>8</td>
</tr>
</tbody>
</table>

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. 168) 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.

<table>
<thead>
<tr>
<th>Credit earn rate</th>
<th>Launch credits</th>
<th>Earned credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td>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. 165).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Credit earn limit</th>
<th>Launch credits</th>
<th>Earned credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>The limit for receiving launch credits is 100 launches or starts of all T2 Standard instances combined per account, per Region, per rolling 24-hour period. New accounts may have a lower limit, which increases over time based on your usage.</td>
<td>A T2 instance cannot accrue more credits than the CPU credit balance limit. If the CPU credit balance has reached its limit, any credits that are earned after the limit is reached are discarded. Launch credits do not count towards the limit. For more information about the CPU credit balance limit for each T2 instance size, see the credit table (p. 165).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Credit use</th>
<th>Launch credits</th>
<th>Earned credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch credits are spent first, before earned credits.</td>
<td>Earned credits are spent only after all launch credits are spent.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Credit expiration</th>
<th>Launch credits</th>
<th>Earned credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

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. 190).

**Standard mode examples**

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

**Examples**

- Example 1: Explain credit use with T3 Standard (p. 177)
- Example 2: Explain credit use with T2 Standard (p. 179)

**Example 1: Explain credit use with T3 Standard**

In this example, you see how a t3.nano instance launched as standard earns, accrues, and spends earned credits. You see how the credit balance reflects the accrued earned credits.
A running `t3.nano` instance earns 144 credits every 24 hours. Its credit balance limit is 144 earned credits. After the limit is reached, new credits that are earned are discarded. For more information about the number of credits that can be earned and accrued, see the credit table (p. 165).

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

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

The following workflow references the numbered points on the graph:

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

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

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

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

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

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

**P7** – For the last 24 hours in this example, the instance is idle and CPU utilization is 0%. During this time, the instance earns 144 credits, which it accrues in its CPUCreditBalance.
Example 2: Explain 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. 165). For more information about limits, see Launch credit limits (p. 176).

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.

<table>
<thead>
<tr>
<th>Credit Spend Rate</th>
<th>0 credits per 24 hours (0% CPU utilization)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Earn Rate</td>
<td>72 credits per 24 hours</td>
</tr>
<tr>
<td>Credit Discard Rate</td>
<td>0 credits per 24 hours</td>
</tr>
<tr>
<td>Credit Balance</td>
<td>102 credits (30 launch credits + 72 earned credits)</td>
</tr>
</tbody>
</table>
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.
Amazon Elastic Compute Cloud
User Guide for Windows Instances
General purpose

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

| 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. 165).

**Work with burstable performance instances**

The steps for launching, monitoring, and modifying these instances are similar. The key difference is the default credit specification when they launch. If you do not change the default credit specification, the default is that:

- T3a and T3 instances launch as unlimited
• T2 instances launch as standard

Contents

• Launch a burstable performance instance as Unlimited or Standard (p. 185)
• Use an Auto Scaling group to launch a burstable performance instance as Unlimited (p. 185)
• View the credit specification of a burstable performance instance (p. 187)
• Modify the credit specification of a burstable performance instance (p. 188)
• Set the default credit specification for the account (p. 189)
• View the default credit specification (p. 189)

Launch a burstable performance instance as Unlimited or Standard

You can launch your instances as unlimited or standard using the Amazon EC2 console, an AWS SDK, a command line tool, or with an Auto Scaling group. For more information, see Use an Auto Scaling group to launch a burstable performance instance as Unlimited (p. 185).

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

1. Follow the Launch an instance using the Launch Instance Wizard (p. 392) procedure.
2. On the Choose an Instance Type page, select an instance type, and choose Next: Configure Instance Details.
3. Choose a credit specification.
   a. To launch a T3a and T3 instance as standard, clear Unlimited.
   b. To launch a T2 instance as unlimited, select Unlimited.
4. Continue as prompted by the wizard. When you've finished reviewing your options on the Review Instance Launch page, choose Launch. For more information, see Launch an instance using the Launch Instance Wizard (p. 392).

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 T3a and 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-abcd12345 --count 1 --instance-type t3.micro --key-name MyKeyPair --credit-specification "CpuCredits=unlimited"
```

Use 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.
Create a launch template

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

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

1. Follow the Creating a Launch Template for an Auto Scaling Group procedure.
2. In Launch template contents, for Instance type, choose an instance size.
3. To launch instances as unlimited in an Auto Scaling group, under Advanced details, for Credit specification, choose Unlimited.
4. When you’ve finished defining the launch template parameters, choose Create launch template. For more information, see Creating a Launch Template for an Auto Scaling Group in the Amazon EC2 Auto Scaling User Guide.

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

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

- For T3a and 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}
```

Associate 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.

**View the credit specification of a burstable performance instance**

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

New console

**To view the credit specification of a burstable instance**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Instances**.
3. Select the instance.
4. Choose **Details** and view the **Credit specification** field. The value is either unlimited or standard.

Old console

**To view the credit specification of a burstable instance**

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

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

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

**Example**

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

The following is example output:

```
{
"InstanceCreditSpecifications": [
{
  "InstanceId": "i-1234567890abcdef0",
  "CpuCredits": "unlimited"
}
]}
```
Modify the credit specification of a burstable performance instance

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

New console

To modify the credit specification of a burstable performance instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose Instances.
3. Select the instance. To modify the credit specification for several instances at one time, select all applicable instances.
4. Choose Actions, Instance settings, Change credit specification. This option is enabled only if you selected a burstable performance instance.
5. To change the credit specification to unlimited, select the check box next to the instance ID. To change the credit specification to standard, clear the check box next to the instance ID.

Old console

To modify the credit specification of a burstable performance instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose Instances.
3. Select the instance. To modify the credit specification for several instances at one time, select all applicable instances.
4. Choose Actions, Instance Settings, Change T2/T3 Unlimited. This option is enabled only if you selected a burstable performance instance.
5. The current credit specification appears in parentheses after the instance ID. To change the credit specification to unlimited, choose Enable. To change the credit specification to standard, choose Disable.

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

Use the modify-instance-credit-specification command. Specify the instance and its credit specification using the --instance-credit-specification parameter. Valid credit specifications are unlimited and standard.

Example

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

The following is example output:

```
{
   "SuccessfulInstanceCreditSpecifications": [
      {
         "InstanceId": "i-1234567890abcdef0"
      }
   ],
   "UnsuccessfulInstanceCreditSpecifications": []
}
```
Set the default credit specification for the account

You can set the default credit specification for each burstable performance instance family at the account level per AWS Region.

If you use the Launch Instance Wizard in the EC2 console to launch instances, the value you select for the credit specification overrides the account-level default credit specification. If you use the AWS CLI to launch instances, all new burstable performance instances in the account launch using the default credit specification. The credit specification for existing running or stopped instances is not affected.

Consideration

The default credit specification for an instance family can be modified only once in a rolling 5-minute period, and up to four times in a rolling 24-hour period.

To set the default credit specification at the account level (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the left navigation pane, choose EC2 Dashboard.
3. From Account attributes, choose Default credit specification.
4. Choose Manage.
5. For each instance family, choose Unlimited or Standard, and then choose Update.

To set the default credit specification at the account level (AWS CLI)

Use the modify-default-credit-specification command. Specify the AWS Region, instance family, and the default credit specification using the --cpu-credits parameter. Valid default credit specifications are unlimited and standard.

```
aws ec2 modify-default-credit-specification --region us-east-1 --instance-family t2 --cpu-credits unlimited
```

View the default credit specification

You can view the default credit specification of a burstable performance instance family at the account level per AWS Region.

To view the default credit specification at the account level (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the left navigation pane, choose EC2 Dashboard.
3. From Account attributes, choose Default credit specification.

To view the default credit specification at the account level (AWS CLI)

Use the get-default-credit-specification command. Specify the AWS Region and instance family.

```
aws ec2 get-default-credit-specification --region us-east-1 --instance-family t2
```

Monitor your CPU credits

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

Burstable performance instances have these additional CloudWatch metrics, which are updated every five minutes:

- CPUCreditUsage – The number of CPU credits spent during the measurement period.
- CPUCreditBalance – The number of CPU credits that an instance has accrued. This balance is depleted when the CPU bursts and CPU credits are spent more quickly than they are earned.
- CPUSurplusCreditBalance – The number of surplus CPU credits spent to sustain CPU utilization when the CPUCreditBalance value is zero.
- CPUSurplusCreditsCharged – The number of surplus CPU credits exceeding the maximum number of CPU credits (p. 165) 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. 842).

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPUCreditUsage</td>
<td>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).&lt;br&gt;&lt;br&gt;CPU credit metrics are available at a five-minute frequency only. If you specify a period greater than five minutes, use the \textit{Sum} statistic instead of the \textit{Average} statistic.&lt;br&gt;&lt;br&gt;Units: Credits (vCPU-minutes)</td>
</tr>
<tr>
<td>CPUCreditBalance</td>
<td>The number of earned CPU credits that an instance has accrued since it was launched or started. For T2 Standard, the CPUCreditBalance also includes the number of launch credits that have been accrued.&lt;br&gt;&lt;br&gt;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.&lt;br&gt;&lt;br&gt;The credits in the CPUCreditBalance are available for the instance to spend to burst beyond its baseline CPU utilization.&lt;br&gt;&lt;br&gt;When an instance is running, credits in the CPUCreditBalance do not expire. When a T3a or T3 instance stops, the</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CPUCreditBalance</td>
<td>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. CPU credit metrics are available at a five-minute frequency only. Units: Credits (vCPU-minutes)</td>
</tr>
<tr>
<td>CPUSurplusCreditBalance</td>
<td>The number of surplus credits that have been spent by an unlimited instance when its CPUCreditBalance value is zero.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Units: Credits (vCPU-minutes)</td>
</tr>
<tr>
<td>CPUSurplusCreditsCharged</td>
<td>The number of spent surplus credits that are not paid down by earned CPU credits, and which thus incur an additional charge.</td>
</tr>
<tr>
<td></td>
<td>Spent surplus credits are charged when any of the following occurs:</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• The instance is stopped or terminated.</td>
</tr>
<tr>
<td></td>
<td>• The instance is switched from unlimited to standard.</td>
</tr>
<tr>
<td></td>
<td>Units: Credits (vCPU-minutes)</td>
</tr>
</tbody>
</table>

**Calculate 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.

**Calculate 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**

\[
\text{CPUCreditBalance} = \text{prior CPUCreditBalance} + \left[ \text{Credits earned per hour} \times \frac{5}{60} \right] - \text{CPUCreditUsage}
\]
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. 165).

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**

\[
\text{CPUCreditBalance} = 2 + [0.5 - 1] = 1.5
\]

**Calculate CPU credit usage for Unlimited instances**

When a burstable performance instance needs to burst above the baseline, it always spends accrued credits before spending surplus credits. When it depletes its accrued CPU credit balance, it can spend surplus credits to burst CPU for as long as it needs. When CPU utilization falls below the baseline, surplus credits are always paid down before the instance accrues earned credits.

We use the term Adjusted balance in the following equations to reflect the activity that occurs in this five-minute interval. We use this value to arrive at the values for the CPUCreditBalance and CPUSurplusCreditBalance CloudWatch metrics.

**Example**

\[
\text{Adjusted balance} = [\text{prior CPUCreditBalance} - \text{prior CPUSurplusCreditBalance}] + [\text{Credits earned per hour} \times (5/60) - \text{CPUCreditUsage}]
\]

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. 165) that it can accrue.

**Example**

\[
\begin{align*}
\text{CPUCreditBalance} &= \min \left[\max \text{ earned credit balance, Adjusted balance}\right] \\
\text{CPUSurplusCreditBalance} &= 0
\end{align*}
\]
A negative \textit{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 \textit{Adjusted balance} value is assigned to \texttt{CPUSurplusCreditBalance} and \texttt{CPUCreditBalance} is set to 0. Again, the instance size determines the \textit{maximum number of credits (p. 165)} that it can accrue.

\textbf{Example}

\begin{tabular}{|l|}
\hline
\texttt{CPUSurplusCreditBalance} = \texttt{min [max earned credit balance, -Adjusted balance]} \\
\texttt{CPUCreditBalance} = 0 \\
\hline
\end{tabular}

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 \texttt{CPUSurplusCreditsCharged} metric.

\textbf{Example}

\begin{tabular}{|l|}
\hline
\texttt{CPUSurplusCreditsCharged} = \texttt{max [-Adjusted balance - max earned credit balance, 0]} \\
\hline
\end{tabular}

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

\section*{Compute optimized instances}

Compute optimized instances are ideal for compute-bound applications that benefit from high-performance processors.

\subsection*{C5 and C5n instances}

These instances are well suited for the following:

- Batch processing workloads
- Media transcoding
- High-performance web servers
- High-performance computing (HPC)
- Scientific modeling
- Dedicated gaming servers and ad serving engines
- Machine learning inference and other compute-intensive applications

Bare metal instances, such as \texttt{c5.metal}, provide your applications with direct access to physical resources of the host server, such as processors and memory.

For more information, see \texttt{Amazon EC2 C5 Instances}.

\section*{Contents}

- Hardware specifications (p. 194)
- Instance performance (p. 195)
- Network performance (p. 195)
- SSD I/O performance (p. 197)
- Instance features (p. 198)
## Hardware specifications

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

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Memory (GiB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>c4.large</td>
<td>2</td>
<td>3.75</td>
</tr>
<tr>
<td>c4.xlarge</td>
<td>4</td>
<td>7.5</td>
</tr>
<tr>
<td>c4.2xlarge</td>
<td>8</td>
<td>15</td>
</tr>
<tr>
<td>c4.4xlarge</td>
<td>16</td>
<td>30</td>
</tr>
<tr>
<td>c4.8xlarge</td>
<td>36</td>
<td>60</td>
</tr>
<tr>
<td>c5.large</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>c5.xlarge</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>c5.2xlarge</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>c5.4xlarge</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>c5.9xlarge</td>
<td>36</td>
<td>72</td>
</tr>
<tr>
<td>c5.12xlarge</td>
<td>48</td>
<td>96</td>
</tr>
<tr>
<td>c5.18xlarge</td>
<td>72</td>
<td>144</td>
</tr>
<tr>
<td>c5.24xlarge</td>
<td>96</td>
<td>192</td>
</tr>
<tr>
<td>c5.metal</td>
<td>96</td>
<td>192</td>
</tr>
<tr>
<td>c5a.large</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>c5a.xlarge</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>c5a.2xlarge</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>c5a.4xlarge</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>c5a.8xlarge</td>
<td>32</td>
<td>64</td>
</tr>
<tr>
<td>c5a.12xlarge</td>
<td>48</td>
<td>96</td>
</tr>
<tr>
<td>c5a.16xlarge</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>c5a.24xlarge</td>
<td>96</td>
<td>192</td>
</tr>
<tr>
<td>c5ad.large</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>c5ad.xlarge</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>c5ad.2xlarge</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>c5ad.4xlarge</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>c5ad.8xlarge</td>
<td>32</td>
<td>64</td>
</tr>
<tr>
<td>Instance type</td>
<td>Default vCPUs</td>
<td>Memory (GiB)</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>c5ad.12xlarge</td>
<td>48</td>
<td>96</td>
</tr>
<tr>
<td>c5ad.16xlarge</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>c5ad.24xlarge</td>
<td>96</td>
<td>192</td>
</tr>
<tr>
<td>c5d.large</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>c5d.xlarge</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>c5d.2xlarge</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>c5d.4xlarge</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>c5d.9xlarge</td>
<td>36</td>
<td>72</td>
</tr>
<tr>
<td>c5d.12xlarge</td>
<td>48</td>
<td>96</td>
</tr>
<tr>
<td>c5d.18xlarge</td>
<td>72</td>
<td>144</td>
</tr>
<tr>
<td>c5d.24xlarge</td>
<td>96</td>
<td>192</td>
</tr>
<tr>
<td>c5d.metal</td>
<td>96</td>
<td>192</td>
</tr>
<tr>
<td>c5n.large</td>
<td>2</td>
<td>5.25</td>
</tr>
<tr>
<td>c5n.xlarge</td>
<td>4</td>
<td>10.5</td>
</tr>
<tr>
<td>c5n.2xlarge</td>
<td>8</td>
<td>21</td>
</tr>
<tr>
<td>c5n.4xlarge</td>
<td>16</td>
<td>42</td>
</tr>
<tr>
<td>c5n.9xlarge</td>
<td>36</td>
<td>96</td>
</tr>
<tr>
<td>c5n.18xlarge</td>
<td>72</td>
<td>192</td>
</tr>
<tr>
<td>c5n.metal</td>
<td>72</td>
<td>192</td>
</tr>
</tbody>
</table>

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 Optimize CPU options (p. 540).

**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. 1344).

**Network performance**

You can enable enhanced networking on supported instance types to provide lower latencies, lower network jitter, and higher packet-per-second (PPS) performance. Most applications do not consistently need a high level of network performance, but can benefit from access to increased bandwidth when they send or receive data. For more information, see Enhanced networking on Windows (p. 960).

The following is a summary of network performance for compute optimized instances that support enhanced networking.
### Instance type | Network performance | Enhanced networking
--- | --- | ---
c4.large | Moderate | Intel 82599 VF (p. 968)
c4.xlarge | High | Intel 82599 VF (p. 968)
c5.4xlarge and smaller | Up to 10 Gbps † | ENA (p. 961)
c4.8xlarge | 10 Gbps | Intel 82599 VF (p. 968)
c5.9xlarge | 10 Gbps | ENA (p. 961)
c5.12xlarge | 12 Gbps | ENA (p. 961)
c5n.4xlarge and smaller | Up to 25 Gbps † | ENA (p. 961)
c5.18xlarge | 25 Gbps | ENA (p. 961)
c5n.9xlarge | 50 Gbps | ENA (p. 961)
c5n.18xlarge | 100 Gbps | ENA (p. 961)

† These instances have a baseline bandwidth and can use a network I/O credit mechanism to burst beyond their baseline bandwidth on a best effort basis. For more information, see instance network bandwidth (p. 959).

### Instance type | Baseline bandwidth (Gbps) | Burst bandwidth (Gbps)
c5.large | .75 | 10
c5.xlarge | 1.25 | 10
c5.2xlarge | 2.5 | 10
c5.4xlarge | 5 | 10
c5a.large | .75 | 10
c5a.xlarge | 1.25 | 10
c5a.2xlarge | 2.5 | 10
c5a.4xlarge | 5 | 10
c5ad.large | .75 | 10
c5ad.xlarge | 1.25 | 10
c5ad.2xlarge | 2.5 | 10
c5ad.4xlarge | 5 | 10
### 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.

<table>
<thead>
<tr>
<th>Instance Size</th>
<th>100% Random Read IOPS</th>
<th>Write IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>c5d.large</td>
<td>16,283</td>
<td>7,105</td>
</tr>
<tr>
<td>c5d.xlarge</td>
<td>32,566</td>
<td>14,211</td>
</tr>
<tr>
<td>c5d.2xlarge</td>
<td>65,132</td>
<td>28,421</td>
</tr>
<tr>
<td>c5d.4xlarge</td>
<td>130,263</td>
<td>56,842</td>
</tr>
<tr>
<td>c5d.8xlarge</td>
<td>260,526</td>
<td>113,684</td>
</tr>
<tr>
<td>c5d.12xlarge</td>
<td>412,500</td>
<td>180,000</td>
</tr>
<tr>
<td>c5d.16xlarge</td>
<td>521,053</td>
<td>227,368</td>
</tr>
<tr>
<td>c5d.24xlarge</td>
<td>825,000</td>
<td>360,000</td>
</tr>
<tr>
<td>c5d.large*</td>
<td>20,000</td>
<td>9,000</td>
</tr>
<tr>
<td>c5d.xlarge*</td>
<td>40,000</td>
<td>18,000</td>
</tr>
<tr>
<td>c5d.2xlarge*</td>
<td>80,000</td>
<td>37,000</td>
</tr>
<tr>
<td>c5d.4xlarge*</td>
<td>175,000</td>
<td>75,000</td>
</tr>
<tr>
<td>c5d.9xlarge</td>
<td>350,000</td>
<td>170,000</td>
</tr>
<tr>
<td>c5d.12xlarge</td>
<td>700,000</td>
<td>340,000</td>
</tr>
<tr>
<td>c5d.18xlarge</td>
<td>700,000</td>
<td>340,000</td>
</tr>
<tr>
<td>c5d.24xlarge</td>
<td>1,400,000</td>
<td>680,000</td>
</tr>
<tr>
<td>c5d.metal</td>
<td>1,400,000</td>
<td>680,000</td>
</tr>
</tbody>
</table>
* 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. 1404).

Instance features

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

<table>
<thead>
<tr>
<th></th>
<th>EBS only</th>
<th>NVMe EBS</th>
<th>Instance store</th>
<th>Placement group</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>C5</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>C5a</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>C5ad</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
</tr>
<tr>
<td>C5d</td>
<td>No</td>
<td>Yes</td>
<td>NVMe *</td>
<td>Yes</td>
</tr>
<tr>
<td>C5n</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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

For more information, see the following:

- Amazon EBS and NVMe on Windows instances (p. 1343)
- Amazon EC2 instance store (p. 1392)
- Placement groups (p. 975)

Release notes

- C5 and C5d instances feature a 3.1 GHz Intel Xeon Platinum 8000 series processor from either the first generation (Skylake-SP) or second generation (Cascade Lake).
• C5a and C5ad instances feature a second-generation AMD EPYC processor (Rome) running at frequencies as high as 3.3 GHz.
• C4 instances and instances built on the Nitro System (p. 146) require 64-bit EBS-backed HVM AMIs. They have high-memory and require a 64-bit operating system to take advantage of that capacity. HVM AMIs provide superior performance in comparison to paravirtual (PV) AMIs on high-memory instance types. In addition, you must use an HVM AMI to take advantage of enhanced networking.
• Instances built on the Nitro System have the following requirements:
  • NVMe drivers (p. 1343) must be installed
  • Elastic Network Adapter (ENA) drivers (p. 961) must be installed

The current AWS Windows AMIs (p. 27) meet these requirements.
• Instances built on the Nitro System instances support a maximum of 28 attachments, including network interfaces, EBS volumes, and NVMe instance store volumes. For more information, see Nitro System volume limits (p. 1407).
• Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
• To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support.
• Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
• There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see How many instances can I run in Amazon EC2? in the Amazon EC2 FAQ.

Memory optimized instances

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

R5, R5a, R5b, and R5n instances

These instances are well suited for the following:
• High-performance, relational (MySQL) and NoSQL (MongoDB, Cassandra) databases.
• Distributed web scale cache stores that provide in-memory caching of key-value type data (Memcached and Redis).
• In-memory databases using optimized data storage formats and analytics for business intelligence (for example, SAP HANA).
• Applications performing real-time processing of big unstructured data (financial services, Hadoop/Spark clusters).
• High-performance computing (HPC) and Electronic Design Automation (EDA) applications.

R5b instances support io2 Block Express volumes. All io2 volumes attached to an R5b instance during or after launch automatically run on EBS Block Express. For more information, see io2 Block Express volumes.

Bare metal instances, such as r5.metal, provide your applications with direct access to physical resources of the host server, such as processors and memory.

For more information, see Amazon EC2 R5 Instances.
High memory (u-*) instances

These instances offer 6 TiB, 9 TiB, 12 TiB, 18 TiB, and 24 TiB of memory per instance. They are designed to run large in-memory databases, including production deployments of the SAP HANA in-memory database.

For more information, see Amazon EC2 High Memory Instances and Storage Configuration for SAP HANA. For information about supported operating systems, see Migrating SAP HANA on AWS to an EC2 High Memory Instance.

X1 instances

These instances are well suited for the following:

- In-memory databases such as SAP HANA, including SAP-certified support for Business Suite S/4HANA, Business Suite on HANA (SoH), Business Warehouse on HANA (BW), and Data Mart Solutions on HANA. For more information, see SAP HANA on the AWS Cloud.
- Big-data processing engines such as Apache Spark or Presto.
- High-performance computing (HPC) applications.

For more information, see Amazon EC2 X1 Instances.

X1e instances

These instances are well suited for the following:

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

For more information, see Amazon EC2 X1e Instances.

z1d instances

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

- Electronic Design Automation (EDA)
- Relational database workloads

z1d.metal instances provide your applications with direct access to physical resources of the host server, such as processors and memory.

For more information, see Amazon EC2 z1d Instances.

Contents

- Hardware specifications (p. 201)
- Memory performance (p. 204)
- Instance performance (p. 204)
- Network performance (p. 204)
- SSD I/O performance (p. 206)
- Instance features (p. 208)
- High availability and reliability (X1) (p. 209)
- Support for vCPUs (p. 209)
Hardware specifications

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

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Memory (GiB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>r4.large</td>
<td>2</td>
<td>15.25</td>
</tr>
<tr>
<td>r4.xlarge</td>
<td>4</td>
<td>30.5</td>
</tr>
<tr>
<td>r4.2xlarge</td>
<td>8</td>
<td>61</td>
</tr>
<tr>
<td>r4.4xlarge</td>
<td>16</td>
<td>122</td>
</tr>
<tr>
<td>r4.8xlarge</td>
<td>32</td>
<td>244</td>
</tr>
<tr>
<td>r4.16xlarge</td>
<td>64</td>
<td>488</td>
</tr>
<tr>
<td>r5.large</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>r5.xlarge</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>r5.2xlarge</td>
<td>8</td>
<td>64</td>
</tr>
<tr>
<td>r5.4xlarge</td>
<td>16</td>
<td>128</td>
</tr>
<tr>
<td>r5.8xlarge</td>
<td>32</td>
<td>256</td>
</tr>
<tr>
<td>r5.12xlarge</td>
<td>48</td>
<td>384</td>
</tr>
<tr>
<td>r5.16xlarge</td>
<td>64</td>
<td>512</td>
</tr>
<tr>
<td>r5.24xlarge</td>
<td>96</td>
<td>768</td>
</tr>
<tr>
<td>r5.meta</td>
<td>96</td>
<td>768</td>
</tr>
<tr>
<td>r5a.large</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>r5a.xlarge</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>r5a.2xlarge</td>
<td>8</td>
<td>64</td>
</tr>
<tr>
<td>r5a.4xlarge</td>
<td>16</td>
<td>128</td>
</tr>
<tr>
<td>r5a.8xlarge</td>
<td>32</td>
<td>256</td>
</tr>
<tr>
<td>r5a.12xlarge</td>
<td>48</td>
<td>384</td>
</tr>
<tr>
<td>r5a.16xlarge</td>
<td>64</td>
<td>512</td>
</tr>
<tr>
<td>r5a.24xlarge</td>
<td>96</td>
<td>768</td>
</tr>
<tr>
<td>r5ad.large</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>r5ad.xlarge</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>r5ad.2xlarge</td>
<td>8</td>
<td>64</td>
</tr>
<tr>
<td>r5ad.4xlarge</td>
<td>16</td>
<td>128</td>
</tr>
<tr>
<td>Instance type</td>
<td>Default vCPUs</td>
<td>Memory (GiB)</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>r5ad.8xlarge</td>
<td>32</td>
<td>256</td>
</tr>
<tr>
<td>r5ad.12xlarge</td>
<td>48</td>
<td>384</td>
</tr>
<tr>
<td>r5ad.16xlarge</td>
<td>64</td>
<td>512</td>
</tr>
<tr>
<td>r5ad.24xlarge</td>
<td>96</td>
<td>768</td>
</tr>
<tr>
<td>r5b.large</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>r5b.xlarge</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>r5b.2xlarge</td>
<td>8</td>
<td>64</td>
</tr>
<tr>
<td>r5b.4xlarge</td>
<td>16</td>
<td>128</td>
</tr>
<tr>
<td>r5b.8xlarge</td>
<td>32</td>
<td>256</td>
</tr>
<tr>
<td>r5b.12xlarge</td>
<td>48</td>
<td>384</td>
</tr>
<tr>
<td>r5b.16xlarge</td>
<td>64</td>
<td>512</td>
</tr>
<tr>
<td>r5b.24xlarge</td>
<td>96</td>
<td>768</td>
</tr>
<tr>
<td>r5b.metal</td>
<td>96</td>
<td>768</td>
</tr>
<tr>
<td>r5d.large</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>r5d.xlarge</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>r5d.2xlarge</td>
<td>8</td>
<td>64</td>
</tr>
<tr>
<td>r5d.4xlarge</td>
<td>16</td>
<td>128</td>
</tr>
<tr>
<td>r5d.8xlarge</td>
<td>32</td>
<td>256</td>
</tr>
<tr>
<td>r5d.12xlarge</td>
<td>48</td>
<td>384</td>
</tr>
<tr>
<td>r5d.16xlarge</td>
<td>64</td>
<td>512</td>
</tr>
<tr>
<td>r5d.24xlarge</td>
<td>96</td>
<td>768</td>
</tr>
<tr>
<td>r5d.metal</td>
<td>96</td>
<td>768</td>
</tr>
<tr>
<td>r5dn.large</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>r5dn.xlarge</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>r5dn.2xlarge</td>
<td>8</td>
<td>64</td>
</tr>
<tr>
<td>r5dn.4xlarge</td>
<td>16</td>
<td>128</td>
</tr>
<tr>
<td>r5dn.8xlarge</td>
<td>32</td>
<td>256</td>
</tr>
<tr>
<td>r5dn.12xlarge</td>
<td>48</td>
<td>384</td>
</tr>
<tr>
<td>r5dn.16xlarge</td>
<td>64</td>
<td>512</td>
</tr>
<tr>
<td>r5dn.24xlarge</td>
<td>96</td>
<td>768</td>
</tr>
<tr>
<td>r5dn.metal</td>
<td>96</td>
<td>768</td>
</tr>
<tr>
<td>Instance type</td>
<td>Default vCPUs</td>
<td>Memory (GiB)</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>r5n.large</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>r5n.xlarge</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>r5n.2xlarge</td>
<td>8</td>
<td>64</td>
</tr>
<tr>
<td>r5n.4xlarge</td>
<td>16</td>
<td>128</td>
</tr>
<tr>
<td>r5n.8xlarge</td>
<td>32</td>
<td>256</td>
</tr>
<tr>
<td>r5n.12xlarge</td>
<td>48</td>
<td>384</td>
</tr>
<tr>
<td>r5n.16xlarge</td>
<td>64</td>
<td>512</td>
</tr>
<tr>
<td>r5n.24xlarge</td>
<td>96</td>
<td>768</td>
</tr>
<tr>
<td>r5n.metal</td>
<td>96</td>
<td>768</td>
</tr>
<tr>
<td>u-6tb1.56xlarge</td>
<td>224</td>
<td>6,144</td>
</tr>
<tr>
<td>u-6tb1.112xlarge</td>
<td>448</td>
<td>6,144</td>
</tr>
<tr>
<td>u-6tb1.metal</td>
<td>448 *</td>
<td>6,144</td>
</tr>
<tr>
<td>u-9tb1.112xlarge</td>
<td>448</td>
<td>9,216</td>
</tr>
<tr>
<td>u-9tb1.metal</td>
<td>448 *</td>
<td>9,216</td>
</tr>
<tr>
<td>u-12tb1.112xlarge</td>
<td>448</td>
<td>12,288</td>
</tr>
<tr>
<td>u-12tb1.metal</td>
<td>448 *</td>
<td>12,288</td>
</tr>
<tr>
<td>u-18tb1.metal</td>
<td>448 *</td>
<td>18,432</td>
</tr>
<tr>
<td>u-24tb1.metal</td>
<td>448 *</td>
<td>24,576</td>
</tr>
<tr>
<td>x1.16xlarge</td>
<td>64</td>
<td>976</td>
</tr>
<tr>
<td>x1.32xlarge</td>
<td>128</td>
<td>1,952</td>
</tr>
<tr>
<td>x1e.xlarge</td>
<td>4</td>
<td>122</td>
</tr>
<tr>
<td>x1e.2xlarge</td>
<td>8</td>
<td>244</td>
</tr>
<tr>
<td>x1e.4xlarge</td>
<td>16</td>
<td>488</td>
</tr>
<tr>
<td>x1e.8xlarge</td>
<td>32</td>
<td>976</td>
</tr>
<tr>
<td>x1e.16xlarge</td>
<td>64</td>
<td>1,952</td>
</tr>
<tr>
<td>x1e.32xlarge</td>
<td>128</td>
<td>3,904</td>
</tr>
<tr>
<td>z1d.large</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>z1d.xlarge</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>z1d.2xlarge</td>
<td>8</td>
<td>64</td>
</tr>
<tr>
<td>z1d.3xlarge</td>
<td>12</td>
<td>96</td>
</tr>
<tr>
<td>z1d.6xlarge</td>
<td>24</td>
<td>192</td>
</tr>
</tbody>
</table>
Memory optimized

**Instance type** | **Default vCPUs** | **Memory (GiB)**
---|---|---
`z1d.12xlarge` | 48 | 384
`z1d.metal` | 48 | 384

* Each logical processor is a hyperthread on 224 cores.

For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instance Types.

For more information about specifying CPU options, see Optimize CPU options (p. 540).

**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. 201).

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**

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 on supported instance types to provide lower latencies, lower network jitter, and higher packet-per-second (PPS) performance. Most applications do not consistently need a high level of network performance, but can benefit from access to increased bandwidth when they send or receive data. For more information, see Enhanced networking on Windows (p. 960).

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

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Network performance</th>
<th>Enhanced networking</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>r4.4xlarge</code> and smaller</td>
<td><code>r5.4xlarge</code> and smaller</td>
<td><code>r5a.8xlarge</code> and smaller</td>
</tr>
<tr>
<td><code>r4.8xlarge</code></td>
<td><code>r5.8xlarge</code></td>
<td><code>r5.12xlarge</code></td>
</tr>
</tbody>
</table>
## Memory optimized Instance type

<table>
<thead>
<tr>
<th>Instance</th>
<th>Network performance</th>
<th>Enhanced networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>r5d.12xlarge</td>
<td>x1.16xlarge</td>
<td>xle.16xlarge</td>
</tr>
<tr>
<td>r5a.16xlarge</td>
<td>r5d.16xlarge</td>
<td>20 Gbps</td>
</tr>
<tr>
<td>r5.16xlarge</td>
<td>r5a.24xlarge</td>
<td>r5ad.24xlarge</td>
</tr>
<tr>
<td>r5dn.4xlarge and smaller</td>
<td>r5n.4xlarge and smaller</td>
<td>100 Gbps</td>
</tr>
<tr>
<td>r4.16xlarge</td>
<td>r5.24xlarge</td>
<td>r5.meta</td>
</tr>
<tr>
<td>r5dn.12xlarge</td>
<td>r5n.12xlarge</td>
<td>50 Gbps</td>
</tr>
<tr>
<td>r5dn.16xlarge</td>
<td>r5n.16xlarge</td>
<td>75 Gbps</td>
</tr>
<tr>
<td>r5dn.24xlarge</td>
<td>r5dn.meta</td>
<td>r5n.24xlarge</td>
</tr>
</tbody>
</table>

* Instances of this type launched after March 12, 2020 provide network performance of 100 Gbps. Instances of this type launched before March 12, 2020 might only provide network performance of 25 Gbps. To ensure that instances launched before March 12, 2020 have a network performance of 100 Gbps, contact your account team to upgrade your instance at no additional cost.

† These instances have a baseline bandwidth and can use a network I/O credit mechanism to burst beyond their baseline bandwidth on a best effort basis. For more information, see instance network bandwidth (p. 959).

## Instance type Baseline bandwidth (Gbps) Burst bandwidth (Gbps)

<table>
<thead>
<tr>
<th>Instance</th>
<th>Baseline bandwidth (Gbps)</th>
<th>Burst bandwidth (Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>r5.large</td>
<td>.75</td>
<td>10</td>
</tr>
<tr>
<td>r5.xlarge</td>
<td>1.25</td>
<td>10</td>
</tr>
<tr>
<td>r5.2xlarge</td>
<td>2.5</td>
<td>10</td>
</tr>
<tr>
<td>r5.4xlarge</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>r5a.large</td>
<td>.75</td>
<td>10</td>
</tr>
<tr>
<td>r5a.xlarge</td>
<td>1.25</td>
<td>10</td>
</tr>
<tr>
<td>r5a.2xlarge</td>
<td>2.5</td>
<td>10</td>
</tr>
<tr>
<td>r5a.4xlarge</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>r5a.8xlarge</td>
<td>7.5</td>
<td>10</td>
</tr>
</tbody>
</table>
### Instance type

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Baseline bandwidth (Gbps)</th>
<th>Burst bandwidth (Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>r5ad.large</td>
<td>.75</td>
<td>10</td>
</tr>
<tr>
<td>r5ad.xlarge</td>
<td>1.25</td>
<td>10</td>
</tr>
<tr>
<td>r5ad.2xlarge</td>
<td>2.5</td>
<td>10</td>
</tr>
<tr>
<td>r5ad.4xlarge</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>r5ad.8xlarge</td>
<td>7.5</td>
<td>10</td>
</tr>
<tr>
<td>r5b.large</td>
<td>.75</td>
<td>10</td>
</tr>
<tr>
<td>r5b.xlarge</td>
<td>1.25</td>
<td>10</td>
</tr>
<tr>
<td>r5b.2xlarge</td>
<td>2.5</td>
<td>10</td>
</tr>
<tr>
<td>r5b.4xlarge</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>r5d.large</td>
<td>.75</td>
<td>10</td>
</tr>
<tr>
<td>r5d.xlarge</td>
<td>1.25</td>
<td>10</td>
</tr>
<tr>
<td>r5d.2xlarge</td>
<td>2.5</td>
<td>10</td>
</tr>
<tr>
<td>r5d.4xlarge</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>r5dn.large</td>
<td>2.1</td>
<td>25</td>
</tr>
<tr>
<td>r5dn.xlarge</td>
<td>4.1</td>
<td>25</td>
</tr>
<tr>
<td>r5dn.2xlarge</td>
<td>8.125</td>
<td>25</td>
</tr>
<tr>
<td>r5dn.4xlarge</td>
<td>16.25</td>
<td>25</td>
</tr>
<tr>
<td>r5n.large</td>
<td>2.1</td>
<td>25</td>
</tr>
<tr>
<td>r5n.xlarge</td>
<td>4.1</td>
<td>25</td>
</tr>
<tr>
<td>r5n.2xlarge</td>
<td>8.125</td>
<td>25</td>
</tr>
<tr>
<td>r5n.4xlarge</td>
<td>16.25</td>
<td>25</td>
</tr>
<tr>
<td>z1d.large</td>
<td>.75</td>
<td>10</td>
</tr>
<tr>
<td>z1d.xlarge</td>
<td>1.25</td>
<td>10</td>
</tr>
<tr>
<td>z1d.2xlarge</td>
<td>2.5</td>
<td>10</td>
</tr>
<tr>
<td>z1d.3xlarge</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

### 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.
<table>
<thead>
<tr>
<th>Instance Size</th>
<th>100% Random Read IOPS</th>
<th>Write IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>r5ad.large *</td>
<td>30,000</td>
<td>15,000</td>
</tr>
<tr>
<td>r5ad.xlarge *</td>
<td>59,000</td>
<td>29,000</td>
</tr>
<tr>
<td>r5ad.2xlarge *</td>
<td>117,000</td>
<td>57,000</td>
</tr>
<tr>
<td>r5ad.4xlarge *</td>
<td>234,000</td>
<td>114,000</td>
</tr>
<tr>
<td>r5ad.8xlarge</td>
<td>466,666</td>
<td>233,333</td>
</tr>
<tr>
<td>r5ad.12xlarge</td>
<td>700,000</td>
<td>340,000</td>
</tr>
<tr>
<td>r5ad.16xlarge</td>
<td>933,333</td>
<td>466,666</td>
</tr>
<tr>
<td>r5ad.24xlarge</td>
<td>1,400,000</td>
<td>680,000</td>
</tr>
<tr>
<td>r5d.large *</td>
<td>30,000</td>
<td>15,000</td>
</tr>
<tr>
<td>r5d.xlarge *</td>
<td>59,000</td>
<td>29,000</td>
</tr>
<tr>
<td>r5d.2xlarge *</td>
<td>117,000</td>
<td>57,000</td>
</tr>
<tr>
<td>r5d.4xlarge *</td>
<td>234,000</td>
<td>114,000</td>
</tr>
<tr>
<td>r5d.8xlarge</td>
<td>466,666</td>
<td>233,333</td>
</tr>
<tr>
<td>r5d.12xlarge</td>
<td>700,000</td>
<td>340,000</td>
</tr>
<tr>
<td>r5d.16xlarge</td>
<td>933,333</td>
<td>466,666</td>
</tr>
<tr>
<td>r5d.24xlarge</td>
<td>1,400,000</td>
<td>680,000</td>
</tr>
<tr>
<td>r5d.metal</td>
<td>1,400,000</td>
<td>680,000</td>
</tr>
<tr>
<td>r5dn.large *</td>
<td>30,000</td>
<td>15,000</td>
</tr>
<tr>
<td>r5dn.xlarge *</td>
<td>59,000</td>
<td>29,000</td>
</tr>
<tr>
<td>r5dn.2xlarge *</td>
<td>117,000</td>
<td>57,000</td>
</tr>
<tr>
<td>r5dn.4xlarge *</td>
<td>234,000</td>
<td>114,000</td>
</tr>
<tr>
<td>r5dn.8xlarge</td>
<td>466,666</td>
<td>233,333</td>
</tr>
<tr>
<td>r5dn.12xlarge</td>
<td>700,000</td>
<td>340,000</td>
</tr>
<tr>
<td>r5dn.16xlarge</td>
<td>933,333</td>
<td>466,666</td>
</tr>
<tr>
<td>r5dn.24xlarge</td>
<td>1,400,000</td>
<td>680,000</td>
</tr>
<tr>
<td>r5dn.metal</td>
<td>1,400,000</td>
<td>680,000</td>
</tr>
<tr>
<td>z1d.large *</td>
<td>30,000</td>
<td>15,000</td>
</tr>
<tr>
<td>z1d.xlarge *</td>
<td>59,000</td>
<td>29,000</td>
</tr>
<tr>
<td>z1d.2xlarge *</td>
<td>117,000</td>
<td>57,000</td>
</tr>
<tr>
<td>z1d.3xlarge *</td>
<td>175,000</td>
<td>75,000</td>
</tr>
<tr>
<td>z1d.6xlarge</td>
<td>350,000</td>
<td>170,000</td>
</tr>
</tbody>
</table>
Memory optimized

<table>
<thead>
<tr>
<th>Instance Size</th>
<th>100% Random Read IOPS</th>
<th>Write IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>z1d.12xlarge</td>
<td>700,000</td>
<td>340,000</td>
</tr>
<tr>
<td>z1d.metal</td>
<td>700,000</td>
<td>340,000</td>
</tr>
</tbody>
</table>

* 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. 1404).

### Instance features

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

<table>
<thead>
<tr>
<th>Pattern</th>
<th>EBS only</th>
<th>NVMe EBS</th>
<th>Instance store</th>
<th>Placement group</th>
</tr>
</thead>
<tbody>
<tr>
<td>R4</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>R5</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>R5a</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>R5ad</td>
<td>No</td>
<td>Yes</td>
<td>NVME *</td>
<td>Yes</td>
</tr>
<tr>
<td>R5b</td>
<td>Yes **</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>R5d</td>
<td>No</td>
<td>Yes</td>
<td>NVME *</td>
<td>Yes</td>
</tr>
<tr>
<td>R5dn</td>
<td>No</td>
<td>Yes</td>
<td>NVME *</td>
<td>Yes</td>
</tr>
<tr>
<td>R5n</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
| High memory | Yes  | Yes | No | Virtualized: Yes
Bare metal: No |
<p>| X1      | No       | No       | SSD            | Yes             |</p>
<table>
<thead>
<tr>
<th>EBS only</th>
<th>NVMe EBS</th>
<th>Instance store</th>
<th>Placement group</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1e</td>
<td>No</td>
<td>No</td>
<td>SSD *</td>
</tr>
<tr>
<td>z1d</td>
<td>No</td>
<td>Yes</td>
<td>NVME *</td>
</tr>
</tbody>
</table>

** All io2 volumes attached to an R5b instance during or after launch automatically run on EBS Block Express. For more information, see io2 Block Express volumes.

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

For more information, see the following:
- Amazon EBS and NVMe on Windows instances (p. 1343)
- Amazon EC2 instance store (p. 1392)
- Placement groups (p. 975)

**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. 447).

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 2019
- Windows Server 2016
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2 64-bit
- Windows Server 2008 SP2 64-bit
Release notes

- R4 instances feature up to 64 vCPUs and are powered by two AWS-customized Intel XEON processors based on E5-2686v4 that feature high-memory bandwidth and larger L3 caches to boost the performance of in-memory applications.
- R5, R5b, and R5d instances feature a 3.1 GHz Intel Xeon Platinum 8000 series processor from either the first generation (Skylake-SP) or second generation (Cascade Lake).
- R5a and R5ad instances feature a 2.5 GHz AMD EPYC 7000 series processor.
- High memory instances (u-6tb1.metal, u-9tb1.metal, and u-12tb1.metal) are the first instances to be powered by an eight-socket platform with the latest generation Intel Xeon Platinum 8176M (Skylake) processors that are optimized for mission-critical enterprise workloads. High Memory instances with 18 TB and 24 TB of memory (u-18tb1.metal and u-24tb1.metal) are the first instances powered by an 8-socket platform with 2nd Generation Intel Xeon Scalable 8280L (Cascade Lake) processors.
- X1e and X1 instances feature up to 128 vCPUs and are powered by four Intel Xeon E7-8880 v3 processors that feature high-memory bandwidth and larger L3 caches to boost the performance of in-memory applications.
- Instances built on the Nitro System have the following requirements:
  - NVMe drivers (p. 1343) must be installed
  - Elastic Network Adapter (ENA) drivers (p. 961) must be installed

The current AWS Windows AMIs (p. 27) meet these requirements.

- Instances built on the Nitro System instances support a maximum of 28 attachments, including network interfaces, EBS volumes, and NVMe instance store volumes. For more information, see Nitro System volume limits (p. 1407).
- All io2 volumes attached to an R5b instance during or after launch automatically run on EBS Block Express. For more information, see io2 Block Express volumes.
- Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
- To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support.
- Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
- You can’t launch X1 instances using a Windows Server 2008 SP2 64-bit AMI, except for x1.16xlarge instances.
- You can’t launch X1e instances using a Windows Server 2008 SP2 64-bit AMI.
- With earlier versions of the Windows Server 2008 R2 64-bit AMI, you can’t launch r4.large and r4.4xlarge instances. If you experience this issue, update to the latest version of this AMI.
- There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see How many instances can I run in Amazon EC2? in the Amazon EC2 FAQ.

Storage optimized instances

Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage. They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications.
D2 instances

These instances are well suited for the following:

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

D3 and D3en instances

These instances offer scale out of instance storage and are well suited for the following:

- Distributed file systems for Hadoop workloads
- File storage workloads such as GPFC and BeeFS
- Large data lakes for HPC workloads

H1 instances

These instances are well suited for the following:

- Data-intensive workloads such as MapReduce and distributed file systems
- Applications requiring sequential access to large amounts of data on direct-attached instance storage
- Applications that require high-throughput access to large quantities of data

I3 and I3en instances

These instances are well suited for the following:

- High frequency online transaction processing (OLTP) systems
- Relational databases
- NoSQL databases
- Cache for in-memory databases (for example, Redis)
- Data warehousing applications
- Distributed file systems

Bare metal instances provide your applications with direct access to physical resources of the host server, such as processors and memory.

For more information, see Amazon EC2 I3 Instances.

Contents

- Hardware specifications (p. 212)
- Instance performance (p. 213)
- Network performance (p. 213)
- SSD I/O performance (p. 214)
- Instance features (p. 215)
- Release notes (p. 216)
The primary data storage for D2, D3, and D3en instances is HDD instance store volumes. The primary data storage for I3 and I3en instances is non-volatile memory express (NVMe) SSD instance store volumes.

Instance store volumes persist only for the life of the instance. When you stop, hibernate, 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. 1392) and SSD instance store volumes (p. 1403).

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

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Memory (GiB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>d2.xlarge</td>
<td>4</td>
<td>30.5</td>
</tr>
<tr>
<td>d2.2xlarge</td>
<td>8</td>
<td>61</td>
</tr>
<tr>
<td>d2.4xlarge</td>
<td>16</td>
<td>122</td>
</tr>
<tr>
<td>d2.8xlarge</td>
<td>36</td>
<td>244</td>
</tr>
<tr>
<td>d3.xlarge</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>d3.2xlarge</td>
<td>8</td>
<td>64</td>
</tr>
<tr>
<td>d3.4xlarge</td>
<td>16</td>
<td>128</td>
</tr>
<tr>
<td>d3.8xlarge</td>
<td>32</td>
<td>256</td>
</tr>
<tr>
<td>d3en.large</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>d3en.xlarge</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>d3en.2xlarge</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>d3en.4xlarge</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>d3en.6xlarge</td>
<td>24</td>
<td>96</td>
</tr>
<tr>
<td>d3en.8xlarge</td>
<td>32</td>
<td>128</td>
</tr>
<tr>
<td>d3en.12xlarge</td>
<td>48</td>
<td>192</td>
</tr>
<tr>
<td>h1.2xlarge</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>h1.4xlarge</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>h1.8xlarge</td>
<td>32</td>
<td>128</td>
</tr>
<tr>
<td>h1.16xlarge</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>i3.large</td>
<td>2</td>
<td>15.25</td>
</tr>
<tr>
<td>i3.xlarge</td>
<td>4</td>
<td>30.5</td>
</tr>
<tr>
<td>i3.2xlarge</td>
<td>8</td>
<td>61</td>
</tr>
<tr>
<td>i3.4xlarge</td>
<td>16</td>
<td>122</td>
</tr>
<tr>
<td>i3.8xlarge</td>
<td>32</td>
<td>244</td>
</tr>
</tbody>
</table>
### Storage optimized

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Memory (GiB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>i3.16xlarge</td>
<td>64</td>
<td>488</td>
</tr>
<tr>
<td>i3.metal</td>
<td>72</td>
<td>512</td>
</tr>
<tr>
<td>i3en.large</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>i3en.xlarge</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>i3en.2xlarge</td>
<td>8</td>
<td>64</td>
</tr>
<tr>
<td>i3en.3xlarge</td>
<td>12</td>
<td>96</td>
</tr>
<tr>
<td>i3en.6xlarge</td>
<td>24</td>
<td>192</td>
</tr>
<tr>
<td>i3en.12xlarge</td>
<td>48</td>
<td>384</td>
</tr>
<tr>
<td>i3en.24xlarge</td>
<td>96</td>
<td>768</td>
</tr>
<tr>
<td>i3en.metal</td>
<td>96</td>
<td>768</td>
</tr>
</tbody>
</table>

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 Optimize CPU options (p. 540).

#### 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. 538).

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. 1344).

#### Network performance

You can enable enhanced networking on supported instance types to provide lower latencies, lower network jitter, and higher packet-per-second (PPS) performance. Most applications do not consistently need a high level of network performance, but can benefit from access to increased bandwidth when they send or receive data. For more information, see Enhanced networking on Windows (p. 960).

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

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Network performance</th>
<th>Enhanced networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>d2.xlarge</td>
<td>Moderate</td>
<td>Intel 82599 VF (p. 968)</td>
</tr>
<tr>
<td>d2.2xlarge</td>
<td>d2.4xlarge</td>
<td>High</td>
</tr>
<tr>
<td>i3.4xlarge and smaller</td>
<td>Up to 10 Gbps †</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>d2.8xlarge</td>
<td>10 Gbps</td>
<td>Intel 82599 VF (p. 968)</td>
</tr>
<tr>
<td>i3.8xlarge</td>
<td>h1.8xlarge</td>
<td>10 Gbps</td>
</tr>
</tbody>
</table>
Storage optimized

Instance type | Network performance | Enhanced networking  
---|---|---  
d3.4xlarge and smaller | Up to 15 Gbps † | ENA (p. 961)  
d3en.2xlarge and smaller | Up to 25 Gbps † | ENA (p. 961)  
d3.8xlarge | 25 Gbps | ENA (p. 961)  
d3en.6xlarge | 40 Gbps | ENA (p. 961)  
d3.8xlarge | 50 Gbps | ENA (p. 961)  
d3en.12xlarge | 75 Gbps | ENA (p. 961)  
i3en.24xlarge | 100 Gbps | ENA (p. 961)  

† These instances have a baseline bandwidth and can use a network I/O credit mechanism to burst beyond their baseline bandwidth on a best effort basis. For more information, see instance network bandwidth (p. 959).

| Instance type | Baseline bandwidth (Gbps) | Burst bandwidth (Gbps)  
---|---|---  
d3.xlarge | 3 | 15  
d3.2xlarge | 6 | 15  
d3.4xlarge | 12.5 | 15  
d3en.large | 3 | 25  
d3en.xlarge | 6 | 25  
d3en.2xlarge | 12.5 | 25  
i3en.large | 2.1 | 25  
i3en.xlarge | 4.2 | 25  
i3en.2xlarge | 8.4 | 25  
i3en.3xlarge | 12.5 | 25  

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  

214
<table>
<thead>
<tr>
<th>Instance Size</th>
<th>100% Random Read IOPS</th>
<th>Write IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>i3.xlarge *</td>
<td>206,250</td>
<td>70,000</td>
</tr>
<tr>
<td>i3.2xlarge</td>
<td>412,500</td>
<td>180,000</td>
</tr>
<tr>
<td>i3.4xlarge</td>
<td>825,000</td>
<td>360,000</td>
</tr>
<tr>
<td>i3.8xlarge</td>
<td>1.65 million</td>
<td>720,000</td>
</tr>
<tr>
<td>i3.16xlarge</td>
<td>3.3 million</td>
<td>1.4 million</td>
</tr>
<tr>
<td>i3.metal</td>
<td>3.3 million</td>
<td>1.4 million</td>
</tr>
<tr>
<td>i3en.large *</td>
<td>42,500</td>
<td>32,500</td>
</tr>
<tr>
<td>i3en.xlarge *</td>
<td>85,000</td>
<td>65,000</td>
</tr>
<tr>
<td>i3en.2xlarge *</td>
<td>170,000</td>
<td>130,000</td>
</tr>
<tr>
<td>i3en.3xlarge</td>
<td>250,000</td>
<td>200,000</td>
</tr>
<tr>
<td>i3en.6xlarge</td>
<td>500,000</td>
<td>400,000</td>
</tr>
<tr>
<td>i3en.12xlarge</td>
<td>1 million</td>
<td>800,000</td>
</tr>
<tr>
<td>i3en.24xlarge</td>
<td>2 million</td>
<td>1.6 million</td>
</tr>
<tr>
<td>i3en.metal</td>
<td>2 million</td>
<td>1.6 million</td>
</tr>
</tbody>
</table>

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

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

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

For instance store volumes that support TRIM, you can use the TRIM command to notify the SSD controller whenever you no longer need data that you've written. This provides the controller with more free space, which can reduce write amplification and increase performance. For more information, see Instance store volume TRIM support (p. 1404).

**Instance features**

The following is a summary of features for storage optimized instances:
<table>
<thead>
<tr>
<th>EBS only</th>
<th>Instance store</th>
<th>Placement group</th>
</tr>
</thead>
<tbody>
<tr>
<td>D2</td>
<td>No</td>
<td>HDD</td>
</tr>
<tr>
<td>D3</td>
<td>No</td>
<td>HDD *</td>
</tr>
<tr>
<td>D3en</td>
<td>No</td>
<td>HDD *</td>
</tr>
<tr>
<td>H1</td>
<td>No</td>
<td>HDD *</td>
</tr>
<tr>
<td>I3</td>
<td>No</td>
<td>NVMe *</td>
</tr>
<tr>
<td>I3en</td>
<td>No</td>
<td>NVMe *</td>
</tr>
</tbody>
</table>

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

For more information, see the following:

- Amazon EBS and NVMe on Windows instances (p. 1343)
- Amazon EC2 instance store (p. 1392)
- Placement groups (p. 975)

Release notes

- You must launch storage optimized instances using an HVM AMI.
- Instances built on the Nitro System (p. 146) have the following requirements:
  - NVMe drivers (p. 1343) must be installed
  - Elastic Network Adapter (ENA) drivers (p. 961) must be installed

The current AWS Windows AMIs (p. 27) meet these requirements.

- Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
- To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support.
- Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
- The d3.8xlarge and d3en.12xlarge instances support a maximum of three attachments, including the root volume. If you exceed the attachment limit when you add a network interface or EBS volume, this causes attachment issues on your 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? in the Amazon EC2 FAQ.

Windows accelerated computing instances

Accelerated computing instances use hardware accelerators, or co-processors, to perform some functions, such as floating point number calculations, graphics processing, or data pattern matching, more efficiently than is possible in software running on CPUs. These instances enable more parallelism for higher throughput on compute-intensive workloads.
If you require high processing capability, you’ll benefit from using accelerated computing instances, which provide access to hardware-based compute accelerators such as Graphics Processing Units (GPUs).

Contents
- GPU instances (p. 217)
- Hardware specifications (p. 218)
- Instance performance (p. 219)
- Network performance (p. 219)
- Instance features (p. 220)
- Release notes (p. 221)
- Install NVIDIA drivers on Windows instances (p. 222)
- Install AMD drivers on Windows instances (p. 227)
- Activate NVIDIA GRID Virtual Applications (p. 229)
- Optimize GPU settings (p. 229)

GPU instances

GPU-based instances provide access to NVIDIA GPUs with thousands of compute cores. You can use these instances to accelerate scientific, engineering, and rendering applications by leveraging the CUDA or Open Computing Language (OpenCL) parallel computing frameworks. You can also use them for graphics applications, including game streaming, 3-D application streaming, and other graphics workloads.

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. 793).

G4ad and G4dn instances

G4ad instances use AMD Radeon Pro V520 GPUs and 2nd generation AMD EPYC processors, and are well-suited for graphics applications such as remote graphics workstations, game streaming, and rendering that leverage industry-standard APIs such as OpenGL, DirectX, and Vulkan. They provide up to 4 AMD Radeon Pro V520 GPUs, 64 vCPUs, 25 Gbps networking, and 2.4 TB local NVMe-based SSD storage.

G4dn instances use NVIDIA Tesla GPUs and provide a cost-effective, high-performance platform for general purpose GPU computing using the CUDA or machine learning frameworks along with graphics applications using DirectX or OpenGL. These instances provide high-bandwidth networking, powerful half and single-precision floating-point capabilities, along with INT8 and INT4 precisions. Each GPU has 16 GiB of GDDR6 memory, making G4dn instances well-suited for machine learning inference, video transcoding, and graphics applications like remote graphics workstations and game streaming in the cloud.

For more information, see Amazon EC2 G4 Instances.

G4dn instances support NVIDIA GRID Virtual Workstation. For more information, see NVIDIA Marketplace offerings.

G3 instances

These instances use NVIDIA Tesla M60 GPUs and provide a cost-effective, high-performance platform for graphics applications using DirectX or OpenGL. G3 instances also provide NVIDIA GRID Virtual Workstation features, such as support for four monitors with resolutions up to 4096x2160, and NVIDIA GRID Virtual Applications. G3 instances are well-suited for applications such as 3D visualizations, graphics-intensive remote workstations, 3D rendering, video encoding, virtual reality, and other server-side graphics workloads requiring massively parallel processing power.
For more information, see Amazon EC2 G3 Instances.

G3 instances support NVIDIA GRID Virtual Workstation and NVIDIA GRID Virtual Applications. To activate either of these features, see Activate NVIDIA GRID Virtual Applications (p. 229).

**G2 instances**

These instances use NVIDIA GRID K520 GPUs and provide a cost-effective, high-performance platform for graphics applications using DirectX or OpenGL. NVIDIA GRID GPUs also support NVIDIA’s fast capture and encode API operations. Example applications include video creation services, 3D visualizations, streaming graphics-intensive applications, and other server-side graphics workloads.

**P3 instances**

These 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 single and double-precision floating-point capabilities, and up to 32 GiB of memory per GPU, which makes them ideal for deep learning, computational fluid dynamics, computational finance, seismic analysis, molecular modeling, genomics, rendering, and other server-side GPU compute workloads. Tesla V100 GPUs do not support graphics mode.

For more information, see Amazon EC2 P3 Instances.

P3 instances support NVIDIA NVLink peer to peer transfers. For more information, see NVIDIA NVLink.

**P2 instances**

P2 instances use NVIDIA Tesla K80 GPUs and are designed for general purpose GPU computing using the CUDA or OpenCL programming models. P2 instances provide high-bandwidth networking, powerful single and double precision floating-point capabilities, and 12 GiB of memory per GPU, which makes them ideal for deep learning, graph databases, high-performance databases, computational fluid dynamics, computational finance, seismic analysis, molecular modeling, genomics, rendering, and other server-side GPU compute workloads.

P2 instances support NVIDIA GPUDirect peer to peer transfers. For more information, see NVIDIA GPUDirect.

**Hardware specifications**

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

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Memory (GiB)</th>
<th>Accelerators</th>
</tr>
</thead>
<tbody>
<tr>
<td>p2.xlarge</td>
<td>4</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>p2.8xlarge</td>
<td>32</td>
<td>488</td>
<td>8</td>
</tr>
<tr>
<td>p2.16xlarge</td>
<td>64</td>
<td>732</td>
<td>16</td>
</tr>
<tr>
<td>p3.2xlarge</td>
<td>8</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>p3.8xlarge</td>
<td>32</td>
<td>244</td>
<td>4</td>
</tr>
<tr>
<td>p3.16xlarge</td>
<td>64</td>
<td>488</td>
<td>8</td>
</tr>
<tr>
<td>p3dn.24xlarge</td>
<td>96</td>
<td>768</td>
<td>8</td>
</tr>
<tr>
<td>g2.2xlarge</td>
<td>8</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>Instance type</td>
<td>Default vCPUs</td>
<td>Memory (GiB)</td>
<td>Accelerators</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>g2.8xlarge</td>
<td>32</td>
<td>60</td>
<td>4</td>
</tr>
<tr>
<td>g3s.xlarge</td>
<td>4</td>
<td>30.5</td>
<td>1</td>
</tr>
<tr>
<td>g3.4xlarge</td>
<td>16</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>g3.8xlarge</td>
<td>32</td>
<td>244</td>
<td>2</td>
</tr>
<tr>
<td>g3.16xlarge</td>
<td>64</td>
<td>488</td>
<td>4</td>
</tr>
<tr>
<td>g4ad.xlarge</td>
<td>4</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>g4ad.2xlarge</td>
<td>8</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td>g4ad.4xlarge</td>
<td>16</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td>g4ad.8xlarge</td>
<td>32</td>
<td>128</td>
<td>2</td>
</tr>
<tr>
<td>g4ad.16xlarge</td>
<td>64</td>
<td>256</td>
<td>4</td>
</tr>
<tr>
<td>g4dn.xlarge</td>
<td>4</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>g4dn.2xlarge</td>
<td>8</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td>g4dn.4xlarge</td>
<td>16</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td>g4dn.8xlarge</td>
<td>32</td>
<td>128</td>
<td>1</td>
</tr>
<tr>
<td>g4dn.12xlarge</td>
<td>48</td>
<td>192</td>
<td>4</td>
</tr>
<tr>
<td>g4dn.16xlarge</td>
<td>64</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>g4dn.metal</td>
<td>96</td>
<td>384</td>
<td>8</td>
</tr>
<tr>
<td>f1.2xlarge</td>
<td>8</td>
<td>122</td>
<td>1</td>
</tr>
<tr>
<td>f1.4xlarge</td>
<td>16</td>
<td>244</td>
<td>2</td>
</tr>
<tr>
<td>f1.16xlarge</td>
<td>64</td>
<td>976</td>
<td>8</td>
</tr>
</tbody>
</table>

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 Optimize CPU options (p. 540).

**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. 1344).

**Network performance**

You can enable enhanced networking on supported instance types to provide lower latencies, lower network jitter, and higher packet-per-second (PPS) performance. Most applications do not consistently need a high level of network performance, but can benefit from access to increased bandwidth when they send or receive data. For more information, see Enhanced networking on Windows (p. 960).
The following is a summary of network performance for accelerated computing instances that support enhanced networking.

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Network performance</th>
<th>Enhanced networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1.4xlarge and smaller</td>
<td>Up to 10 Gbps †</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>g3.4xlarge</td>
<td>10 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>g4ad.4xlarge</td>
<td>10 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>p3.2xlarge</td>
<td>10 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>g3.8xlarge</td>
<td>15 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>p2.8xlarge</td>
<td>15 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>p3.8xlarge</td>
<td>15 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>g4ad.8xlarge</td>
<td>25 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>g4dn.4xlarge and smaller</td>
<td>Up to 25 Gbps †</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>g4ad.16xlarge</td>
<td>50 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>g4dn.8xlarge</td>
<td>50 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>g4dn.12xlarge</td>
<td>50 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>g4dn.16xlarge</td>
<td>50 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
<tr>
<td>g4dn.metal</td>
<td>100 Gbps</td>
<td>ENA (p. 961)</td>
</tr>
</tbody>
</table>

† These instances have a baseline bandwidth and can use a network I/O credit mechanism to burst beyond their baseline bandwidth on a best effort basis. For more information, see instance network bandwidth (p. 959).

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Baseline bandwidth (Gbps)</th>
<th>Burst bandwidth (Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>g4ad.xlarge</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>g4ad.2xlarge</td>
<td>4.167</td>
<td>10</td>
</tr>
<tr>
<td>g4ad.4xlarge</td>
<td>8.333</td>
<td>10</td>
</tr>
<tr>
<td>g4dn.xlarge</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>g4dn.2xlarge</td>
<td>10</td>
<td>25</td>
</tr>
<tr>
<td>g4dn.4xlarge</td>
<td>20</td>
<td>25</td>
</tr>
</tbody>
</table>

**Instance features**

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

<table>
<thead>
<tr>
<th>EBS only</th>
<th>NVMe EBS</th>
<th>Instance store</th>
<th>Placement group</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>No</td>
<td>No</td>
<td>NVM *</td>
</tr>
<tr>
<td>G2</td>
<td>No</td>
<td>No</td>
<td>SSD</td>
</tr>
<tr>
<td>G3</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

220
The root device volume must be an Amazon EBS volume.

For more information, see the following:

- Amazon EBS and NVMe on Windows instances (p. 1343)
- Amazon EC2 instance store (p. 1392)
- Placement groups (p. 975)

Release notes

- You must launch the instance using an HVM AMI.
- Instances built on the Nitro System (p. 146) have the following requirements:
  - NVMe drivers (p. 1343) must be installed
  - Elastic Network Adapter (ENA) drivers (p. 961) must be installed

The current AWS Windows AMIs (p. 27) meet these requirements.
- GPU-based instances can't access the GPU unless the NVIDIA drivers are installed. For more information, see Install NVIDIA drivers on Windows instances (p. 222).
- Launching a bare metal instance boots the underlying server, which includes verifying all hardware and firmware components. This means that it can take 20 minutes from the time the instance enters the running state until it becomes available over the network.
- To attach or detach EBS volumes or secondary network interfaces from a bare metal instance requires PCIe native hotplug support.
- Bare metal instances use a PCI-based serial device rather than an I/O port-based serial device. The upstream Linux kernel and the latest Amazon Linux AMIs support this device. Bare metal instances also provide an ACPI SPCR table to enable the system to automatically use the PCI-based serial device. The latest Windows AMIs automatically use the PCI-based serial device.
- There is a limit of 100 AFIs per Region.
- There is a limit on the total number of instances that you can launch in a Region, and there are additional limits on some instance types. For more information, see How many instances can I run in Amazon EC2? in the Amazon EC2 FAQ.
- 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. We recommend that you use a different remote access tool to access your GPU, 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.

Install NVIDIA drivers on Windows instances

An instance with an attached NVIDIA GPU, such as a P3 or G4dn instance, must have the appropriate NVIDIA driver installed. Depending on the instance type, you can either download a public NVIDIA driver, download a driver from Amazon S3 that is available only to AWS customers, or use an AMI with the driver pre-installed.

To install AMD drivers on an instance with an attached AMD GPU, such as a G4ad instance, see Install AMD drivers on Windows instances (p. 227) instead.

Contents
• Types of NVIDIA drivers (p. 222)
• Available drivers by instance type (p. 223)
• Installation options (p. 223)
  • Option 1: AMIs with the NVIDIA drivers installed (p. 223)
  • Option 2: Public NVIDIA drivers (p. 224)
  • Option 3: GRID drivers (G3 and G4dn instances) (p. 224)
  • Option 4: NVIDIA gaming drivers (G4dn instances) (p. 225)
  • Install an additional version of CUDA (p. 227)

Types of NVIDIA drivers

The following are the main types of NVIDIA drivers that can be used with GPU-based instances.

Tesla drivers

These drivers are intended primarily for compute workloads, which use GPUs for computational tasks such as parallelized floating-point calculations for machine learning and fast Fourier transforms for high performance computing applications.

GRID drivers

These drivers are certified to provide optimal performance for professional visualization applications that render content such as 3D models or high-resolution videos. You can configure GRID drivers to support two modes. Quadro Virtual Workstations provide access to four 4K displays per GPU. GRID vApps provide RDSH App hosting capabilities.

Gaming drivers

These drivers contain optimizations for gaming and are updated frequently to provide performance enhancements. They support a single 4K display per GPU.

Configured mode

On Windows, the Tesla drivers are configured to run in Tesla Compute Cluster (TCC) mode. The GRID and gaming drivers are configured to run in Windows Display Driver Model (WDDM) mode. In TCC mode, the card is dedicated to compute workloads. In WDDM mode, the card supports both compute and graphics workloads.

NVIDIA control panel

The NVIDIA control panel is supported with GRID and Gaming drivers. It is not supported with Tesla drivers.
Supported APIs for Tesla drivers

- OpenCL
- NVIDIA CUDA and related libraries (for example, cuDNN, TensorRT, nvJPEG, and cuBLAS)
- NVENC for video encoding and NVDEC for video decoding

Supported APIs for GRID and gaming drivers

- DirectX, Direct2D, DirectX Video Acceleration, DirectX Raytracing
- OpenCL, OpenGL, and Vulkan
- NVIDIA CUDA and related libraries (for example, cuDNN, TensorRT, nvJPEG, and cuBLAS)
- NVENC for video encoding and NVDEC for video decoding

Available drivers by instance type

The following table summarizes the supported NVIDIA drivers for each GPU instance type.

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Tesla driver</th>
<th>GRID driver</th>
<th>Gaming driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>G2</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>G3</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>G4dn</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>P2</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>P3</td>
<td>Yes</td>
<td>Yes †</td>
<td>No</td>
</tr>
</tbody>
</table>

† Using Marketplace AMIs only

Installation options

Use one of the following options to get the NVIDIA drivers required for your GPU instance.

Options

- Option 1: AMIs with the NVIDIA drivers installed (p. 223)
- Option 2: Public NVIDIA drivers (p. 224)
- Option 3: GRID drivers (G3 and G4dn instances) (p. 224)
- Option 4: NVIDIA gaming drivers (G4dn instances) (p. 225)

Option 1: AMIs with the NVIDIA drivers installed

AWS and NVIDIA offer different Amazon Machine Images (AMI) that come with the NVIDIA drivers installed.

- Marketplace offerings with the Tesla driver
- Marketplace offerings with the GRID driver
- Marketplace offerings with the Gaming driver

If you create a custom Windows AMI using one of the AWS Marketplace offerings, the AMI must be a standardized image created using Sysprep (p. 40) to ensure that the GRID driver works.
Option 2: Public NVIDIA drivers

The options offered by AWS come with the necessary license for the driver. Alternatively, you can install the public drivers and bring your own license. To install a public driver, download it from the NVIDIA site as described here.

Alternatively, you can use the options offered by AWS instead of the public drivers. To use a GRID driver on a P3 instance, use the AWS Marketplace AMIs as described in Option 1 (p. 223). To use a GRID driver on a G3 or G4dn instance, use the AWS Marketplace AMIs, as described in Option 1 or install the NVIDIA drivers provided by AWS as described in Option 3 (p. 224).

To download a public NVIDIA driver

Log on to your Windows instance and download the 64-bit NVIDIA driver appropriate for the instance type from http://www.nvidia.com/Download/Find.aspx. For Product Type, Product Series, and Product, use the options in the following table.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Product Type</th>
<th>Product Series</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>G2</td>
<td>GRID</td>
<td>GRID Series</td>
<td>GRID K520</td>
</tr>
<tr>
<td>G3</td>
<td>Tesla</td>
<td>M-Class</td>
<td>M60</td>
</tr>
<tr>
<td>G4dn †</td>
<td>Tesla</td>
<td>T-Series</td>
<td>T4</td>
</tr>
<tr>
<td>P2</td>
<td>Tesla</td>
<td>K-Series</td>
<td>K80</td>
</tr>
<tr>
<td>P3</td>
<td>Tesla</td>
<td>V-Series</td>
<td>V100</td>
</tr>
</tbody>
</table>

† G4dn instances require driver version 426.00 or later.

To install the NVIDIA driver on Windows

1. 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.
2. Disable the built-in display adapter using Device Manager. Install these Windows features: Media Foundation and Quality Windows Audio Video Experience.
3. Check Device Manager to verify that the GPU is working correctly.
4. To achieve the best performance from your GPU, complete the optimization steps in Optimize GPU settings (p. 229).

Option 3: GRID drivers (G3 and G4dn instances)

These downloads are available to AWS customers only. By downloading, you agree to use the downloaded software only to develop AMIs for use with the NVIDIA Tesla T4 or NVIDIA Tesla M60 hardware. Upon installation of the software, you are bound by the terms of the NVIDIA GRID Cloud End User License Agreement.

Prerequisites

- If you launch your Windows instance using a custom Windows AMI, the AMI must be a standardized image created using Sysprep (p. 40) to ensure that the GRID driver works.
- IAM users must have the permissions granted by the AmazonS3ReadOnlyAccess policy.
To install the NVIDIA GRID driver on your Windows instance

1. Connect to your Windows instance and open a PowerShell window.
2. Download the drivers and the NVIDIA GRID Cloud End User License Agreement from Amazon S3 to your desktop using the following PowerShell commands.

```powershell
$Bucket = "ec2-windows-nvidia-drivers"
$KeyPrefix = "latest"
$LocalPath = "$home\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.

Starting with GRID version 11.0, you can use the drivers under `latest` for both G3 and G4dn instances. We will not add versions later than 11.0 to `g4/latest`, but will keep version 11.0 and the earlier versions specific to G4dn under `g4/latest`.

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. (Optional) Use the following command to 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.

```powershell
New-ItemProperty -Path "HKLM:\SOFTWARE\NVIDIA Corporation\Global\GridLicensing" -Name "NvCplDisableManageLicensePage" -PropertyType "DWord" -Value "1"
```

5. (Optional) Depending on your use case, you might complete the following optional steps. If you do not require this functionality, do not complete these steps.

   a. To help take advantage of the four displays of up to 4K resolution, set up the high-performance display protocol, NICE DCV.
   b. NVIDIA Quadro Virtual Workstation mode is enabled by default. To activate GRID Virtual Applications for RDSH Application hosting capabilities, complete the GRID Virtual Application activation steps in Activate NVIDIA GRID Virtual Applications (p. 229).

Option 4: NVIDIA gaming drivers (G4dn instances)

These drivers are available to AWS customers only. By downloading them, you agree to use the downloaded software only to develop AMIs for use with the NVIDIA Tesla T4 hardware. Upon installation of the software, you are bound by the terms of the NVIDIA GRID Cloud End User License Agreement.

Prerequisites

- If you launch your Windows instance using a custom Windows AMI, the AMI must be a standardized image created using Sysprep (p. 40) to ensure that the gaming driver works.
• IAM users must have the permissions granted by the AmazonS3ReadOnlyAccess policy.

To install the NVIDIA gaming driver on your Windows instance

1. Connect to your Windows instance and open a PowerShell window.
2. Download and install the gaming driver using the following PowerShell commands.

```powershell
$Bucket = "nvidia-gaming"
$KeyPrefix = "windows/latest"
$LocalPath = "$home\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 S3 bucket. You can download all of the available versions in the bucket by removing the -KeyPrefix $KeyPrefix option.

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. Create a registry value in the HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global key with the name vGamingMarketplace, the type DWord, and the value 2. You can use either the Command Prompt window or a 64-bit version of PowerShell as follows.

- Use the following PowerShell command to create this registry value. By default, the AWS Tools for PowerShell in AWS Windows AMIs is a 32-bit version and this command fails. Instead, use the 64-bit version of PowerShell included with the operating system.

```powershell
New-ItemProperty -Path "HKLM:\SOFTWARE\NVIDIA Corporation\Global" -Name "vGamingMarketplace" -PropertyType "DWord" -Value "2"
```

- Use the following registry command to create this registry value. You can run it using the Command Prompt window or a 64-bit version of PowerShell.

```cmd
reg add "HKLM\SOFTWARE\NVIDIA Corporation\Global" /v vGamingMarketplace /t REG_DWORD /d 2
```

5. Use the following command to download the certification file, rename the file GridSwCert.txt, and move the file to the Public Documents folder on your system drive. Typically, the folder path is C:\Users\Public\Public Documents (Windows Explorer) or C:\Users\Public\Documents (Command Prompt window).

- For version 461.40 or later:

```powershell
Invoke-WebRequest -Uri "https://nvidia-gaming.s3.amazonaws.com/GridSwCert-Archive/GridSwCertWindows_2021_10_2.cert" -OutFile "$Env:PUBLIC\Documents\GridSwCert.txt"
```

- For version 445.87:

```powershell
```

- For earlier versions:

6. Reboot your instance.
7. Verify the NVIDIA Gaming license using the following command.

"C:\Program Files\NVIDIA Corporation\NVSMI\nvidia-smi.exe" -q

The output should be similar to the following.

<table>
<thead>
<tr>
<th>GRID Licensed Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Name</td>
</tr>
<tr>
<td>License Status</td>
</tr>
</tbody>
</table>

8. (Optional) To help take advantage of the single display of up to 4K resolution, set up the high-performance display protocol **NICE DCV**. If you do not require this functionality, do not complete this step.

### Install an additional version of CUDA

After you install an NVIDIA graphics driver on your instance, you can install a version of CUDA other than the version that is bundled with the graphics driver. The following procedure demonstrates how to configure multiple versions of CUDA on the instance.

**To install the CUDA toolkit**

1. Connect to your Windows instance.
2. Open the [NVIDIA website](https://www.nvidia.com/) and select the version of CUDA that you need.
3. For **Installer Type**, select **exe (local)** and then choose **Download**.
4. Using your browser, run the downloaded install file. Follow the instructions to install the CUDA toolkit. You might be required to reboot the instance.

### Install AMD drivers on Windows instances

An instance with an attached AMD GPU, such as a G4ad instance, must have the appropriate AMD driver installed. Depending on your requirements, you can either use an AMI with the driver preinstalled or download a driver from Amazon S3.

To install NVIDIA drivers on an instance with an attached NVIDIA GPU, such as a G4dn instance, see [Install NVIDIA drivers on Windows instances](#) instead.

**Contents**

- [AMD Radeon Pro Software for Enterprise Driver](#)
- AMIs with the AMD driver installed
- AMD driver download

**AMD Radeon Pro Software for Enterprise Driver**

The AMD Radeon Pro Software for Enterprise Driver is built to deliver support for professional-grade graphics use cases. Using the driver, you can configure your instances with two 4K displays per GPU.
Supported APIs

• OpenGL, OpenCL
• Vulkan
• DirectX 9 and later
• AMD Advanced Media Framework
• Microsoft Hardware Media Foundation Transform

AMIs with the AMD driver installed

AWS offers different Amazon Machine Images (AMI) that come with the AMD drivers installed. Open Marketplace offerings with the AMD driver.

AMD driver download

If you aren’t using an AMI with the AMD driver installed, you can download the AMD driver and install it on your instance.

These downloads are available to AWS customers only. By downloading, you agree to use the downloaded software only to develop AMIs for use with the AMD Radeon Pro V520 hardware. Upon installation of the software, you are bound by the terms of the AMD Software End User License Agreement.

Prerequisites


• IAM users must have the permissions granted by the AmazonS3ReadOnlyAccess policy.

To install the AMD driver on your Windows instance

1. Connect to your Windows instance and open a PowerShell window.

2. Download the drivers from Amazon S3 to your desktop using the following PowerShell commands.

```
$Bucket = "ec2-amd-windows-drivers"
$KeyPrefix = "latest"
$LocalPath = "$home\Desktop\AMD"
$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
    }
}
```

3. Unzip the downloaded driver file and run the installer using the following PowerShell commands.

```
Expand-Archive $LocalFilePath -DestinationPath $home\Desktop -Verbose
$Driverdir = Get-ChildItem $home\Desktop\ -Directory -Filter "*WHQL*"
Write-Host $Driverdir
pnputil /add-driver $home\Desktop\$Driverdir\Drivers\Display\WT6A_INF\*inf /install
```

4. Follow the instructions to install the driver and reboot your instance as required.
5. To verify that the GPU is working properly, check Device Manager. You should see "AMD Radeon Pro V520 MxGPU" listed as a display adapter.

6. To help take advantage of the four displays of up to 4K resolution, set up the high-performance display protocol, NICE DCV.

**Activate NVIDIA GRID Virtual Applications**

To activate the GRID Virtual Applications on G3 and G4dn 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 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.

**Optimize GPU settings**

There are several GPU setting optimizations that you can perform to achieve the best performance on G3, G4dn, P2, P3, and P3dn instances. With some of these instance types, the NVIDIA driver uses an autoboot feature, which varies the GPU clock speeds. By disabling autoboot 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.
   ```powershell
cd "C:\Program Files\NVIDIA Corporation\NVSMI"
   ```
2. G2, G3, and P2 instances: Disable the autoboot feature for all GPUs on the instance.
   ```
   .\nvidia-smi --auto-boost-default=0
   ```
3. Set all GPU clock speeds to their maximum frequency. Use the memory and graphics clock speeds specified in the following commands.

Some versions of the NVIDIA driver do not support setting the application clock speed, and display the error "Setting applications clocks is not supported for GPU...", which you can ignore.

- G3 instances:
Find an Amazon EC2 instance type

Before you can launch an instance, you must select an instance type to use. The instance type that you choose might depend on your requirements for the instances that you'll launch. For example, you might choose an instance type based on the following requirements:

- Availability Zone or Region
- Compute
- Memory
- Networking
- Pricing
- Storage

Find an instance type using the console

You can find an instance type that meets your needs using the Amazon EC2 console.

To find an instance type using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region in which to launch your instances. You can select any Region that's available to you, regardless of your location.
3. In the navigation pane, choose Instance Types.
4. (Optional) Choose the preferences (gear) icon to select which instance type attributes to display, such as On-Demand Linux pricing, and then choose Confirm. Alternatively, select an instance type and view all attributes using the Details pane.
5. Use the instance type attributes to filter the list of displayed instance types to only the instance types that meet your needs. For example, you can list all instance types that have more than eight vCPUs and also support hibernation.
6. (Optional) Select multiple instance types to see a side-by-side comparison across all attributes in the Details pane.
7. (Optional) To save the list of instance types to a comma-separated values (.csv) file for further review, choose Download list CSV. The file includes all instance types that match the filters you set.
8. After locating instance types that meet your needs, you can use them to launch instances. For more information, see Launch an instance using the Launch Instance Wizard (p. 392).
Find an instance type using the AWS CLI

You can use AWS CLI commands for Amazon EC2 to find an instance type that meet your needs.

To find an instance type using the AWS CLI

1. If you have not done so already, install the AWS CLI For more information, see the AWS Command Line Interface User Guide.
2. Use the `describe-instance-types` command to filter instance types based on instance attributes. For example, you can use the following command to display only instance types with 48 vCPUs.

   ```bash
   aws ec2 describe-instance-types --filters "Name=vcpu-info.default-vcpus,Values=48"
   ```

3. Use the `describe-instance-type-offerings` command to filter instance types offered by location (Region or Availability Zone). For example, you can use the following command to display the instance types offered in the specified Availability Zone.

   ```bash
   aws ec2 describe-instance-type-offerings --location-type "availability-zone" --filters Name=location,Values=us-east-2a --region us-east-2
   ```

4. After locating instance types that meet your needs, make note of them so that you can use these instance types when you launch instances. For more information, see Launching your instance in the AWS Command Line Interface User Guide.

Change 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 resize your instance by changing its instance type. For example, if your t2.micro instance is too small for its workload, you can change it to another instance type that is appropriate for the workload.

You might also want to migrate from a previous generation instance type to a current generation instance type to take advantage of some features; for example, support for IPv6.

If you want a recommendation for an instance type that is best able to handle your existing workload, you can use AWS Compute Optimizer. For more information, see Get recommendations for an instance type (p. 236).

Contents
- Requirements for changing the instance type (p. 231)
- Compatibility for resizing instances (p. 232)
- Resize an Amazon EBS–backed instance (p. 232)
- Migrate to a new instance configuration (p. 235)

Requirements for changing the instance type

To resize your Amazon EC2 instance by changing its instance type, consider the following requirements:

- 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.
- To change the instance type, the instance must be in the stopped state.
- You cannot resize an instance if hibernation is enabled.
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:
  - If you are resizing an instance type with a processor based on the Arm architecture, you are limited to the instance types that support a processor based on the Arm architecture, such as C6g and M6g.
  - The following instance types are the only instance types that support 32-bit AMIs: t2.nano, t2.micro, t2.small, t2.medium, c3.large, t1.micro, m1.small, m1.medium, and c1.medium. If you are resizing a 32-bit instance, you are limited to these instance types.

- **Network**: Newer instance types must be launched in a VPC. Therefore, you can't resize an instance in the EC2-Classic platform to a instance type that is available only in a VPC unless you have a nondefault VPC. To check whether your instance is in a VPC, check the VPC ID value on the details pane of the Instances screen in the Amazon EC2 console. For more information, see Migrate from EC2-Classic to a VPC (p. 1041).

- **Network adapters**: If you switch from a driver for one network adapter to another, the network adapter settings are reset when the operating system creates the new adapter. To reconfigure the settings, you might need access to a local account with administrator permissions. The following are examples of moving from one network adapter to another:
  - AWS PV (T2 instances) to Intel 82599 VF (M4 instances)
  - Intel 82599 VF (most M4 instances) to ENA (M5 instances)
  - ENA (M5 instances) to high-bandwidth ENA (M5n instances)

- **Enhanced networking**: Instance types that support enhanced networking (p. 960) require the necessary drivers installed. For example, instances based on the Nitro System (p. 146) require EBS-backed AMIs with the Elastic Network Adapter (ENA) drivers installed. To resize an instance from a type that does not support enhanced networking to a type that supports enhanced networking, you must install the ENA drivers (p. 961) or ixgbevf drivers (p. 968) on the instance, as appropriate.

- **NVMe**: EBS volumes are exposed as NVMe block devices on instances built on the Nitro System (p. 146). 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. 1343) 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. 150)
  - Compute optimized instances (p. 193)
  - Memory optimized instances (p. 199)
  - Storage optimized instances (p. 210)

Resize 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.
- When you resize an instance, the resized instance usually has the same number of instance store volumes that you specified when you launched the original instance. With instance types that support
NVMe instance store volumes (which are available by default), the resized instance might have additional instance store volumes, depending on the AMI. Otherwise, you can migrate your application to an instance with a new instance type manually, specifying the number of instance store volumes that you need when you launch the new instance.

- If your instance is in an Auto Scaling group, the Amazon EC2 Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. To prevent this, you can suspend the scaling processes for the group while you’re resizing your instance. For more information, see Suspending and Resuming Scaling Processes in the Amazon EC2 Auto Scaling User Guide.

- If your instance is in a cluster placement group (p. 976) and, after changing the instance type, the instance start fails, try the following: stop all the instances in the cluster placement group, change the instance type for the affected instance, and then restart all the instances in the cluster placement group.

- Ensure that you plan for downtime while your instance is stopped. Stopping and resizing an instance may take a few minutes, and restarting your instance may take a variable amount of time depending on your application's startup scripts.

For more information, see Stop and start your instance (p. 425).

Use the following procedure to resize an Amazon EBS–backed instance using the AWS Management Console.

New console

**To resize an Amazon EBS–backed instance**

1. (Optional) If the new instance type requires drivers that are not installed on the existing instance, you must connect to your instance and install the drivers first. For more information, see Compatibility for resizing instances (p. 232).

   **Note**
   
   The AWS PV driver package should be updated before changing instance families. For more information, see Upgrade PV drivers on Windows instances (p. 524).

2. (Optional) If you configured your Windows instance to use static IP addressing (p. 568) 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 and later] Connect to your Windows instance and run the following EC2Launch PowerShell script to configure the instance after it is resized.

   ```powershell
   PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -Schedule
   ```

5. In the navigation pane, choose Instances.

6. Select the instance and choose Actions, Instance state, Stop instance.

7. In the confirmation dialog box, choose Stop. It can take a few minutes for the instance to stop.

8. With the instance still selected, choose Actions, Instance settings, Change instance type. This action is grayed out 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). For more information, see Compatibility for resizing instances (p. 232).

b. (Optional) If the instance type that you selected supports EBS–optimization, select EBS-optimized to enable EBS–optimization or deselect EBS-optimized to disable EBS–optimization. If the instance type that you selected is EBS–optimized by default, EBS-optimized is selected and you can’t deselect it.

c. Choose Apply to accept the new settings.

10. To restart the stopped instance, select the instance and choose Instance state, Start instance. It can take a few minutes for the instance to enter the running state.

Old console

**To resize an Amazon EBS–backed instance**

1. (Optional) If the new instance type requires drivers that are not installed on the existing instance, you must connect to your instance and install the drivers first. For more information, see Compatibility for resizing instances (p. 232).

   **Note**
   The AWS PV driver package should be updated before changing instance families. For more information, see Upgrade PV drivers on Windows instances (p. 524).

2. (Optional) If you configured your Windows instance to use static IP addressing (p. 568) 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 and later] Connect to your Windows instance and run the following EC2Launch PowerShell script to configure the instance after it is resized.

   ```powershell
   PS C:\> C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -Schedule
   ```

5. In the navigation pane, choose Instances.

6. Select the instance and choose Actions, Instance State, 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, Instance Settings, Change Instance Type. This action is grayed out 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). For more information, see Compatibility for resizing instances (p. 232).

   b. (Optional) If the instance type that you selected supports EBS–optimization, select EBS-optimized to enable EBS–optimization or deselect EBS-optimized to disable EBS–optimization. If the instance type that you selected is EBS–optimized by default, EBS-optimized is selected and you can’t deselect it.
c. Choose Apply to accept the new settings.

10. To restart the stopped instance, select the instance and choose Actions, Instance State, Start.

11. In the confirmation dialog box, choose Yes, Start. It can take a few minutes for the instance to enter the running state.

Migrate 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.

New console

To migrate your application to a compatible instance

1. Back up any data on your instance store volumes that you need to keep to persistent storage. To migrate data on your EBS volumes that you need to keep, create a snapshot of the volumes (see Create Amazon EBS snapshots (p. 1211)) or detach the volume from the instance so that you can attach it to the new instance later (see Detach an Amazon EBS volume from a Windows instance (p. 1204)).

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 and choose Actions, Disassociate Elastic IP address. When prompted for confirmation, choose Disassociate.
   c. With the Elastic IP address still selected, choose Actions, Associate Elastic IP address.
   d. For Resource type, choose Instance.
   e. For Instance, choose the instance with which to associate the Elastic IP address. You can also enter text to search for a specific instance.
   f. (Optional) For Private IP address, specify a private IP address with which to associate the Elastic IP address.
   g. Choose Associate.

6. (Optional) You can terminate the original instance if it's no longer needed. Select the instance and verify that you are about to terminate the original instance, not the new instance (for example, check the name or launch time). Choose Instance state, Terminate instance.
Old console

To migrate your application to a compatible instance

1. Back up any data on your instance store volumes that you need to keep to persistent storage. To migrate data on your EBS volumes that you need to keep, create a snapshot of the volumes (see Create Amazon EBS snapshots (p. 1211)) or detach the volume from the instance so that you can attach it to the new instance later (see Detach an Amazon EBS volume from a Windows instance (p. 1204)).

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 and choose Actions, Disassociate address. When prompted for confirmation, choose Disassociate address.
   c. With the Elastic IP address still selected, choose Actions, Associate address.
   d. From Instance, select the new instance, and then choose Associate.

6. (Optional) You can terminate the original instance if it’s no longer needed. Select the instance and verify that you are about to terminate the original instance, not the new instance (for example, check the name or launch time). Choose Actions, Instance State, Terminate.

Get recommendations for an instance type

AWS Compute Optimizer provides Amazon EC2 instance recommendations to help you improve performance, save money, or both. You can use these recommendations to decide whether to move to a new instance type.

To make recommendations, Compute Optimizer analyzes your existing instance specifications and utilization metrics. The compiled data is then used to recommend which Amazon EC2 instance types are best able to handle the existing workload. Recommendations are returned along with per-hour instance pricing.

This topic outlines how to view recommendations through the Amazon EC2 console. For more information, see the AWS Compute Optimizer User Guide.

Note
To get recommendations from Compute Optimizer, you must first opt in to Compute Optimizer. For more information, see Getting Started with AWS Compute Optimizer in the AWS Compute Optimizer User Guide.

Contents
- Limitations (p. 237)
- Findings (p. 237)
Limitations

Compute Optimizer currently generates recommendations for M, C, R, T, and X instance types. Other instance types are not considered by Compute Optimizer. If you're using other instance types, they will not be listed in the Compute Optimizer recommendations view. For information about these and other instance types, see Instance types (p. 141).

Findings

Compute Optimizer classifies its findings for EC2 instances as follows:

- **Under-provisioned** – An EC2 instance is considered under-provisioned when at least one specification of your instance, such as CPU, memory, or network, does not meet the performance requirements of your workload. Under-provisioned EC2 instances might lead to poor application performance.

- **Over-provisioned** – An EC2 instance is considered over-provisioned when at least one specification of your instance, such as CPU, memory, or network, can be sized down while still meeting the performance requirements of your workload, and when no specification is under-provisioned. Over-provisioned EC2 instances might lead to unnecessary infrastructure cost.

- **Optimized** – An EC2 instance is considered optimized when all specifications of your instance, such as CPU, memory, and network, meet the performance requirements of your workload, and the instance is not over-provisioned. An optimized EC2 instance runs your workloads with optimal performance and infrastructure cost. For optimized instances, Compute Optimizer might sometimes recommend a new generation instance type.

- **None** – There are no recommendations for this instance. This might occur if you've been opted in to Compute Optimizer for less than 12 hours, or when the instance has been running for less than 30 hours, or when the instance type is not supported by Compute Optimizer. For more information, see Limitations (p. 237) in the previous section.

View recommendations

After you opt in to Compute Optimizer, you can view the findings that Compute Optimizer generates for your EC2 instances in the EC2 console. You can then access the Compute Optimizer console to view the recommendations. If you recently opted in, findings might not be reflected in the EC2 console for up to 12 hours.

New console

**To view a recommendation for an EC2 instance through the EC2 console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and then choose the instance ID.
3. On the instance summary page, in the **AWS Compute Optimizer** banner near the bottom of the page, choose **View detail**.

The instance opens in Compute Optimizer, where it is labeled as the **Current** instance. Up to three different instance type recommendations, labeled **Option 1**, **Option 2**, and **Option 3**, are provided. The bottom half of the window shows recent CloudWatch metric data for the current instance: **CPU utilization**, **Memory utilization**, **Network in**, and **Network out**.

237
4. (Optional) In the Compute Optimizer console, choose the settings (⚙️) icon to change the visible columns in the table, or to view the public pricing information for a different purchasing option for the current and recommended instance types.

**Note**
If you’ve purchased a Reserved Instance, your On-Demand Instance might be billed as a Reserved Instance. Before you change your current instance type, first evaluate the impact on Reserved Instance utilization and coverage.

---

**Old console**

**To view a recommendation for an EC2 instance through the EC2 console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.

The instance opens in Compute Optimizer, where it is labeled as the Current instance. Up to three different instance type recommendations, labeled Option 1, Option 2, and Option 3, are provided. The bottom half of the window shows recent CloudWatch metric data for the current instance: CPU utilization, Memory utilization, Network in, and Network out.

4. (Optional) In the Compute Optimizer console, choose the settings (⚙️) icon to change the visible columns in the table, or to view the public pricing information for a different purchasing option for the current and recommended instance types.

**Note**
If you’ve purchased a Reserved Instance, your On-Demand Instance might be billed as a Reserved Instance. Before you change your current instance type, first evaluate the impact on Reserved Instance utilization and coverage.

Determine whether you want to use one of the recommendations. Decide whether to optimize for performance improvement, for cost reduction, or for a combination of the two. For more information, see Viewing Resource Recommendations in the AWS Compute Optimizer User Guide.

**To view recommendations for all EC2 instances across all Regions through the Compute Optimizer console**

2. Choose View recommendations for all EC2 instances.
3. You can perform the following actions on the recommendations page:
   a. To filter recommendations to one or more AWS Regions, enter the name of the Region in the Filter by one or more Regions text box, or choose one or more Regions in the drop-down list that appears.
   b. To view recommendations for resources in another account, choose Account, and then select a different account ID.

   This option is available only if you are signed in to a management account of an organization, and you opted in all member accounts within the organization.
   c. To clear the selected filters, choose Clear filters.
   d. To change the purchasing option that is displayed for the current and recommended instance types, choose the settings (⚙️) icon, and then choose On-Demand Instances, Reserved Instances, standard 1-year no upfront, or Reserved Instances, standard 3-year no upfront.
e. To view details, such as additional recommendations and a comparison of utilization metrics, choose the finding (Under-provisioned, Over-provisioned, or Optimized) listed next to the desired instance. For more information, see Viewing Resource Details in the AWS Compute Optimizer User Guide.

Considerations for evaluating recommendations

Before changing an instance type, consider the following:

• The recommendations don’t forecast your usage. Recommendations are based on your historical usage over the most recent 14-day time period. Be sure to choose an instance type that is expected to meet your future resource needs.

• Focus on the graphed metrics to determine whether actual usage is lower than instance capacity. You can also view metric data (average, peak, percentile) in CloudWatch to further evaluate your EC2 instance recommendations. For example, notice how CPU percentage metrics change during the day and whether there are peaks that need to be accommodated. For more information, see Viewing Available Metrics in the Amazon CloudWatch User Guide.

• Compute Optimizer might supply recommendations for burstable performance instances, which are T3, T3a, and T2 instances. If you periodically burst above the baseline, make sure that you can continue to do so based on the vCPUs of the new instance type. For more information, see Key concepts and definitions for burstable performance instances (p. 162).

• If you’ve purchased a Reserved Instance, your On-Demand Instance might be billed as a Reserved Instance. Before you change your current instance type, first evaluate the impact on Reserved Instance utilization and coverage.

• Consider conversions to newer generation instances, where possible.

• When migrating to a different instance family, make sure the current instance type and the new instance type are compatible, for example, in terms of virtualization, architecture, or network type. For more information, see Compatibility for resizing instances (p. 232).

• Finally, consider the performance risk rating that’s provided for each recommendation. Performance risk indicates the amount of effort you might need to spend in order to validate whether the recommended instance type meets the performance requirements of your workload. We also recommend rigorous load and performance testing before and after making any changes.

There are other considerations when resizing an EC2 instance. For more information, see Change the instance type (p. 231).

Additional resources

For more information:

• Instance types (p. 141)
• AWS Compute Optimizer User Guide

Instance purchasing options

Amazon EC2 provides the following purchasing options to enable you to optimize your costs based on your needs:

• On-Demand Instances – Pay, by the second, for the instances that you launch.
• Savings Plans – Reduce your Amazon EC2 costs by making a commitment to a consistent amount of usage, in USD per hour, for a term of 1 or 3 years.
Contents
- Determine the instance lifecycle (p. 240)
- On-Demand Instances (p. 241)
- Reserved Instances (p. 245)
- Scheduled Reserved Instances (p. 281)
- Spot Instances (p. 282)
- Dedicated Hosts (p. 329)
- Dedicated Instances (p. 360)
- On-Demand Capacity Reservations (p. 366)

Determine 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.

Use the following procedure to determine the lifecycle of an instance.

New console

To determine the instance lifecycle using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance.
4. On the Details tab, under Instance details, find Lifecycle. If the value is spot, the instance is a Spot Instance. If the value is normal, the instance is either an On-Demand Instance or a Reserved Instance.
5. On the Details tab, under Host and placement group, find Tenancy. If the value is host, the instance is running on a Dedicated Host. If the value is dedicated, the instance is a Dedicated Instance.
6. (Optional) If you have purchased a Reserved Instance and want to verify that it is being applied, you can check the usage reports for Amazon EC2. For more information, see Amazon EC2 usage reports (p. 1464).
Old console

To determine the instance lifecycle using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance.
4. On the Description tab, find Tenancy. If the value is host, the instance is running on a Dedicated Host. If the value is dedicated, the instance is a Dedicated Instance.
5. On the Description tab, find Lifecycle. If the value is spot, the instance is a Spot Instance. If the value is 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. 1464).

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"
```

Otherwise, the output does not contain InstanceLifecycle.

On-Demand Instances

With On-Demand Instances, you pay for compute capacity by the hour with no long-term commitments. You have full control over its lifecycle—you decide when to launch, stop, hibernate, start, reboot, or terminate it.

There is no long-term commitment required when you purchase On-Demand Instances. You pay only for the hours that your On-Demand Instances are in the running state. The price per hour for a running On-Demand Instance is fixed, and is listed on the Amazon EC2 Pricing, On-Demand Pricing page.

We recommend that you use On-Demand Instances for applications with short-term, irregular workloads that cannot be interrupted.

For significant savings over On-Demand Instances, use AWS Savings Plans, Spot Instances (p. 282), or Reserved Instances (p. 245).
• Work with On-Demand Instances (p. 242)
• On-Demand Instance limits (p. 242)
  • Calculate how many vCPUs you need (p. 243)
  • Request a limit increase (p. 244)
  • Monitor On-Demand Instance limits and usage (p. 244)
• Query the prices of On-Demand Instances (p. 244)

Work with On-Demand Instances

You can work with On-Demand Instances in the following ways:

• Launch your instance (p. 390)
• Connect to your Windows instance (p. 413)
• Stop and start your instance (p. 425)
• Hibernate your On-Demand or Reserved Windows instance (p. 428)
• Reboot your instance (p. 438)
• Instance retirement (p. 439)
• Terminate your instance (p. 441)
• Recover your instance (p. 447)
• Configure your Windows instance (p. 448)
• Identify EC2 Windows instances (p. 655)

If you're new to Amazon EC2, see How to get started with Amazon EC2 (p. 1).

On-Demand Instance limits

There is a limit on the number of running On-Demand Instances per AWS account per Region. On-Demand Instance limits are managed in terms of the number of virtual central processing units (vCPUs) that your running On-Demand Instances are using, regardless of the instance type. The following table lists the On-Demand Instance limits. Each limit specifies the default vCPUs for one or more instance families. For information about the different instance families, generations, and sizes, see Amazon EC2 Instance Types.

Note
New AWS accounts might start with limits that are lower than these defaults. Amazon EC2 monitors your usage and raises your limits automatically based on your usage.

<table>
<thead>
<tr>
<th>Limit</th>
<th>Default vCPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running On-Demand All Standard (A, C, D, H, I, M, R, T, Z) instances</td>
<td>1,152</td>
</tr>
<tr>
<td>Running On-Demand All F instances</td>
<td>128</td>
</tr>
<tr>
<td>Running On-Demand All G instances</td>
<td>128</td>
</tr>
<tr>
<td>Running On-Demand High Memory (u-*) instances</td>
<td>448</td>
</tr>
<tr>
<td>Running On-Demand All Inf instances</td>
<td>128</td>
</tr>
<tr>
<td>Running On-Demand All P instances</td>
<td>128</td>
</tr>
</tbody>
</table>
You can launch any combination of instance types that meet your changing application needs, as long as the number of vCPUs does not exceed your account limit. For example, with a Standard instance limit of 256 vCPUs, you could launch 32 m5.2xlarge instances (32 x 8 vCPUs) or 16 c5.4xlarge instances (16 x 16 vCPUs). For more information, see EC2 On-Demand Instance limits.

Calculate how many vCPUs you need

You can use the vCPU limits calculator to determine the number of vCPUs that you require for your application needs.

When using the calculator, keep the following in mind: The calculator assumes that you have reached your current limit. The value that you enter for Instance count is the number of instances that you need to launch in addition to what is permitted by your current limit. The calculator adds your current limit to the Instance count to arrive at a new limit.

The following screenshot shows the vCPU limits calculator.

You can view and use the following controls and information:

- **Instance type** – The instance types that you add to the vCPU limits calculator.
- **Instance count** – The number of instances that you require for the selected instance type.
- **vCPU count** – The number of vCPUs that corresponds to the Instance count.
- **Current limit** – Your current limit for the limit type to which the instance type belongs. The limit applies to all instance types of the same limit type. For example, in the preceding screenshot, the current limit for m5.2xlarge and c5.4xlarge is 1,920 vCPUs, which is the limit for all the instance types that belong to the All Standard instances limit.
- **New limit** – The new limit, in number of vCPUs, which is calculated by adding vCPU count and Current limit.
- **X** – Choose the X to remove the row.
- **Add instance type** – Choose Add instance type to add another instance type to the calculator.
• **Limits calculation** – Displays the current limit, vCPUs needed, and new limit for the limit types.
  • **Instance limit name** – The limit type for the instance types that you selected.
  • **Current limit** – The current limit for the limit type.
  • **vCPUs needed** – The number of vCPUs that corresponds to the number of instances that you specified in **Instance count**. For the All Standard instances limit type, the vCPUs needed is calculated by adding the values for **vCPU count** for all the instance types of this limit type.
  • **New limit** – The new limit is calculated by adding **Current limit** and **vCPUs needed**.
  • **Options** – Choose **Request limit increase** to request a limit increase for the corresponding limit type.

**To calculate the number of required vCPUs**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. From the navigation bar, select a Region.
3. From the left navigator, choose **Limits**.
4. Choose **Calculate vCPU limit**.
5. Choose **Add instance type**, choose the required instance type, and specify the required number of instances. To add more instance types, choose **Add instance type** again.
6. View **Limits calculation** for the required new limit.
7. When you've finished using the calculator, choose **Close**.

**Request a limit increase**

You can request a limit increase for each On-Demand Instance limit type from the **Limits page** or the vCPU limits calculator in the Amazon EC2 console. Complete the required fields on the AWS Support Center limit increase form with your use case. For **Primary Instance Type**, select the limit type that corresponds to the **Instance limit name** in the vCPU limits calculator. For the new limit value, use the value that appears in the **New limit** column in the vCPU limits calculator. For more information about requesting a limit increase, see **Amazon EC2 service quotas** (p. 1463).

**Monitor On-Demand Instance limits and usage**

You can view and manage your On-Demand Instance limits using the following:

• The **Limits page** in the Amazon EC2 console
• The Amazon EC2 Services quotas page in the Service Quotas console
• The `get-service-quota` AWS CLI
• The **Service Limits page** in the AWS Trusted Advisor console

For more information, see **Amazon EC2 service quotas** (p. 1463) in the **Amazon EC2 User Guide for Linux Instances**, **Viewing a Service Quota** in the **Service Quotas User Guide**, and **AWS Trusted Advisor**.

With Amazon CloudWatch metrics integration, you can monitor EC2 usage against limits. You can also configure alarms to warn about approaching limits. For more information, see **Using Amazon CloudWatch Alarms** in the **Service Quotas User Guide**.

**Query the prices of On-Demand Instances**

You can use the Price List Service API or the AWS Price List API to query the prices of On-Demand Instances. For more information, see **Using the AWS Price List API** in the **AWS Billing and Cost Management User Guide**.
Reserved Instances

Reserved Instances provide you with significant savings on your Amazon EC2 costs compared to On-Demand Instance pricing. Reserved Instances are not physical instances, but rather a billing discount applied to the use of On-Demand Instances in your account. These On-Demand Instances must match certain attributes, such as instance type and Region, in order to benefit from the billing discount.

Savings Plans also offer significant savings on your Amazon EC2 costs compared to On-Demand Instance pricing. With Savings Plans, you make a commitment to a consistent usage amount, measured in USD per hour. This provides you with the flexibility to use the instance configurations that best meet your needs and continue to save money, instead of making a commitment to a specific instance configuration. For more information, see the AWS Savings Plans User Guide.

Reserved Instances topics
- Reserved Instance overview (p. 245)
- Key variables that determine Reserved Instance pricing (p. 246)
- Reserved Instance limits (p. 247)
- Regional and zonal Reserved Instances (scope) (p. 247)
- Types of Reserved Instances (offering classes) (p. 248)
- How Reserved Instances are applied (p. 249)
- Use your Reserved Instances (p. 254)
- How you are billed (p. 254)
- Buying Reserved Instances (p. 259)
- Sell in the Reserved Instance Marketplace (p. 267)
- Modify Reserved Instances (p. 273)
- Exchange Convertible Reserved Instances (p. 277)

Reserved Instance overview

The following diagram shows a basic overview of purchasing and using Reserved Instances.

In this scenario, you have a running On-Demand Instance (T2) in your account, for which you're currently paying On-Demand rates. You purchase a Reserved Instance that matches the attributes of your running instance, and the billing benefit is immediately applied. Next, you purchase a Reserved Instance for
a C4 instance. You do not have any running instances in your account that match the attributes of this Reserved Instance. In the final step, you launch an instance that matches the attributes of the C4 Reserved Instance, and the billing benefit is immediately applied.

**Key variables that determine Reserved Instance pricing**

The Reserved Instance pricing is determined by the following key variables.

**Instance attributes**

A Reserved Instance has four instance attributes that determine its price.

- **Instance type**: For example, m4.large. This is composed of the instance family (for example, m4) and the instance size (for example, large).
- **Region**: The Region in which the Reserved Instance is purchased.
- **Tenancy**: Whether your instance runs on shared (default) or single-tenant (dedicated) hardware. For more information, see Dedicated Instances (p. 360).
- **Platform**: The operating system; for example, Windows or Linux/Unix. For more information, see Choosing a platform (p. 259).

**Term commitment**

You can purchase a Reserved Instance for a one-year or three-year commitment, with the three-year commitment offering a bigger discount.

- **One-year**: A year is defined as 31536000 seconds (365 days).
- **Three-year**: Three years is defined as 94608000 seconds (1095 days).

Reserved Instances do not renew automatically; when they expire, you can continue using the EC2 instance without interruption, but you are charged On-Demand rates. In the above example, when the Reserved Instances that cover the T2 and C4 instances expire, you go back to paying the On-Demand rates until you terminate the instances or purchase new Reserved Instances that match the instance attributes.

**Payment options**

The following payment options are available for Reserved Instances:

- **All Upfront**: Full payment is made at the start of the term, with no other costs or additional hourly charges incurred for the remainder of the term, regardless of hours used.
- **Partial Upfront**: A portion of the cost must be paid upfront and the remaining hours in the term are billed at a discounted hourly rate, regardless of whether the Reserved Instance is being used.
- **No Upfront**: You are billed a discounted hourly rate for every hour within the term, regardless of whether the Reserved Instance is being used. No upfront payment is required.

**Note**

No Upfront Reserved Instances are based on a contractual obligation to pay monthly for the entire term of the reservation. For this reason, a successful billing history is required before you can purchase No Upfront Reserved Instances.

Generally speaking, you can save more money making a higher upfront payment for Reserved Instances. You can also find Reserved Instances offered by third-party sellers at lower prices and shorter term lengths on the Reserved Instance Marketplace. For more information, see Sell in the Reserved Instance Marketplace (p. 267).
Offering class

If your computing needs change, you might be able to modify or exchange your Reserved Instance, depending on the offering class.

- **Standard**: These provide the most significant discount, but can only be modified. Standard Reserved Instances can't be exchanged.
- **Convertible**: These provide a lower discount than Standard Reserved Instances, but can be exchanged for another Convertible Reserved Instance with different instance attributes. Convertible Reserved Instances can also be modified.

For more information, see Types of Reserved Instances (offering classes) (p. 248).

After you purchase a Reserved Instance, you cannot cancel your purchase. However, you might be able to modify (p. 273), exchange (p. 277), or sell (p. 267) your Reserved Instance if your needs change.

For more information, see the Amazon EC2 Reserved Instances Pricing page.

Reserved Instance limits

There is a limit to the number of Reserved Instances that you can purchase per month. For each Region you can purchase 20 regional (p. 249) Reserved Instances per month plus an additional 20 zonal (p. 249) 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 (20x3=60).

A regional Reserved Instance applies a discount to a running On-Demand Instance. The default On-Demand Instance limit is 20. You cannot exceed your running On-Demand Instance limit by purchasing regional Reserved Instances. For example, if you already have 20 running On-Demand Instances, and you purchase 20 regional Reserved Instances, the 20 regional Reserved Instances are used to apply a discount to the 20 running On-Demand Instances. If you purchase more regional Reserved Instances, you will not be able to launch more instances because you have reached your On-Demand Instance limit.

Before purchasing regional Reserved Instances, make sure your On-Demand Instance limit matches or exceeds the number of regional Reserved Instances you intend to own. If required, make sure you request an increase to your On-Demand Instance limit before purchasing more regional Reserved Instances.

A zonal Reserved Instance—a Reserved Instance that is purchased for a specific Availability Zone—provides capacity reservation as well as a discount. You can exceed your running On-Demand Instance limit by purchasing zonal Reserved Instances. For example, if you already have 20 running On-Demand Instances, and you purchase 20 zonal Reserved Instances, you can launch a further 20 On-Demand Instances that match the specifications of your zonal Reserved Instances, giving you a total of 40 running instances.

The Amazon EC2 console provides limit information. For more information, see View your current limits (p. 1463).

Regional and zonal Reserved Instances (scope)

When you purchase a Reserved Instance, you determine the scope of the Reserved Instance. The scope is either regional or zonal.

- **Regional**: When you purchase a Reserved Instance for a Region, it's referred to as a *regional* Reserved Instance.
- **Zonal**: When you purchase a Reserved Instance for a specific Availability Zone, it's referred to as a *zonal* Reserved Instance.
The scope does not affect the price. You pay the same price for a regional or zonal Reserved Instance. For more information about Reserved Instance pricing, see Key variables that determine Reserved Instance pricing (p. 246) and Amazon EC2 Reserved Instances Pricing.

Differences between regional and zonal Reserved Instances

The following table highlights some key differences between regional Reserved Instances and zonal Reserved Instances:

<table>
<thead>
<tr>
<th></th>
<th>Regional Reserved Instances</th>
<th>Zonal Reserved Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ability to reserve capacity</td>
<td>A regional Reserved Instance does not reserve capacity.</td>
<td>A zonal Reserved Instance reserves capacity in the specified Availability Zone.</td>
</tr>
<tr>
<td>Availability Zone flexibility</td>
<td>The Reserved Instance discount applies to instance usage in any Availability Zone in the specified Region.</td>
<td>No Availability Zone flexibility—the Reserved Instance discount applies to instance usage in the specified Availability Zone only.</td>
</tr>
<tr>
<td>Instance size flexibility</td>
<td>The Reserved Instance discount applies to instance usage within the instance family, regardless of size. Only supported on Amazon Linux/Unix Reserved Instances with default tenancy. For more information, see Instance size flexibility determined by normalization factor (p. 250).</td>
<td>No instance size flexibility—the Reserved Instance discount applies to instance usage for the specified instance type and size only.</td>
</tr>
<tr>
<td>Queuing a purchase</td>
<td>You can queue purchases for regional Reserved Instances.</td>
<td>You can't queue purchases for zonal Reserved Instances.</td>
</tr>
</tbody>
</table>

For more information and examples, see How Reserved Instances are applied (p. 249).

Types of Reserved Instances (offering classes)

The offering class of a Reserved Instance is either Standard or Convertible. A Standard Reserved Instance provides a more significant discount than a Convertible Reserved Instance, but you can't exchange a Standard Reserved Instance. You can exchange Convertible Reserved Instances. You can modify Standard and Convertible Reserved Instances.

The configuration of a Reserved Instance comprises a single instance type, platform, scope, and tenancy over a term. If your computing needs change, you might be able to modify or exchange your Reserved Instance.

Differences between Standard and Convertible Reserved Instances

The following are the differences between Standard and Convertible Reserved Instances:

<table>
<thead>
<tr>
<th></th>
<th>Standard Reserved Instance</th>
<th>Convertible Reserved Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modifying Reserved Instances</td>
<td>Some attributes can be modified. For more information, see Modify Reserved Instances (p. 273).</td>
<td>Some attributes can be modified. For more information, see Modify Reserved Instances (p. 273).</td>
</tr>
</tbody>
</table>
# Reserved Instances

<table>
<thead>
<tr>
<th>Standard Reserved Instance</th>
<th>Convertible Reserved Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exchanging Reserved Instances</td>
<td>Can't be exchanged.</td>
</tr>
</tbody>
</table>

## 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 Use your Reserved Instances (p. 254).

Reserved Instances apply to usage in the same manner, irrespective of the offering type (Standard or Convertible), and are automatically applied to running On-Demand Instances with matching attributes.

## How zonal Reserved Instances are applied

Reserved Instances assigned to a specific Availability Zone provide the Reserved Instance discount to matching instance usage in that Availability Zone. For example, if you purchase two `c4.xlarge` default tenancy Linux/Unix Standard Reserved Instances in Availability Zone `us-east-1a`, then up to two `c4.xlarge` default tenancy Linux/Unix instances running in the Availability Zone `us-east-1a` can benefit from the Reserved Instance discount. The attributes (tenancy, platform, Availability Zone, instance type, and instance size) of the running instances must match that of the Reserved Instances.

## How regional Reserved Instances are applied

Regional Reserved Instances are purchased for a Region and provide Availability Zone flexibility. The Reserved Instance discount applies to instance usage in any Availability Zone in that Region.

Regional Reserved Instances also provide instance size flexibility where the Reserved Instance discount applies to instance usage within the instance family, regardless of size.

## Limitations for instance size flexibility

Instance size flexibility does not apply to the following Reserved Instances:

- Reserved Instances that are purchased for a specific Availability Zone (zonal Reserved Instances)
- Reserved Instances with dedicated tenancy
- Reserved Instances for G4dn instances
Instance size flexibility determined by normalization factor

Instance size flexibility is determined by the normalization factor of the instance size. The discount applies either fully or partially to running instances of the same instance family, depending on the instance size of the reservation, in any Availability Zone in the Region. The only attributes that must be matched are the instance family, tenancy, and platform.

Instance size flexibility is applied from the smallest to the largest instance size within the instance family based on the normalization factor.

The following table lists the different sizes within an instance family, and the corresponding normalization factor per hour. This scale is used to apply the discounted rate of Reserved Instances to the normalized usage of the instance family.

<table>
<thead>
<tr>
<th>Instance size</th>
<th>Normalization factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>nano</td>
<td>0.25</td>
</tr>
<tr>
<td>micro</td>
<td>0.5</td>
</tr>
<tr>
<td>small</td>
<td>1</td>
</tr>
<tr>
<td>medium</td>
<td>2</td>
</tr>
<tr>
<td>large</td>
<td>4</td>
</tr>
<tr>
<td>xlarge</td>
<td>8</td>
</tr>
<tr>
<td>2xlarge</td>
<td>16</td>
</tr>
<tr>
<td>3xlarge</td>
<td>24</td>
</tr>
<tr>
<td>4xlarge</td>
<td>32</td>
</tr>
<tr>
<td>6xlarge</td>
<td>48</td>
</tr>
<tr>
<td>8xlarge</td>
<td>64</td>
</tr>
<tr>
<td>9xlarge</td>
<td>72</td>
</tr>
<tr>
<td>10xlarge</td>
<td>80</td>
</tr>
<tr>
<td>12xlarge</td>
<td>96</td>
</tr>
<tr>
<td>16xlarge</td>
<td>128</td>
</tr>
<tr>
<td>18xlarge</td>
<td>144</td>
</tr>
<tr>
<td>24xlarge</td>
<td>192</td>
</tr>
<tr>
<td>32xlarge</td>
<td>256</td>
</tr>
<tr>
<td>56xlarge</td>
<td>448</td>
</tr>
<tr>
<td>112xlarge</td>
<td>896</td>
</tr>
</tbody>
</table>

For example, a t2.medium instance has a normalization factor of 2. If you purchase a t2.medium default tenancy Amazon Linux/Unix Reserved Instance in the US East (N. Virginia) and you have two running t2.small instances in your account in that Region, the billing benefit is applied in full to both instances.
Or, if you have one `t2.large` instance running in your account in the US East (N. Virginia) Region, the billing benefit is applied to 50% of the usage of the instance.

The normalization factor is also applied when modifying Reserved Instances. For more information, see `Modify Reserved Instances (p. 273)`.

**Normalization factor for bare metal instances**

Instance size flexibility also applies to bare metal instances within the instance family. If you have regional Amazon Linux/Unix Reserved Instances with shared tenancy on bare metal instances, you can benefit from the Reserved Instance savings within the same instance family. The opposite is also true: if you have regional Amazon Linux/Unix Reserved Instances with shared tenancy on instances in the same family as a bare metal instance, you can benefit from the Reserved Instance savings on the bare metal instance.

The metal instance size does not have a single normalization factor. A bare metal instance has the same normalization factor as the equivalent virtualized instance size within the same instance family. For example, an `i3.metal` instance has the same normalization factor as an `i3.16xlarge` instance.

<table>
<thead>
<tr>
<th>Instance size</th>
<th>Normalization factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>m5zn.metal</td>
<td>96</td>
</tr>
<tr>
<td>i3.metal</td>
<td>128</td>
</tr>
<tr>
<td>c5n.metal</td>
<td>144</td>
</tr>
<tr>
<td>c5.metal</td>
<td>192</td>
</tr>
<tr>
<td>m5n.metal</td>
<td>192</td>
</tr>
<tr>
<td>r5n.metal</td>
<td>192</td>
</tr>
<tr>
<td>u-*.metal</td>
<td>896</td>
</tr>
</tbody>
</table>
For example, an `i3.metal` instance has a normalization factor of 128. If you purchase an `i3.metal` default tenancy Amazon Linux/Unix Reserved Instance in the US East (N. Virginia), the billing benefit can apply as follows:

- If you have one running `i3.16xlarge` in your account in that Region, the billing benefit is applied in full to the `i3.16xlarge` instance (`i3.16xlarge` normalization factor = 128).
- Or, if you have two running `i3.8xlarge` instances in your account in that Region, the billing benefit is applied in full to both `i3.8xlarge` instances (`i3.8xlarge` normalization factor = 64).
- Or, if you have four running `i3.4xlarge` instances in your account in that Region, the billing benefit is applied in full to all four `i3.4xlarge` instances (`i3.4xlarge` normalization factor = 32).

The opposite is also true. For example, if you purchase two `i3.8xlarge` default tenancy Amazon Linux/Unix Reserved Instances in the US East (N. Virginia), and you have one running `i3.metal` instance in that Region, the billing benefit is applied in full to the `i3.metal` instance.

**Examples of applying Reserved Instances**

The following scenarios cover the ways in which Reserved Instances are applied.

**Example Scenario 1: Reserved Instances in a single account**

You are running the following On-Demand Instances in account A:

- 4 x `m3.large` Linux, default tenancy instances in Availability Zone us-east-1a
- 2 x `m4.xlarge` Amazon Linux, default tenancy instances in Availability Zone us-east-1b
- 1 x `c4.xlarge` Amazon Linux, default tenancy instances in Availability Zone us-east-1c

You purchase the following Reserved Instances in account A:

- 4 x `m3.large` Linux, default tenancy Reserved Instances in Availability Zone us-east-1a (capacity is reserved)
- 4 x `m4.large` Amazon Linux, default tenancy Reserved Instances in Region us-east-1
- 1 x `c4.large` Amazon Linux, default tenancy Reserved Instances in Region us-east-1

The Reserved Instance benefits are applied in the following way:

- The discount and capacity reservation of the four `m3.large` zonal Reserved Instances is used by the four `m3.large` instances because the attributes (instance size, Region, platform, tenancy) between them match.
- The `m4.large` regional Reserved Instances provide Availability Zone and instance size flexibility, because they are regional Amazon Linux Reserved Instances with default tenancy.

An `m4.large` is equivalent to 4 normalized units/hour.

You've purchased four `m4.large` regional Reserved Instances, and in total, they are equal to 16 normalized units/hour (4x4). Account A has two `m4.xlarge` instances running, which is equivalent to 16 normalized units/hour (2x8). In this case, the four `m4.large` regional Reserved Instances provide the billing benefit to an entire hour of usage of the two `m4.xlarge` instances.

- The `c4.large` regional Reserved Instance in us-east-1 provides Availability Zone and instance size flexibility, because it is a regional Amazon Linux Reserved Instance with default tenancy, and applies to the `c4.xlarge` instance. A `c4.large` instance is equivalent to 4 normalized units/hour and a `c4.xlarge` is equivalent to 8 normalized units/hour.

In this case, the `c4.large` regional Reserved Instance provides partial benefit to `c4.xlarge` usage. This is because the `c4.large` Reserved Instance is equivalent to 4 normalized units/hour of usage,
but the c4.xlarge instance requires 8 normalized units/hour. Therefore, the c4.large Reserved Instance billing discount applies to 50% of c4.xlarge usage. The remaining c4.xlarge usage is charged at the On-Demand rate.

Example Scenario 2: Regional Reserved Instances in linked accounts

Reserved Instances are first applied to usage within the purchasing account, followed by qualifying usage in any other account in the organization. For more information, see Reserved Instances and consolidated billing (p. 256). For regional Reserved Instances that offer instance size flexibility, the benefit is applied from the smallest to the largest instance size within the instance family.

You're running the following On-Demand Instances in account A (the purchasing account):

- 2 x m4.xlarge Linux, default tenancy instances in Availability Zone us-east-1a
- 1 x m4.2xlarge Linux, default tenancy instances in Availability Zone us-east-1b
- 2 x c4.xlarge Linux, default tenancy instances in Availability Zone us-east-1a
- 1 x c4.2xlarge Linux, default tenancy instances in Availability Zone us-east-1b

Another customer is running the following On-Demand Instances in account B—a linked account:

- 2 x m4.xlarge Linux, default tenancy instances in Availability Zone us-east-1a

You purchase the following regional Reserved Instances in account A:

- 4 x m4.xlarge Linux, default tenancy Reserved Instances in Region us-east-1
- 2 x c4.xlarge Linux, default tenancy Reserved Instances in Region us-east-1

The regional Reserved Instance benefits are applied in the following way:

- The discount of the four m4.xlarge Reserved Instances is used by the two m4.xlarge instances and the single m4.2xlarge instance in account A (purchasing account). All three instances match the attributes (instance family, Region, platform, tenancy). The discount is applied to instances in the purchasing account (account A) first, even though account B (linked account) has two m4.xlarge that also match the Reserved Instances. There is no capacity reservation because the Reserved Instances are regional Reserved Instances.
- The discount of the two c4.xlarge Reserved Instances applies to the two c4.xlarge instances, because they are a smaller instance size than the c4.2xlarge instance. There is no capacity reservation because the Reserved Instances are regional Reserved Instances.

Example Scenario 3: Zonal Reserved Instances in a linked account

In general, Reserved Instances that are owned by an account are applied first to usage in that account. However, if there are qualifying, unused Reserved Instances for a specific Availability Zone (zonal Reserved Instances) in other accounts in the organization, they are applied to the account before regional Reserved Instances owned by the account. This is done to ensure maximum Reserved Instance utilization and a lower bill. For billing purposes, all the accounts in the organization are treated as one account. The following example might 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.

Use 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 zonal 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. 360).

For more information, see Launch an instance using the Launch Instance Wizard (p. 392). For examples of how Reserved Instances are applied to your running instances, see How Reserved Instances are applied (p. 249).

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.

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. 246) specified for the Reserved Instance.

When Reserved Instances expire, you are charged On-Demand rates for EC2 instance usage. You can queue a Reserved Instance for purchase up to three years in advance. This can help you ensure that you have uninterrupted coverage. For more information, see Queue your purchase (p. 260).

The AWS Free Tier is available for new AWS accounts. If you are using the AWS Free Tier to run Amazon EC2 instances, and you purchase a Reserved Instance, you are charged under standard pricing guidelines. For information, see AWS Free Tier.

Contents
- Usage billing (p. 255)
- Viewing your bill (p. 256)
- Reserved Instances and consolidated billing (p. 256)
- Reserved Instance discount pricing tiers (p. 256)

Usage billing

Reserved Instances are billed for every clock-hour during the term that you select, regardless of whether an instance is running. Each clock-hour starts on the hour (zero minutes and zero seconds past the hour) of a standard 24-hour clock. For example, 1:00:00 to 1:59:59 is one clock-hour. For more information about instance states, see Instance lifecycle (p. 386).

A Reserved Instance billing benefit can be applied to a running instance on a per-second basis.

A Reserved Instance billing benefit can apply to a maximum of 3600 seconds (one hour) of instance usage per clock-hour. You can run multiple instances concurrently, but can only receive the benefit of the Reserved Instance discount for a total of 3600 seconds per clock-hour; instance usage that exceeds 3600 seconds in a clock-hour is billed at the On-Demand rate.

For example, if you purchase one m4.xlarge Reserved Instance and run four m4.xlarge instances concurrently for one hour, one instance is charged at one hour of Reserved Instance usage and the other three instances are charged at three hours of On-Demand usage.

However, if you purchase one m4.xlarge Reserved Instance and run four m4.xlarge instances for 15 minutes (900 seconds) each within the same hour, the total running time for the instances is one hour, which results in one hour of Reserved Instance usage and 0 hours of On-Demand usage.

If multiple eligible instances are running concurrently, the Reserved Instance billing benefit is applied to all the instances at the same time up to a maximum of 3600 seconds in a clock-hour; thereafter, On-Demand rates apply.
Cost Explorer on the Billing and Cost Management console enables you to analyze the savings against running On-Demand Instances. The Reserved Instances FAQ includes an example of a list value calculation.

If you close your AWS account, On-Demand billing for your resources stops. However, if you have any Reserved Instances in your account, you continue to receive a bill for these until they expire.

Viewing your bill

You can find out about the charges and fees to your account by viewing the AWS Billing and Cost Management console.

- The Dashboard displays a spend summary for your account.
- On the Bills page, under Details expand the Elastic Compute Cloud section and the Region to get billing information about your Reserved Instances.

You can view the charges online, or you can download a CSV file.

You can also track your Reserved Instance utilization using the AWS Cost and Usage Report. For more information, see Reserved Instances under Cost and Usage Report in the AWS Billing and Cost Management User Guide.

Reserved Instances and consolidated billing

The pricing benefits of Reserved Instances are shared when the purchasing account is part of a set of accounts billed under one consolidated billing payer account. The instance usage across all member accounts is aggregated in the payer account every month. This is typically useful for companies in which there are different functional teams or groups; then, the normal Reserved Instance logic is applied to calculate the bill. For more information, see Consolidated billing for AWS Organizations.

If you close the account that purchased the Reserved Instance, the payer account is charged for the Reserved Instance until the Reserved Instance expires. After the closed account is permanently deleted in 90 days, the member accounts no longer benefit from the Reserved Instance billing discount.

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
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**

- Calculate Reserved Instance pricing discounts (p. 257)
- Buy with a discount tier (p. 258)
- Crossing pricing tiers (p. 258)
- Consolidated billing for pricing tiers (p. 258)

**Calculate Reserved Instance pricing discounts**

You can determine the pricing tier for your account by calculating the list value for all of your Reserved Instances in a Region. Multiply the hourly recurring price for each reservation by the total number of hours for the term and add the undiscounted upfront price (also known as the fixed price) at the time of purchase. Because the list value is based on undiscounted (public) pricing, it is not affected if you qualify for a volume discount or if the price drops after you buy your Reserved Instances.

\[
\text{List value} = \text{fixed price} + (\text{undiscounted recurring hourly price} \times \text{hours in term})
\]

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.

\[
121.32 = 60.00 + (0.007 \times 8760)
\]

**New console**

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. To display the Upfront price column, choose the settings icon (⚙️) in the top-right corner, toggle on Upfront price, and choose Confirm.

**Old console**

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.
To display the **Upfront Price** column, choose the settings icon ( ) in the top-right corner, select **Upfront Price**, and choose **Close**.

### 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)

### Buy 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. 259).

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. 258).

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
purchasing 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. 256).

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. 259)
- Queue your purchase (p. 260)
- Buy Standard Reserved Instances (p. 260)
- Buy Convertible Reserved Instances (p. 263)
- Buy from the Reserved Instance Marketplace (p. 265)
- View your Reserved Instances (p. 265)
- Cancel a queued purchase (p. 266)
- Renew a Reserved Instance (p. 266)

Choosing a platform

Amazon EC2 supports the following Windows platforms for Reserved Instances:

- Windows
- Windows with SQL Server Standard
- Windows with SQL Server Web
- Windows with SQL Server Enterprise

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.
Important

If you plan to purchase a Reserved Instance to apply to an On-Demand Instance that was launched from an AWS Marketplace AMI, first check the PlatformDetails field of the AMI. The PlatformDetails field indicates which Reserved Instance to purchase. The platform details of the AMI must match the platform of the Reserved Instance, otherwise the Reserved Instance will not be applied to the On-Demand Instance. For information about how to view the platform details of the AMI, see Understand AMI billing information (p. 133).

For information about the supported platforms for Linux, see Choosing a platform in the Amazon EC2 User Guide for Linux Instances.

Queue your purchase

By default, when you purchase a Reserved Instance, the purchase is made immediately. Alternatively, you can queue your purchases for a future date and time. For example, you can queue a purchase for around the time that an existing Reserved Instance expires. This can help you ensure that you have uninterrupted coverage.

You can queue purchases for regional Reserved Instances, but not zonal Reserved Instances or Reserved Instances from other sellers. You can queue a purchase up to three years in advance. On the scheduled date and time, the purchase is made using the default payment method. After the payment is successful, the billing benefit is applied.

You can view your queued purchases in the Amazon EC2 console. The status of a queued purchase is queued. You can cancel a queued purchase any time before its scheduled time. For details, see Cancel a queued purchase (p. 266).

Buy 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.

New console

To buy Standard Reserved Instances using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Reserved Instances, and then choose Purchase Reserved Instances.
3. For Offering class, choose Standard to display Standard Reserved Instances.
4. To purchase a capacity reservation, toggle on Only show offerings that reserve capacity in the top-right corner of the purchase screen. When you toggle on this setting, the Availability Zone field appears.
   To purchase a regional Reserved Instance, toggle off this setting. When you toggle off this setting, the Availability Zone field disappears.
5. Select other configurations as needed, and then choose Search.
6. For each Reserved Instance that you want to purchase, enter the desired quantity, and choose Add to cart.

   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. For more information, see Buy from the Reserved Instance Marketplace (p. 265).
7. To see a summary of the Reserved Instances that you selected, choose View cart.
8. If Order on is Now, the purchase is completed immediately after you choose Order all. To queue a purchase, choose Now and select a date. You can select a different date for each eligible offering in the cart. The purchase is queued until 00:00 UTC on the selected date.
9. To complete the order, choose **Order all**.

   If, at the time of placing the order, there are offerings similar to your choice but with a lower price, AWS sells you the offerings at the lower price.

10. Choose **Close**.

    The status of your order is listed in the **State** column. When your order is complete, the **State** value changes from **Payment-pending** to **Active**. When the Reserved Instance is **Active**, it is ready to use.

**Note**

If the status goes to **Retired**, AWS might not have received your payment.

---

Old console

**To buy Standard Reserved Instances using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Reserved Instances**, and then choose **Purchase Reserved Instances**.
3. For **Offering Class**, choose **Standard** to display Standard Reserved Instances.
4. To purchase a capacity reservation, choose **Only show offerings that reserve capacity** in the top-right corner of the purchase screen. To purchase a regional Reserved Instance, leave the check box unselected.
5. Select other configurations as needed and choose **Search**.

   To purchase a Standard Reserved Instance from the Reserved Instance Marketplace, look for **3rd Party** in the **Seller** column in the search results. The **Term** column displays non-standard terms.

6. For each Reserved Instance that you want to purchase, enter the quantity, and choose **Add to Cart**.
7. To see a summary of the Reserved Instances that you selected, choose **View Cart**.
8. If **Order On** is **Now**, the purchase is completed immediately. To queue a purchase, choose **Now** and select a date. You can select a different date for each eligible offering in the cart. The purchase is queued until 00:00 UTC on the selected date.
9. To complete the order, choose **Order**.

   If, at the time of placing the order, there are offerings similar to your choice but with a lower price, AWS sells you the offerings at the lower price.

10. Choose **Close**.

    The status of your order is listed in the **State** column. When your order is complete, the **State** value changes from **payment-pending** to **active**. When the Reserved Instance is **active**, it is ready to use.

**Note**

If the status goes to **retired**, AWS might 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:
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 might be shorter than a 1– or 3-year term.

```
aws ec2 describe-reserved-instances-offerings \
  --instance-type t2.large \
  --offering-class standard \
  --product-description "Linux/UNIX" \
  --instance-tenancy default \
  --filters Name=marketplace,Values=true
```

When you find a Reserved Instance that meets your needs, take note of the offering ID. For example:

```
"ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2"
```

2. Use the `purchase-reserved-instances-offering` command to buy your Reserved Instance. You must specify the Reserved Instance offering ID you obtained the previous step and you must specify the number of instances for the reservation.

```
aws ec2 purchase-reserved-instances-offering \
  --reserved-instances-offering-id bec624df-a8cc-4aad-a72f-4f8abc34caf2 \
  --instance-count 1
```

By default, the purchase is completed immediately. Alternatively, to queue the purchase, add the following parameter to the previous call.

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. Use the `describe-reserved-instances` command to get the status of your Reserved Instance.

```
aws ec2 describe-reserved-instances
```

Alternatively, use the following AWS Tools for Windows PowerShell commands:

- `Get-EC2ReservedInstancesOffering`
- `New-EC2ReservedInstance`
- `Get-EC2ReservedInstance`

After the purchase is complete, if you already have a running instance that matches the specifications of the Reserved Instance, the billing benefit is immediately applied. You do not have to restart your instances. If you do not have a suitable running instance, launch an instance and ensure that you match the same criteria that you specified for your Reserved Instance. For more information, see Use your Reserved Instances (p. 254).

For examples of how Reserved Instances are applied to your running instances, see How Reserved Instances are applied (p. 249).
Buy 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.

New console

**To buy Convertible Reserved Instances using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Reserved Instances**, and then choose **Purchase Reserved Instances**.
3. For **Offering class**, choose **Convertible** to display Convertible Reserved Instances.
4. To purchase a capacity reservation, toggle on **Only show offerings that reserve capacity** in the top-right corner of the purchase screen. When you toggle on this setting, the **Availability Zone** field appears.
   
   To purchase a regional Reserved Instance, toggle off this setting. When you toggle off this setting, the **Availability Zone** field disappears.
5. Select other configurations as needed and choose **Search**.
6. For each Convertible Reserved Instance that you want to purchase, enter the quantity, and choose **Add to cart**.
7. To see a summary of your selection, choose **View cart**.
8. If **Order on** is **Now**, the purchase is completed immediately after you choose **Order all**. To queue a purchase, choose **Now** and select a date. You can select a different date for each eligible offering in the cart. The purchase is queued until 00:00 UTC on the selected date.
9. To complete the order, choose **Order all**.
   
   If, at the time of placing the order, there are offerings similar to your choice but with a lower price, AWS sells you the offerings at the lower price.
10. Choose **Close**.

   The status of your order is listed in the **State** column. When your order is complete, the **State** value changes from Payment-pending to **Active**. When the Reserved Instance is **Active**, it is ready to use.

**Note**

If the status goes to **Retired**, AWS might not have received your payment.

Old console

**To buy Convertible Reserved Instances using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Reserved Instances**, and then choose **Purchase Reserved Instances**.
3. For **Offering Class**, choose **Convertible** to display Convertible Reserved Instances.
4. To purchase a capacity reservation, choose **Only show offerings that reserve capacity** in the top-right corner of the purchase screen. To purchase a regional Reserved Instance, leave the check box unselected.
5. Select other configurations as needed and choose **Search**.
6. For each Convertible Reserved Instance that you want to purchase, enter the quantity, and choose **Add to Cart**.
7. To see a summary of your selection, choose **View Cart**.
8. If Order On is Now, the purchase is completed immediately. To queue a purchase, choose Now and select a date. You can select a different date for each eligible offering in the cart. The purchase is queued until 00:00 UTC on the selected date.

9. To complete the order, choose Order.

If, at the time of placing the order, there are offerings similar to your choice but with a lower price, AWS sells you the offerings at the lower price.

10. Choose Close.

The status of your order is listed in the State column. When your order is complete, the State value changes from payment-pending to active. When the Reserved Instance is active, it is ready to use.

Note
If the status goes to retired, AWS might not have received your payment.

To buy a Convertible Reserved Instance using the AWS CLI

1. Find available Reserved Instances using the describe-reserved-instances-offerings command. Specify convertible for the --offering-class parameter to return only Convertible Reserved Instances. You can apply additional parameters to narrow your results; for example, if you want to purchase a regional t2.large Reserved Instance with a default tenancy for Linux/UNIX:

```
aws ec2 describe-reserved-instances-offerings
   --instance-type t2.large \ 
   --offering-class convertible \ 
   --product-description "Linux/UNIX" \ 
   --instance-tenancy default \ 
   --filters Name=scope,Values=Region
```

When you find a Reserved Instance that meets your needs, take note of the offering ID. For example:

```
"ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2"
```

2. Use the purchase-reserved-instances-offering command to buy your Reserved Instance. You must specify the Reserved Instance offering ID you obtained the previous step and you must specify the number of instances for the reservation.

```
aws ec2 purchase-reserved-instances-offering
   --reserved-instances-offering-id bec624df-a8cc-4aad-a72f-4f8abc34caf2 \ 
   --instance-count 1
```

By default, the purchase is completed immediately. Alternatively, to queue the purchase, add the following parameter to the previous call.

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. Use the describe-reserved-instances command to get the status of your Reserved Instance.

```
aws ec2 describe-reserved-instances
```

Alternatively, use the following AWS Tools for Windows PowerShell commands:

- Get-EC2ReservedInstancesOffering
• New-EC2ReservedInstance
• Get-EC2ReservedInstance

If you already have a running instance that matches the specifications of the Reserved Instance, the billing benefit is immediately applied. You do not have to restart your instances. If you do not have a suitable running instance, launch an instance and ensure that you match the same criteria that you specified for your Reserved Instance. For more information, see Use your Reserved Instances (p. 254).

For examples of how Reserved Instances are applied to your running instances, see How Reserved Instances are applied (p. 249).

Buy from the Reserved Instance Marketplace

You can purchase Reserved Instances from third-party sellers who own Reserved Instances that they no longer need from the Reserved Instance Marketplace. You can do this using the Amazon EC2 console or a command line tool. The process is similar to purchasing Reserved Instances from AWS. For more information, see Buy Standard Reserved Instances (p. 260).

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.

View 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 queued, 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. 267)**. For more information, see Reserved Instance listing states (p. 271).
To view your Reserved Instances using the command line

- describe-reserved-instances (AWS CLI)
- Get-EC2ReservedInstance (Tools for Windows PowerShell)

Cancel a queued purchase

You can queue a purchase up to three years in advance. You can cancel a queued purchase any time before its scheduled time.

New console

To cancel a queued purchase

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Reserved Instances.
3. Select one or more Reserved Instances.
4. Choose Actions, Delete queued Reserved Instances.
5. When prompted for confirmation, choose Delete, and then Close.

Old console

To cancel a queued purchase

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Reserved Instances.
3. Select one or more Reserved Instances.
4. Choose Actions, Delete Queued Reserved Instances.
5. When prompted for confirmation, choose Yes, Delete.

To cancel a queued purchase using the command line

- delete-queued-reserved-instances (AWS CLI)
- Remove-EC2QueuedReservedInstance (Tools for Windows PowerShell)

Renew a Reserved Instance

You can renew a Reserved Instance before it is scheduled to expire. Renewing a Reserved Instance queues the purchase of a Reserved Instance with the same configuration until the current Reserved Instance expires.

New console

To renew an Reserved Instance using a queued purchase

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Reserved Instances.
3. Select one or more Reserved Instances.
5. To complete the order, choose Order all, and then Close.
Old console

To renew an Reserved Instance using a queued purchase

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Reserved Instances.
3. Select one or more Reserved Instances.
5. To complete the order, choose Order.

Sell in 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 might 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.

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, AWS sells 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.

If you want to sell your unused Reserved Instances on the Reserved Instance Marketplace, you must meet certain eligibility criteria.

For information about buying Reserved Instances on the Reserved Instance Marketplace, see Buy from the Reserved Instance Marketplace (p. 265).

Contents

- Restrictions and limitations (p. 267)
- Register as a seller (p. 268)
- Bank account for disbursement (p. 268)
- Tax information (p. 269)
- Price your Reserved Instances (p. 270)
- List your Reserved Instances (p. 270)
- Reserved Instance listing states (p. 271)
- Lifecycle of a listing (p. 271)
- After your Reserved Instance is sold (p. 272)
- Getting paid (p. 272)
- Information shared with the buyer (p. 273)

Restrictions and limitations

Before you can sell your unused reservations, you must register as a seller in the Reserved Instance Marketplace. For information, see Register as a seller (p. 268).
The following limitations and restrictions apply when selling Reserved Instances:

- Only Amazon EC2 Standard Reserved Instances can be sold in the Reserved Instance Marketplace. Amazon EC2 Convertible Reserved Instances cannot be sold. Reserved Instances for other AWS services, such as Amazon RDS and Amazon ElastiCache, cannot be sold.
- There must be at least one month remaining in the term of the Standard Reserved Instance.
- You cannot sell a Standard Reserved Instance in a Region that is disabled by default.
- 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 Price your Reserved Instances (p. 270). You can also modify your Reserved Instances before listing them. For information, see Modify Reserved Instances (p. 273).
- In order to list a regional Reserved Instance in the marketplace, you must modify the scope to zonal as it is not possible to sell regional Reserved Instances via the console.
- AWS charges a service fee of 12 percent of the total upfront price of each Standard Reserved Instance you sell in the Reserved Instance Marketplace. The upfront price is the price the seller is charging for the Standard Reserved Instance.
- When you register as a seller, the bank you specify must have a US address. For more information, see Bank account for disbursement (p. 268).
- Amazon Internet Services Private Limited (AISPL) customers can't sell Reserved Instances in the Reserved Instance Marketplace even if they have a US bank account. For more information, see What are the differences between AWS accounts and AISPL accounts?

Register as a seller

Note
Only the AWS account root user can register an account as a seller.

To sell in the Reserved Instance Marketplace, you must first register as a seller. During registration, you provide the following information:

- **Bank information**—AWS must have your bank information in order to disburse funds collected when you sell your reservations. The bank you specify must have a US address. For more information, see Bank account for disbursement (p. 268).
- **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. 269).

After AWS receives your completed seller registration, you receive an email confirming your registration and informing you that you can get started selling in the Reserved Instance Marketplace.

Bank account for disbursement

AWS must have your bank information in order to disburse funds collected when you sell your Reserved Instance. The bank you specify must have a US address. For more information, see Additional seller requirements for paid products in the AWS Marketplace Seller Guide.

To register a default bank account for disbursements

1. Open the Reserved Instance Marketplace Seller Registration page and sign in using your AWS credentials.
2. On the **Manage Bank Account** page, provide the following information about the bank through to receive payment:

   - Bank account holder name
   - Routing number
   - Account number
   - Bank account type

   **Note**
   If you are using a corporate bank account, you are prompted to send the information about the bank account via fax (1-206-765-3424).

After registration, the bank account provided is set as the default, pending verification with the bank. It can take up to two weeks to verify a new bank account, during which time you can't receive disbursements. For an established account, it usually takes about two days for disbursements to complete.

**To change the default bank account for disbursement**

1. On the **Reserved Instance Marketplace Seller Registration** page, sign in with the account that you used when you registered.
2. On the **Manage Bank Account** page, add a new bank account or modify the default bank account as needed.

**Tax information**

Your sale of Reserved Instances might be subject to a transaction-based tax, such as sales tax or value-added tax. You should check with your business's tax, legal, finance, or accounting department to determine if transaction-based taxes are applicable. You are responsible for collecting and sending the transaction-based taxes to the appropriate tax authority.

As part of the seller registration process, you must complete a tax interview in the **Seller Registration Portal**. The interview collects your tax information and populates an IRS form W-9, W-8BEN, or W-8BEN-E, which is used to determine any necessary tax reporting obligations.

The tax information you enter as part of the tax interview might differ depending on whether you operate as an individual or business, and whether you or your business are a US or non-US person or entity. As you fill out the tax interview, keep in mind the following:

- Information provided by AWS, including the information in this topic, does not constitute tax, legal, or other professional advice. To find out how the IRS reporting requirements might affect your business, or if you have other questions, contact your tax, legal, or other professional advisor.
- To fulfill the IRS reporting requirements as efficiently as possible, answer all questions and enter all information requested during the interview.
- Check your answers. Avoid misspellings or entering incorrect tax identification numbers. They can result in an invalidated tax form.

Based on your tax interview responses and IRS reporting thresholds, Amazon might file Form 1099-K. Amazon mails a copy of your Form 1099-K on or before January 31 in the year following the year that your tax account reaches the threshold levels. For example, if your account reaches the threshold in 2018, your Form 1099-K is mailed on or before January 31, 2019.

For more information about IRS requirements and Form 1099-K, see the **IRS** website.
Price your Reserved Instances

The upfront fee is the only fee that you can specify for the Reserved Instance that you're selling. The upfront fee is the one-time fee that the buyer pays when they purchase a Reserved Instance.

The following are important limits to note:

- **You can sell up to $50,000 in Reserved Instances.** To increase this limit, complete the EC2 Reserved Instance Sales form.
- **You can sell up to 5,000 Reserved Instances.** To increase this limit, complete the EC2 Reserved Instance Sales form.
- **The minimum price is $0.** The minimum allowed price in the Reserved Instance Marketplace is $0.00.

You cannot modify your listing directly. However, you can change your listing by first canceling it and then creating another listing with new parameters.

You can cancel your listing at any time, as long as it's in the active state. You cannot cancel the listing if it's already matched or being processed for a sale. If some of the instances in your listing are matched and you cancel the listing, only the remaining unmatched instances are removed from the listing.

Because the value of Reserved Instances decreases over time, by default, AWS can set prices to decrease in equal increments month over month. However, you can set different upfront prices based on when your reservation sells.

For example, if your Reserved Instance has nine months of its term remaining, you can specify the amount that you would accept if a customer were to purchase that Reserved Instance with nine months remaining. You could set another price with five months remaining, and yet another price with one month remaining.

List your Reserved Instances

As a registered seller, you can choose to sell one or more of your Reserved Instances. You can choose to sell all of them in one listing or in portions. In addition, you can list Reserved Instances with any configuration of instance type, platform, and scope.

The console determines a suggested price. It checks for offerings that match your Reserved Instance and matches the one with the lowest price. Otherwise, it calculates a suggested price based on the cost of the Reserved Instance for its remaining time. If the calculated value is less than $1.01, the suggested price is $1.01.

If you cancel your listing and a portion of that listing has already been sold, the cancellation is not effective on the portion that has been sold. Only the unsold portion of the listing is no longer available in the Reserved Instance Marketplace.

To list a Reserved Instance in the Reserved Instance Marketplace using the AWS Management Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Reserved Instances.
3. Select the Reserved Instances to list, and choose Actions, 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 the My Listings tab near the bottom of the page.

To manage Reserved Instances in the Reserved Instance Marketplace using the AWS CLI

1. Get a list of your Reserved Instances by using the describe-reserved-instances command.
2. Note the ID of the Reserved Instance you want to list and call create-reserved-instances-listing. You must specify the ID of the Reserved Instance, the number of instances, and the pricing schedule.
3. To view your listing, use the describe-reserved-instances-listings command.
4. To cancel your listing, use the cancel-reserved-instances-listings command.

Reserved Instance listing states

Listing State on the My Listings tab of the Reserved Instances page displays the current status of your listings:

The information displayed by Listing State is about the status of your listing in the Reserved Instance Marketplace. It is different from the status information that is displayed by the State column in the Reserved Instances page. This State information is about your reservation.

- active—The listing is available for purchase.
- canceled—The listing is canceled and isn’t available for purchase in the Reserved Instance Marketplace.
- closed—The Reserved Instance is not listed. A Reserved Instance might be closed because the sale of the listing was completed.

Lifecycle of a listing

When all the instances in your listing are matched and sold, the My Listings tab shows that the Total instance count matches the count listed under Sold. Also, there are no Available instances left for your listing, and its Status is closed.

When only a portion of your listing is sold, AWS retires the Reserved Instances in the listing and creates the number of Reserved Instances equal to the Reserved Instances remaining in the count. So, the listing ID and the listing that it represents, which now has fewer reservations for sale, is still active.

Any future sales of Reserved Instances in this listing are processed this way. When all the Reserved Instances in the listing are sold, AWS marks the listing as closed.

For example, you create a listing Reserved Instances listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample with a listing count of 5.

The My Listings tab in the Reserved Instance console page displays the listing this way:

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- Total reservation count = 5
• Sold = 0
• Available = 5
• Status = active

A buyer purchases two of the reservations, which leaves a count of three reservations still available for sale. Because of this partial sale, AWS creates a new reservation with a count of three to represent the remaining reservations that are still for sale.

This is how your listing looks in the My Listings tab:

Reserved Instance listing ID 5ec28771-05ff-4b9b-aa31-9e57dexample
• Total reservation count = 5
• Sold = 2
• Available = 3
• Status = active

If you cancel your listing and a portion of that listing has already sold, the cancelation 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. Activities can include when you create or sell a listing, or when AWS sends funds to your account.

To track the status of a Reserved Instance listing in the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation page, choose Reserved Instances.
3. Choose the My Listings tab.
   The My Listings tab contains the Listing State value. It also contains information about the term, listing price, and a breakdown of how many instances in the listing are available, pending, sold, and canceled.

   You can also use the describe-reserved-instances-listings command with the appropriate filter to obtain information about your listings.

Getting paid

As soon as AWS receives funds from the buyer, a message is sent to the registered owner account email for the sold Reserved Instance.

AWS sends an Automated Clearing House (ACH) wire transfer to your specified bank account. Typically, this transfer occurs between one to three days after your Reserved Instance has been sold. Disbursements take place once a day. You will receive an email with a disbursement report after the funds are released. Keep in mind that you can't receive disbursements until AWS receives verification from your bank. This can take up to two weeks.

The Reserved Instance that you sold continues to appear when you describe your Reserved Instances.

You receive a cash disbursement for your Reserved Instances through a wire transfer directly into your bank account. AWS charges a service fee of 12 percent of the total upfront price of each Reserved Instance you sell in the Reserved Instance Marketplace.
Information shared with the buyer

When you sell in the Reserved Instance Marketplace, AWS shares your company's legal name on the buyer's statement in accordance with US regulations. In addition, if the buyer calls AWS Support because the buyer needs to contact you for an invoice or for some other tax-related reason, AWS might 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.

Modify Reserved Instances

When your needs change, you can modify your Standard or Convertible Reserved Instances and continue to benefit from the billing benefit. You can modify attributes such as the Availability Zone and scope of your Reserved Instance.

Note
You can also exchange a Convertible Reserved Instance for another Convertible Reserved Instance with a different configuration. For more information, see Exchange Convertible Reserved Instances (p. 277).

After modification, the benefit of the Reserved Instances is applied only to instances that match the new parameters. For example, if you change the Availability Zone of a reservation, the capacity reservation and pricing benefits are automatically applied to instance usage in the new Availability Zone. Instances that no longer match the new parameters are charged at the On-Demand rate, unless your account has other applicable reservations.

If your modification request succeeds:

- The modified reservation becomes effective immediately and the pricing benefit is applied to the new instances beginning at the hour of the modification request. For example, if you successfully modify your reservations at 9:15PM, the pricing benefit transfers to your new instance at 9:00PM. You can get the effective date of the modified Reserved Instances by using the describe-reserved-instances command.
- The original reservation is retired. Its end date is the start date of the new reservation, and the end date of the new reservation is the same as the end date of the original Reserved Instance. If you modify a three-year reservation that had 16 months left in its term, the resulting modified reservation is a 16-month reservation with the same end date as the original one.
- The modified reservation lists a $0 fixed price and not the fixed price of the original reservation.
- The fixed price of the modified reservation does not affect the discount pricing tier calculations applied to your account, which are based on the fixed price of the original reservation.

If your modification request fails, your Reserved Instances maintain their original configuration, and are immediately available for another modification request.

There is no fee for modification, and you do not receive any new bills or invoices.

You can modify your reservations as frequently as you like, but you cannot change or cancel a pending modification request after you submit it. After the modification has completed successfully, you can submit another modification request to roll back any changes you made, if needed.

Contents
- Requirements and restrictions for modification (p. 274)
• Submit modification requests (p. 275)
• Troubleshoot modification requests (p. 276)

Requirements and restrictions for modification

You can modify these attributes as follows.

<table>
<thead>
<tr>
<th>Modifiable attribute</th>
<th>Supported platforms</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change <strong>Availability Zones</strong> within the same Region</td>
<td>Linux and Windows</td>
<td>-</td>
</tr>
<tr>
<td>Change the <strong>scope</strong> from Availability Zone to Region and vice versa</td>
<td>Linux and Windows</td>
<td>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. 249).</td>
</tr>
<tr>
<td>Change the <strong>instance size</strong> within the same instance family</td>
<td>Linux/UNIX only</td>
<td>The reservation must use default tenancy. Some instance families are not supported, because there are no other sizes available. For more information, see Support for modifying instance sizes in the Amazon EC2 User Guide for Linux Instances.</td>
</tr>
<tr>
<td>Change the <strong>network</strong> from EC2-Classic to Amazon VPC and vice versa</td>
<td>Linux and Windows</td>
<td>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.</td>
</tr>
</tbody>
</table>

Requirements

Amazon EC2 processes your modification request if there is sufficient capacity for your new configuration (if applicable), and if the following conditions are met:

• The Reserved Instance cannot be modified before or at the same time that you purchase it
• The Reserved Instance must be active
• There cannot be a pending modification request
• The Reserved Instance is not listed in the Reserved Instance Marketplace
• The original Reserved Instances are all Standard Reserved Instances or all Convertible Reserved Instances, not some of each type
• The original Reserved Instances must expire within the same hour, if they are Standard Reserved Instances
• The Reserved Instance is not a G4 instance.

Submit modification requests

Before you modify your Reserved Instances, ensure that you have read the applicable restrictions (p. 274).

New console

To modify your Reserved Instances using the AWS Management Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the Reserved Instances page, select one or more Reserved Instances to modify, and choose Actions, Modify Reserved Instances.

   Note
   If your Reserved Instances are not in the active state or cannot be modified, Modify Reserved Instances is disabled.

3. The first entry in the modification table displays attributes of the 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.

   • **Scope**: Choose whether the configuration applies to an Availability Zone or to the whole Region.
   • **Availability Zone**: Choose the required Availability Zone. Not applicable for regional Reserved Instances.
   • **Count**: Specify the number of instances. To split the Reserved Instances into multiple configurations, reduce the count, choose Add, and specify a count for the additional configuration. For example, if you have a single configuration with a count of 10, you can change its count to 6 and add a configuration with a count of 4. This process retires the original Reserved Instance after the new Reserved Instances are activated.

4. Choose Continue.
5. To confirm your modification choices when you finish specifying your target configurations, choose Submit modifications.
6. You can determine the status of your modification request by looking at the State column in the Reserved Instances screen. The following are the possible states.

   • **active (pending modification)** — Transition state for original Reserved Instances
   • **retired (pending modification)** — Transition state for original Reserved Instances while new Reserved Instances are being created
   • **retired** — Reserved Instances successfully modified and replaced
   • **active** — One of the following:
     • New Reserved Instances created from a successful modification request
     • Original Reserved Instances after a failed modification request
Old console

To modify your Reserved Instances using the AWS Management Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the Reserved Instances page, select one or more Reserved Instances to modify, and choose Actions, Modify Reserved Instances.

   Note
   If your Reserved Instances are not in the active state or cannot be modified, Modify Reserved Instances is disabled.

3. The first entry in the modification table displays attributes of selected Reserved Instances, and at least one target configuration beneath it. The Units column displays the total instance size footprint. Choose Add for each new configuration to add. Modify the attributes as needed for each configuration, and then choose Continue:
   - **Scope**: Choose whether the configuration applies to an Availability Zone or to the whole Region.
   - **Availability Zone**: Choose the required Availability Zone. Not applicable for regional Reserved Instances.
   - **Count**: Specify the number of instances. To split the Reserved Instances into multiple configurations, reduce the count, choose Add, and specify a count for the additional configuration. For example, if you have a single configuration with a count of 10, you can change its count to 6 and add a configuration with a count of 4. This process retires the original Reserved Instance after the new Reserved Instances are activated.

4. To confirm your modification choices when you finish specifying your target configurations, choose Submit Modifications.

5. You can determine the status of your modification request by looking at the State column in the Reserved Instances screen. The following are the possible states.
   - **active (pending modification)** — Transition state for original Reserved Instances
   - **retired (pending modification)** — Transition state for original Reserved Instances while new Reserved Instances are being created
   - **retired** — Reserved Instances successfully modified and replaced
   - **active** — One of the following:
     - New Reserved Instances created from a successful modification request
     - Original Reserved Instances after a failed modification request

To modify your Reserved Instances using the command line

1. To modify your Reserved Instances, you can use one of the following commands:
   - `modify-reserved-instances` (AWS CLI)
   - `Edit-EC2ReservedInstance` (AWS Tools for Windows PowerShell)

2. To get the status of your modification request (processing, fulfilled, or failed), use one of the following commands:
   - `describe-reserved-instances-modifications` (AWS CLI)
   - `Get-EC2ReservedInstancesModification` (AWS Tools for Windows PowerShell)

Troubleshoot 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. 274) 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.

**Exchange 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 new Convertible Reserved Instance is of an equal or higher value than the original 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 new Convertible Reserved Instance. Amazon EC2 calculates the number of Reserved Instances that you can receive as a result of the exchange.

You can't exchange Standard Reserved Instances, but you can modify them. For more information, see [Modify Reserved Instances](p. 273).

**Contents**

- Requirements for exchanging Convertible Reserved Instances (p. 277)
- Calculate Convertible Reserved Instances exchanges (p. 279)
- Merge Convertible Reserved Instances (p. 279)
- Exchange a portion of a Convertible Reserved Instance (p. 280)
- Submit exchange requests (p. 280)

**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 Exchange a portion of a Convertible Reserved Instance (p. 280). For more information about modifying your Reserved Instances, see Modify Reserved Instances (p. 273).
- 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 Merge Convertible Reserved Instances (p. 279).
- After you exchange a Convertible Reserved Instance, the original reservation is retired. Its end date is the start date of the new reservation, and the end date of the new reservation is the same as the end date of the original Convertible Reserved Instance. For example, if you modify a three-year reservation that had 16 months left in its term, the resulting modified reservation is a 16-month reservation with the same end date as the original one.
Calculate Convertible Reserved Instances exchanges

Exchanging Convertible Reserved Instances is free. However, you might be required to pay a true-up cost, which is a prorated upfront cost of the difference between the original Convertible Reserved Instances that you had and the new 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.

\[
\frac{35}{10} = 3.5
\]

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:

\[3.5 \times 3 \text{ whole Convertible Reserved Instances} + 1 \text{ additional Convertible Reserved Instance}\]

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 new reservation would normally cost $600 on a prorated basis, you are charged $100.

\[\$600 \text{ prorated upfront cost of new reservations} - \$500 \text{ remaining upfront cost of original reservations} = \$100 \text{ difference}\]

Merge 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:

<table>
<thead>
<tr>
<th>Reserved Instance ID</th>
<th>Term</th>
<th>Expiration date</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa1111</td>
<td>1-year</td>
<td>2018-12-31</td>
</tr>
<tr>
<td>bbbb2222</td>
<td>1-year</td>
<td>2018-07-31</td>
</tr>
<tr>
<td>cccc3333</td>
<td>3-year</td>
<td>2018-06-30</td>
</tr>
<tr>
<td>dddd4444</td>
<td>3-year</td>
<td>2019-12-31</td>
</tr>
</tbody>
</table>

- 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.

**Exchange 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.

**Submit exchange requests**

You can exchange your Convertible Reserved Instances using the Amazon EC2 console or a command line tool.

**Exchange a Convertible Reserved Instance using the console**

You can search for Convertible Reserved Instances offerings and select your new configuration from the choices provided.

New console

**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, and choose Find offering.
4. Select a new Convertible Reserved Instance. At the bottom of the screen, you can view the number of Reserved Instances that you receive for the exchange, and any additional costs.
5. When you have selected a Convertible Reserved Instance that meets your needs, choose Review.
6. Choose Exchange, and then Close.
Old console

**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, 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.

**Exchange a Convertible Reserved Instance using the command line interface**

To exchange a Convertible Reserved Instance, first find a new 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**

With Scheduled Reserved Instances, you can reserve capacity that is scheduled to recur daily, weekly, or monthly, with a specified start time and duration, for a one-year term. After you complete your purchase, the instances are available to launch during the time windows that you specified.

**Important**

You cannot purchase Scheduled Reserved Instances at this time. AWS does not have any capacity available for Scheduled Reserved Instances or any plans to make it available in the future. To reserve capacity, use [On-Demand Capacity Reservations](p. 366) instead. For discounted rates, use [Savings Plans](p. 366).
Spot Instances

A Spot Instance is an instance that uses spare EC2 capacity that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance is called a Spot price. The Spot price of each instance type in each Availability Zone is set by Amazon EC2, and is adjusted gradually based on the long-term supply of and demand for Spot Instances. Your Spot Instance runs whenever capacity is available and the maximum price per hour for your request exceeds the Spot price.

Spot Instances are a cost-effective choice if you can be flexible about when your applications run and if your applications can be interrupted. For example, Spot Instances are well-suited for data analysis, batch jobs, background processing, and optional tasks. For more information, see Amazon EC2 Spot Instances.

Key differences between Spot Instances and On-Demand Instances

The following table lists the key differences between Spot Instances and On-Demand Instances.

<table>
<thead>
<tr>
<th>Spot Instances</th>
<th>On-Demand Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch time</td>
<td>Can only be launched immediately if the Spot Instance request is active and capacity is available.</td>
</tr>
<tr>
<td></td>
<td>Can only be launched immediately if you make a manual launch request and capacity is available.</td>
</tr>
</tbody>
</table>
Available capacity

If capacity is not available, the Spot Instance 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.

Rebalance recommendation

The signal that Amazon EC2 emits for a running Spot Instance when the instance is at an elevated risk of interruption.

You determine when an On-Demand Instance is interrupted (stopped, hibernated, or terminated).

Instance interruption

You can stop and start an Amazon EBS-backed Spot Instance. In addition, the Amazon EC2 Spot service can interrupt (p. 317) an individual Spot Instance if capacity is no longer available, the Spot price exceeds your maximum price, or demand for Spot Instances increases.

You determine when an On-Demand Instance is interrupted (stopped, hibernated, or terminated).

### Strategies for using Spot Instances

One strategy is to maintain a minimum level of guaranteed compute resources for your applications by launching a core group of On-Demand Instances, and supplementing them with Spot Instances when the opportunity arises.

### Compare On-Demand and Spot Instances

### 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

- Set up to use Amazon EC2 (p. 5)
- Tutorial: Get started with Amazon EC2 Windows instances (p. 9)

### Spot basics

- How Spot Instances work (p. 287)
Working with Spot Instances

- Prepare for interruptions (p. 321)
- Create a Spot Instance request (p. 295)
- Get request status information (p. 312)

Related services

You can provision Spot Instances directly using Amazon EC2. You can also provision Spot Instances using other services in AWS. For more information, see the following documentation.

Amazon EC2 Auto Scaling and Spot Instances

You can create launch templates or configurations with the maximum price that you are willing to pay, so that Amazon EC2 Auto Scaling can launch Spot Instances. For more information, see Requesting Spot Instances for fault-tolerant and flexible applications and Auto Scaling groups with multiple instance types and purchase options in the Amazon EC2 Auto Scaling User Guide.

Amazon EMR and Spot Instances

There are scenarios where it can be useful to run Spot Instances in an Amazon EMR cluster. For more information, see Spot Instances and When Should You Use Spot Instances in the Amazon EMR Management Guide.

AWS CloudFormation templates

AWS CloudFormation enables you to create and manage a collection of AWS resources using a template in JSON format. AWS CloudFormation templates can include the maximum price you are willing to pay. For more information, see EC2 Spot Instance Updates - Auto Scaling and CloudFormation Integration.

AWS SDK for Java

You can use the Java programming language to manage your Spot Instances. For more information, see Tutorial: Amazon EC2 Spot Instances and Tutorial: Advanced Amazon EC2 Spot Request Management.

AWS SDK for .NET

You can use the .NET programming environment to manage your Spot Instances. For more information, see Tutorial: Amazon EC2 Spot Instances.

Pricing and savings

You pay the Spot price for Spot Instances, which is set by Amazon EC2 and adjusted gradually based on the long-term supply of and demand for Spot Instances. If the maximum price for your request exceeds the current Spot price, Amazon EC2 fulfills your request if capacity is available. Your Spot Instances run until you terminate them, capacity is no longer available, the Spot price exceeds your maximum price, or your Amazon EC2 Auto Scaling group terminates them during scale in.

If you or Amazon EC2 interrupts a running Spot Instance, you are charged for the seconds used or the full hour, or you receive no charge, depending on the operating system used and who interrupted the Spot Instance. For more information, see Billing for interrupted Spot Instances (p. 324).

View prices

To view the current (updated every five minutes) lowest Spot price per AWS Region and instance type, see the Amazon EC2 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. 289).

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. 290).

**View billing**

Your bill provides details about your service usage. For more information, see Viewing your bill in the AWS Billing and Cost Management User Guide.

**Best practices for EC2 Spot**

Amazon EC2 Spot Instances are spare EC2 compute capacity in the AWS Cloud that are available to you at savings of up to 90% off compared to On-Demand prices. The only difference between On-Demand Instances and Spot Instances is that Spot Instances can be interrupted by Amazon EC2, with two minutes of notification, when Amazon EC2 needs the capacity back.

Spot Instances are recommended for stateless, fault-tolerant, flexible applications. For example, Spot Instances work well for big data, containerized workloads, CI/CD, stateless web servers, high performance computing (HPC), and rendering workloads.

While running, Spot Instances are exactly the same as On-Demand Instances. However, Spot does not guarantee that you can keep your running instances long enough to finish your workloads. Spot also does not guarantee that you can get immediate availability of the instances that you are looking for, or that you can always get the aggregate capacity that you requested. Moreover, Spot Instance interruptions and capacity can change over time because Spot Instance availability varies based on supply and demand, and past performance isn’t a guarantee of future results.

Spot Instances are not suitable for workloads that are inflexible, stateful, fault-intolerant, or tightly coupled between instance nodes. It's also not recommended for workloads that are intolerant of occasional periods when the target capacity is not completely available. We strongly warn against using Spot Instances for these workloads or attempting to fail-over to On-Demand Instances to handle interruptions.

Regardless of whether you're an experienced Spot user or new to Spot Instances, if you are currently experiencing issues with Spot Instance interruptions or availability, we recommend that you follow these best practices to have the best experience using the Spot service.

**Spot best practices**

- Prepare individual instances for interruptions (p. 286)
- Be flexible about instance types and Availability Zones (p. 286)
- Use EC2 Auto Scaling groups or Spot Fleet to manage your aggregate capacity (p. 286)
- Use the capacity optimized allocation strategy (p. 286)
- Use proactive capacity rebalancing (p. 287)
- Use integrated AWS services to manage your Spot Instances (p. 287)
Prepare individual instances for interruptions

The best way for you to gracefully handle Spot Instance interruptions is to architect your application to be fault-tolerant. To accomplish this, you can take advantage of EC2 instance rebalance recommendations and Spot Instance interruption notices.

An EC2 Instance rebalance recommendation is a new signal that notifies you when a Spot Instance is at elevated risk of interruption. The signal gives you the opportunity to proactively manage the Spot Instance in advance of the two-minute Spot Instance interruption notice. You can decide to rebalance your workload to new or existing Spot Instances that are not at an elevated risk of interruption. We've made it easy for you to use this new signal by using the Capacity Rebalancing feature in Auto Scaling groups and Spot Fleet. For more information, see Use proactive capacity rebalancing (p. 287).

A Spot Instance interruption notice is a warning that is issued two minutes before Amazon EC2 interrupts a Spot Instance. If your workload is "time-flexible," you can configure your Spot Instances to be stopped or hibernated, instead of being terminated, when they are interrupted. Amazon EC2 automatically stops or hibernates your Spot Instances on interruption, and automatically resumes the instances when we have available capacity.

We recommend that you create a rule in Amazon EventBridge that captures the rebalance recommendations and interruption notifications, and then triggers a checkpoint for the progress of your workload or gracefully handles the interruption. For more information, see Monitor rebalance recommendation signals (p. 315). For a detailed example that walks you through how to create and use event rules, see Taking Advantage of Amazon EC2 Spot Instance Interruption Notices.

For more information, see EC2 instance rebalance recommendations (p. 314) and Spot Instance interruptions (p. 317).

Be flexible about instance types and Availability Zones

A Spot capacity pool is a set of unused EC2 instances with the same instance type (for example, m5.large) and Availability Zone (for example, us-east-1a). You should be flexible about which instance types you request and in which Availability Zones you can deploy your workload. This gives Spot a better chance to find and allocate your required amount of compute capacity. For example, don't just ask for c5.large if you'd be willing to use larges from the c4, m5, and m4 families.

Depending on your specific needs, you can evaluate which instance types you can be flexible across to fulfill your compute requirements. If a workload can be vertically scaled, you should include larger instance types (more vCPUs and memory) in your requests. If you can only scale horizontally, you should include older generation instance types because they are less in demand from On-Demand customers.

A good rule of thumb is to be flexible across at least 10 instance types for each workload. In addition, make sure that all Availability Zones are configured for use in your VPC and selected for your workload.

Use EC2 Auto Scaling groups or Spot Fleet to manage your aggregate capacity

Spot enables you to think in terms of aggregate capacity—in units that include vCPUs, memory, storage, or network throughput—rather than thinking in terms of individual instances. Auto Scaling groups and Spot Fleet enable you to launch and maintain a target capacity, and to automatically request resources to replace any that are disrupted or manually terminated. When you configure an Auto Scaling group or a Spot Fleet, you need only specify the instance types and target capacity based on your application needs. For more information, see Auto Scaling groups in the Amazon EC2 Auto Scaling User Guide and Create a Spot Fleet request (p. 725) in this user guide.

Use the capacity optimized allocation strategy

Allocation strategies in Auto Scaling groups help you to provision your target capacity without the need to manually look for the Spot capacity pools with spare capacity. We recommend using the capacity
optimized strategy because this strategy automatically provisions instances from the most-available Spot capacity pools. You can also take advantage of the capacity optimized allocation strategy in Spot Fleet. Because your Spot Instance capacity is sourced from pools with optimal capacity, this decreases the possibility that your Spot Instances are reclaimed. For more information about allocation strategies, see Spot Instances in the Amazon EC2 Auto Scaling User Guide and Configure Spot Fleet for capacity optimization (p. 714) in this user guide.

Use proactive capacity rebalancing

Capacity Rebalancing helps you maintain workload availability by proactively augmenting your fleet with a new Spot Instance before a running Spot Instance receives the two-minute Spot Instance interruption notice. When Capacity Rebalancing is enabled, Auto Scaling or Spot Fleet attempts to proactively replace Spot Instances that have received a rebalance recommendation, providing the opportunity to rebalance your workload to new Spot Instances that are not at elevated risk of interruption.

Capacity Rebalancing complements the capacity optimized allocation strategy (which is designed to help find the most optimal spare capacity) and the mixed instances policy (which is designed to enhance availability by deploying instances across multiple instance types running in multiple Availability Zones).

For more information, see Capacity Rebalancing (p. 714).

Use integrated AWS services to manage your Spot Instances

Other AWS services integrate with Spot to reduce overall compute costs without the need to manage the individual instances or fleets. We recommend that you consider the following solutions for your applicable workloads: Amazon EMR, Amazon ECS, AWS Batch, Amazon EKS, SageMaker, AWS Elastic Beanstalk, and Amazon GameLift. To learn more about Spot best practices with these services, see the Amazon EC2 Spot Instances Workshops Website.

How Spot Instances work

To launch a Spot Instance, either you create a Spot Instance request, or Amazon EC2 creates a Spot Instance request on your behalf. The Spot Instance launches when the Spot Instance request is fulfilled.

You can launch a Spot Instance using several different services. For more information, see Getting Started with Amazon EC2 Spot Instances. In this user guide, we describe the following ways to launch a Spot Instance using EC2:

- You can create a Spot Instance request. For more information, see Create a Spot Instance request (p. 295).
- You can create an EC2 Fleet, in which you specify the desired number of Spot Instances. Amazon EC2 creates a Spot Instance request on your behalf for every Spot Instance that is specified in the EC2 Fleet. For more information, see Create an EC2 Fleet (p. 701).
- You can create a Spot Fleet request, in which you specify the desired number of Spot Instances. Amazon EC2 creates a Spot Instance request on your behalf for every Spot Instance that is specified in the Spot Fleet request. For more information, see Create a Spot Fleet request (p. 725).

The Spot Instance request must include the maximum price that you're willing to pay per hour per instance. If you don't specify a price, the price defaults to the On-Demand price. The request can include other constraints such as the instance type and Availability Zone.

Your Spot Instance launches if the maximum price that you're willing to pay exceeds the Spot price, and if there is available capacity. If the maximum price you're willing to pay is lower than the Spot price, then your instance does not launch. However, because Amazon EC2 gradually adjusts the Spot price based on the long-term supply of and demand for Spot Instances, the maximum price you're willing to pay might eventually exceed the Spot price, in which case your instance will launch.
Your Spot Instance runs until you stop or terminate it, or until Amazon EC2 interrupts it (known as a Spot Instance interruption).

When you use Spot Instances, you must be prepared for interruptions. Amazon EC2 can interrupt your Spot Instance when the demand for Spot Instances rises, when the supply of Spot Instances decreases, or when the Spot price exceeds your maximum price. 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. 317).

You can stop, start, reboot, or terminate an Amazon EBS-backed Spot Instance. The Spot service can stop, terminate, or hibernate a Spot Instance when it interrupts it.

Contents

- Launch Spot Instances in a launch group (p. 288)
- Launch Spot Instances in an Availability Zone group (p. 288)
- Launch Spot Instances in a VPC (p. 289)

Launch 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.

Launch 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.
Launch 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.

Spot Instance pricing history

Spot Instance prices are set by Amazon EC2 and adjust gradually based on long-term trends in supply and demand for Spot Instance capacity.

When you request Spot Instances, we recommend that you use the default maximum price (the On-Demand price). When your request is fulfilled, your Spot Instances launch at the current Spot price, not exceeding the On-Demand price. If you want to specify a maximum price, we recommend that you first review the Spot price history. You can view the Spot price history for the last 90 days, filtering by instance type, operating system, and Availability Zone.

To view the current Spot prices

For the current Spot Instance prices, see Amazon EC2 Spot Instances Pricing.

To view the Spot price history (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Spot Requests.
3. Choose Pricing history.
4. For Graph, choose to compare the price history by Availability Zones or by Instance Types.
   - If you choose Availability Zones, then choose the Instance type, operating system (Platform), and Date range for which to view the price history.
   - If you choose Instance Types, then choose up to five Instance type(s), the Availability Zone, operating system (Platform), and Date range for which to view the price history.

The following screenshot shows a price comparison for different instance types.
5. Move your pointer over the graph to display the prices at specific times in the selected date range. The prices are displayed in the information blocks above the graph. The price displayed in the top row shows the price on a specific date. The price displayed in the second row shows the average price over the selected date range.

6. To display the price per vCPU, toggle on Display normalized prices. To display the price for the instance type, toggle off Display normalized prices.

**To view the Spot price history using the command line**

You can use one of the following commands. For more information, see Access 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 **Savings** section shows the Spot usage and savings information for a Spot Fleet.

<table>
<thead>
<tr>
<th>Spot usage and savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Spot Instances</td>
</tr>
<tr>
<td>266 vCPU-hours</td>
</tr>
<tr>
<td>700 Mem(GiB)-hours</td>
</tr>
<tr>
<td>$9.55 On-Demand total</td>
</tr>
<tr>
<td>$2.99 Spot total</td>
</tr>
<tr>
<td>69% Savings</td>
</tr>
</tbody>
</table>

**Details**

<table>
<thead>
<tr>
<th>Instance Type</th>
<th>vCPU-hours</th>
<th>Mem(GiB)-hours</th>
<th>Cost</th>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>t3.medium (1)</td>
<td>2</td>
<td>4</td>
<td>$0.01 total</td>
<td>70% savings</td>
</tr>
<tr>
<td>m4.large (1)</td>
<td>144</td>
<td>576</td>
<td>$2.52 total</td>
<td>68% savings</td>
</tr>
<tr>
<td>t2.micro (2)</td>
<td>120</td>
<td>120</td>
<td>$0.46 total</td>
<td>70% savings</td>
</tr>
</tbody>
</table>

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: \[
\text{Average cost per vCPU-hour} = \frac{\text{Spot total}}{\text{vCPU-hours}}
\]
• **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: \( \text{Average cost per mem(GiB)-hour} = \frac{\text{Spot total}}{\text{Mem(GiB)-hours}}. \)

• **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 the ID of a Spot Fleet request and scroll to the **Savings** section.

   Alternatively, select the check box next to the Spot Fleet request ID and choose the **Savings** tab.
4. By default, the page displays usage and savings information for the last three days. You can choose **last hour** or the **last three days**. For Spot Fleets that were launched less than an hour ago, the page shows the estimated savings for the hour.

**To view the savings information for all running Spot Instances (console)**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, choose **Spot Requests**.
3. Choose **Savings summary**.

**Spot Instance requests**

To use Spot Instances, you create a Spot Instance request that includes the desired number of instances, the instance type, the Availability Zone, and the maximum price that you are willing to pay per instance hour. If your maximum price exceeds the current Spot price, Amazon EC2 fulfills your request immediately if capacity is available. Otherwise, Amazon EC2 waits until your request can be fulfilled or until you cancel the request.

The following illustration shows how Spot Instance requests work. Notice that the request type (one-time or persistent) determines whether the request is opened again when Amazon EC2 interrupts a Spot Instance or if you stop a Spot Instance. If the request is persistent, the request is opened again after your Spot Instance is interrupted. If the request is persistent and you stop your Spot Instance, the request only opens after you start your Spot Instance.
Spot Instance request states

A Spot Instance request can be in one of the following states:

- **open** – The request is waiting to be fulfilled.
- **active** – The request is fulfilled and has an associated Spot Instance.
- **failed** – The request has one or more bad parameters.
- **closed** – The Spot Instance was interrupted or terminated.
- **disabled** – You stopped the Spot Instance.
- **cancelled** – You canceled the request, or the request expired.
The following illustration represents the transitions between the request states. Notice that the transitions depend on the request type (one-time or persistent).

A one-time Spot Instance request remains active until Amazon EC2 launches the Spot Instance, the request expires, or you cancel the request. If the Spot price exceeds your maximum price or capacity is not available, your Spot Instance is terminated and the Spot Instance request is closed.

A persistent Spot Instance request remains active until it expires or you cancel it, even if the request is fulfilled. If the Spot price exceeds your maximum price or capacity is not available, your Spot Instance is interrupted. After your instance is interrupted, when your maximum price exceeds the Spot price or capacity becomes available again, the Spot Instance is started if stopped or resumed if hibernated. You can stop a Spot Instance and start it again if capacity is available and your maximum price exceeds the current Spot price. If the Spot Instance is terminated (irrespective of whether the Spot Instance is in a stopped or running state), the Spot Instance request is opened again and Amazon EC2 launches a new Spot Instance. For more information, see Stop a Spot Instance (p. 304), Start a Spot Instance (p. 305), and Terminate a Spot Instance (p. 306).

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. 309).

**Define a duration for your Spot Instances**

Spot Instances with a defined duration (also known as Spot blocks) are no longer available to new customers from July 1, 2021. For customers who have previously used the feature, we will continue to support Spot Instances with a defined duration until December 31, 2022.

**Specify 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. 360) 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 Create a Spot Instance request (p. 295).
• Request a Spot Instance in a VPC with an instance tenancy of dedicated. For more information, see Create a VPC with an instance tenancy of dedicated (p. 363). You cannot request a Spot Instance with a tenancy of default if you request it in a VPC with an instance tenancy of dedicated.

All instance families support Dedicated Spot Instances except T instances. For each supported instance family, only the largest instance size or metal size supports Dedicated Spot Instances.

Service-linked role for Spot Instance requests

Amazon EC2 uses service-linked roles for the permissions that it requires to call other AWS services on your behalf. A service-linked role is a unique type of IAM role that is linked directly to an AWS service. Service-linked roles provide a secure way to delegate permissions to AWS services because only the linked service can assume a service-linked role. For more information, see Using Service-Linked Roles in the IAM User Guide.

Amazon EC2 uses the service-linked role named AWSServiceRoleForEC2Spot to launch and manage Spot Instances on your behalf.

Permissions granted by AWSServiceRoleForEC2Spot

Amazon EC2 uses AWSServiceRoleForEC2Spot to complete the following actions:

- ec2:DescribeInstances – Describe Spot Instances
- ec2:StopInstances – Stop Spot Instances
- ec2:StartInstances – Start Spot Instances

Create the service-linked role

Under most circumstances, you don't need to manually create a service-linked role. Amazon EC2 creates the AWSServiceRoleForEC2Spot service-linked role the first time you request a Spot Instance using the console.

If you had an active Spot Instance request before October 2017, when Amazon EC2 began supporting this service-linked role, Amazon EC2 created the AWSServiceRoleForEC2Spot role in your AWS account. For more information, see A New Role Appeared in My Account in the IAM User Guide.

If you use the AWS CLI or an API to request a Spot Instance, you must first ensure that this role exists.

To create AWSServiceRoleForEC2Spot using the 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.

To create AWSServiceRoleForEC2Spot using the AWS CLI

Use the create-service-linked-role command as follows.

aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
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.

**Grant access to customer managed keys for use with encrypted AMIs and EBS snapshots**

If you specify an encrypted AMI (p. 127) or an encrypted Amazon EBS snapshot (p. 132) for your Spot Instances and you use a customer managed key for encryption, you must grant the `AWSServiceRoleForEC2Spot` role permission to use the customer managed key so that Amazon EC2 can launch Spot Instances on your behalf. To do this, you must add a grant to the customer managed key, as shown in the following procedure.

When providing permissions, grants are an alternative to key policies. For more information, see Using Grants and Using Key Policies in AWS KMS in the *AWS Key Management Service Developer Guide*.

**To grant the AWSServiceRoleForEC2Spot role permissions to use the customer managed key**

- Use the `create-grant` command to add a grant to the customer managed key and to specify the principal (the `AWSServiceRoleForEC2Spot` service-linked role) that is given permission to perform the operations that the grant permits. The customer managed key is specified by the `key-id` parameter and the ARN of the customer managed key. The principal is specified by the `grantee-principal` parameter and the ARN of the `AWSServiceRoleForEC2Spot` service-linked role.

```
aws kms create-grant
   --region us-east-1
   --key-id arn:aws:kms:us-east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab
   --grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2Spot
   --operations "Decrypt" "Encrypt" "GenerateDataKey"
   "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
   "ReEncryptTo"
```

**Create a Spot Instance request**

The procedure for requesting a Spot Instance is similar to the procedure for launching an On-Demand Instance. You can request a Spot Instance in the following ways:

- To request a Spot Instance using the console, use the launch instance wizard. For more information, see To create a Spot Instance request (console) (p. 296).
- To request a Spot Instance using the CLI, use the `request-spot-instances` command or the `run-instances` command. For more information, see To create a Spot Instance request using request-spot-instances (CLI) and To create a Spot Instance request using run-instances (CLI).

After you've submitted your Spot Instance request, you can't change the parameters of the request. This means that you can't make changes to the maximum price that you're willing to pay.

If you request multiple Spot Instances at one time, Amazon EC2 creates separate Spot Instance requests so that you can track the status of each request separately. For more information about tracking Spot Instance requests, see Spot request status (p. 309).

To launch a fleet that includes Spot Instances and On-Demand Instances, see Create a Spot Fleet request (p. 725).

**Note**

You can't launch a Spot Instance and an On-Demand Instance in the same call using the launch instance wizard or the `run-instances` command.

**Prerequisites**
Before you begin, decide on your maximum price, how many Spot Instances you'd like, and what instance type to use. To review Spot price trends, see Spot Instance pricing history (p. 289).

To create a Spot Instance request (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation bar at the top of the screen, select a Region.
3. From the Amazon EC2 console dashboard, choose Launch Instance.
4. On the Choose an Amazon Machine Image (AMI) page, choose an AMI. For more information, see Step 1: Choose an Amazon Machine Image (AMI) (p. 392).
5. On the Choose an Instance Type page, select the hardware configuration and size of the instance to launch, and then choose Next: Configure Instance Details. For more information, see Step 2: Choose an Instance Type (p. 393).
6. On the Configure Instance Details page, configure the Spot Instance request as follows:

   • **Number of instances**: Enter the number of instances to launch.
     - **Note**: Amazon EC2 creates a separate request for each Spot Instance.
   • (Optional) To help ensure that you maintain the correct number of instances to handle demand on your application, you can choose Launch into Auto Scaling Group to create a launch configuration and an Auto Scaling group. Auto Scaling scales the number of instances in the group according to your specifications. For more information, see the Amazon EC2 Auto Scaling User Guide.
   • **Purchasing option**: Choose Request Spot instances to launch a Spot Instance. When you choose this option, the following fields appear.
   • **Current price**: The current Spot price in each Availability Zone is displayed for the instance type that you selected.
   • (Optional) **Maximum price**: You can leave the field empty, or you can specify the maximum amount you're willing to pay.
     - If you leave the field empty, then the maximum price defaults to the current On-Demand price.
     - Your Spot Instance launches at the current Spot price, not exceeding the On-Demand price.
     - If you specify a maximum price that is more than the current Spot Price, your Spot Instance launches and is charged at the current Spot price.
     - If you specify a maximum price that is lower than the Spot price, your Spot Instance is not launched.
   • **Persistent request**: Choose Persistent request to resubmit the Spot Instance request if your Spot Instance is interrupted.
   • **Interruption behavior**: By default, the Spot service terminates a Spot Instance when it is interrupted. If you choose Persistent request, you can then specify that the Spot service stops or hibernates your Spot Instance when it's interrupted. For more information, see Interruption behaviors (p. 318).
   • (Optional) **Request valid to**: Choose Edit to specify when the Spot Instance request expires.

   For more information about configuring your Spot Instance, see Step 3: Configure Instance Details (p. 394).

7. The AMI you selected includes one or more volumes of storage, including the root device volume. On the Add Storage page, you can specify additional volumes to attach to the instance by choosing Add New Volume. For more information, see Step 4: Add Storage (p. 396).
8. On the Add Tags page, specify tags (p. 1450) by providing key and value combinations. For more information, see Step 5: Add Tags (p. 396).
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. 1135).) Select or create a security group, and then choose Review and Launch. For more information, see Step 6: Configure Security Group (p. 397).

10. On the Review Instance Launch page, check the details of your instance, and make any necessary changes by choosing the appropriate Edit link. When you are ready, choose Launch. For more information, see Step 7: Review Instance Launch and Select Key Pair (p. 397).

11. In the Select an existing key pair or create a new key pair dialog box, you can choose an existing key pair, or create a new one. For example, choose Choose an existing key pair, then select the key pair that you created when getting set up. For more information, see Amazon EC2 key pairs and Windows instances (p. 1127).

Important
If you choose the Proceed without key pair option, you won't be able to connect to the instance unless you choose an AMI that is configured to allow users another way to log in.

12. To launch your instance, select the acknowledgment check box, then choose Launch Instances.

If the instance fails to launch or the state immediately goes to terminated instead of running, see Troubleshoot instance launch issues (p. 1465).

To create a Spot Instance request using request-spot-instances (AWS CLI)

Use the request-spot-instances command to create a one-time request.

```bash
aws ec2 request-spot-instances \
  --instance-count 5 \
  --type "one-time" \
  --launch-specification file://specification.json
```

Use the request-spot-instances command to create a persistent request.

```bash
aws ec2 request-spot-instances \
  --instance-count 5 \
  --type "persistent" \
  --launch-specification file://specification.json
```

For example launch specification files to use with these commands, see Spot Instance request example launch specifications (p. 307). If you download a launch specification file from the console, you must use the request-spot-fleet command instead (the console specifies a Spot Instance request using a Spot Fleet).

To create a Spot Instance request using run-instances (AWS CLI)

Use the run-instances command and specify the Spot Instance options in the --instance-market-options parameter.

```bash
aws ec2 run-instances \
  --image-id ami-0abcdef1234567890 \
  --instance-type t2.micro \
  --count 5 \
  --subnet-id subnet-08fc749671b2d077c \
  --key-name MyKeyPair \
  --security-group-ids sg-0b0384b66d7d92f9 \
  --instance-market-options file://spot-options.json
```

The following is the data structure to specify in the JSON file for --instance-market-options. You can also specify ValidUntil and InstanceInterruptionBehavior. If you do not specify a field in the data structure, the default value is used. This example creates a one-time request and specifies 0.02 as the maximum price you're willing to pay for the Spot Instance.
Find running Spot Instances

Amazon EC2 launches a Spot Instance when the maximum price exceeds the Spot price and capacity is available. A Spot Instance runs until it is interrupted or you terminate it yourself. If your maximum price is exactly equal to the Spot price, there is a chance that your Spot Instance remains running, depending on demand.

To find running Spot Instances (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Spot Requests. You can see both Spot Instance requests and Spot Fleet requests. If a Spot Instance request has been fulfilled, Capacity is the ID of the Spot Instance. For a Spot Fleet, Capacity indicates how much of the requested capacity has been fulfilled. To view the IDs of the instances in a Spot Fleet, choose the expand arrow, or select the fleet and choose Instances.

   Note
   For Spot Instance requests that are created by a Spot Fleet, the requests are not tagged instantly with the system tag that indicates the Spot Fleet to which they belong, and for a period of time may appear separate from Spot Fleet request.

   Alternatively, in the navigation pane, choose Instances. In the top right corner, choose the settings icon (⚙️), and then under Attribute columns, select Instance lifecycle. For each instance, Instance lifecycle is either normal, spot, or scheduled.

To find running Spot Instances (AWS CLI)

To enumerate your Spot Instances, use the describe-spot-instance-requests command with the --query option.

```bash
aws ec2 describe-spot-instance-requests \
   --query "SpotInstanceRequests[*].{ID:InstanceId}"
```

The following is example output:

```json
[
  {
    "ID": "i-1234567890abcdef0"
  },
  {
    "ID": "i-0598c7d356eba48d7"
  }
]
```

Alternatively, you can enumerate your Spot Instances using the describe-instances command with the --filters option.

```bash
aws ec2 describe-instances \
   --filters "Name=instance-lifecycle,Values=spot"
```
To describe a single Spot Instance instance, use the `describe-spot-instance-requests` command with the `--spot-instance-request-ids` option.

```
aws ec2 describe-spot-instance-requests
   --spot-instance-request-ids sir-08b93456
```

**Tag Spot Instance requests**

To help categorize and manage your Spot Instance requests, you can tag them with custom metadata. You can assign a tag to a Spot Instance request when you create it, or afterward. You can assign tags using the Amazon EC2 console or a command line tool.

When you tag a Spot Instance request, the instances and volumes that are launched by the Spot Instance request are not automatically tagged. You need to explicitly tag the instances and volumes launched by the Spot Instance request. You can assign a tag to a Spot Instance and volumes during launch, or afterward.

For more information about how tags work, see Tag your Amazon EC2 resources (p. 1450).

**Contents**

- Prerequisites (p. 299)
- Tag a new Spot Instance request (p. 301)
- Tag an existing Spot Instance request (p. 301)
- View Spot Instance request tags (p. 302)

**Prerequisites**

Grant the IAM user the permission to tag resources. For more information about IAM policies and example policies, see Example: Tag resources (p. 1097).

The IAM policy you create is determined by which method you use for creating a Spot Instance request.

- If you use the launch instance wizard or `run-instances` to request Spot Instances, see To grant an IAM user the permission to tag resources when using the launch instance wizard or run-instances.
- If you use the `request-spot-instances` command to request Spot Instances, see To grant an IAM user the permission to tag resources when using request-spot-instances.

**To grant an IAM user the permission to tag resources when using the launch instance wizard or run-instances**

Create a IAM policy that includes the following:

- The `ec2:RunInstances` action. This grants the IAM user permission to launch an instance.
- For `Resource`, specify `spot-instances-request`. This allows users to create Spot Instance requests, which request Spot Instances.
- The `ec2:CreateTags` action. This grants the IAM user permission to create tags.
- For `Resource`, specify `*`. This allows users to tag all resources that are created during instance launch.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowLaunchInstances",
         "Effect": "Allow",
         "Action": ["ec2:RunInstances"]
      }
   ]
}
```
Note
When you use the RunInstances action to create Spot Instance requests and tag the Spot Instance requests on create, you need to be aware of how Amazon EC2 evaluates the spot-instances-request resource in the RunInstances statement.

The spot-instances-request resource is evaluated in the IAM policy as follows:

- If you don't tag a Spot Instance request on create, Amazon EC2 does not evaluate the spot-instances-request resource in the RunInstances statement.
- If you tag a Spot Instance request on create, Amazon EC2 evaluates the spot-instances-request resource in the RunInstances statement.

Therefore, for the spot-instances-request resource, the following rules apply to the IAM policy:

- If you use RunInstances to create a Spot Instance request and you don't intend to tag the Spot Instance request on create, you don't need to explicitly allow the spot-instances-request resource; the call will succeed.
- If you use RunInstances to create a Spot Instance request and intend to tag the Spot Instance request on create, you must include the spot-instances-request resource in the RunInstances allow statement, otherwise the call will fail.
- If you use RunInstances to create a Spot Instance request and intend to tag the Spot Instance request on create, you must specify the spot-instances-request resource or include a * wildcard in the CreateTags allow statement, otherwise the call will fail.

For example IAM policies, including policies that are not supported for Spot Instance requests, see Work with Spot Instances (p. 1092).

To grant an IAM user the permission to tag resources when using request-spot-instances

Create a IAM policy that includes the following:

- The ec2:RequestSpotInstances action. This grants the IAM user permission to create a Spot Instance request.
- The ec2:CreateTags action. This grants the IAM user permission to create tags.
- For Resource, specify spot-instances-request. This allows users to tag only the Spot Instance request.
Tag a new Spot Instance request

**To tag a new Spot Instance request using the console**

1. Follow the [Create a Spot Instance request (p. 295)](https://aws.amazon.com) procedure.
2. To add a tag, on the **Add Tags** page, choose **Add Tag**, and enter the key and value for the tag. Choose **Add another tag** for each additional tag.

   For each tag, you can tag the Spot Instance request, the Spot Instances, and the volumes with the same tag. To tag all three, ensure that **Instances**, **Volumes**, and **Spot Instance Requests** are selected. To tag only one or two, ensure that the resources you want to tag are selected, and the other resources are cleared.
3. Complete the required fields to create a Spot Instance request, and then choose **Launch**. For more information, see [Create a Spot Instance request (p. 295)](https://aws.amazon.com).

**To tag a new Spot Instance request using the AWS CLI**

To tag a Spot Instance request when you create it, configure the Spot Instance request configuration as follows:

- Specify the tags for the Spot Instance request using the **--tag-specification** parameter.
- For **ResourceType**, specify **spot-instances-request**. If you specify another value, the Spot Instance request will fail.
- For **Tags**, specify the key-value pair. You can specify more than one key-value pair.

In the following example, the Spot Instance request is tagged with two tags: Key=Environment and Value=Production, and Key=Cost-Center and Value=123.

```bash
aws ec2 request-spot-instances
   --instance-count 5
   --type "one-time"
   --launchSpecification file://specification.json
   --tag-specification "ResourceType=spot-instances-request,Tags=[{Key=Environment,Value=Production},{Key=Cost-Center,Value=123}]"
```

Tag an existing Spot Instance request

**To tag an existing Spot Instance request using the console**

After you have created a Spot Instance request, you can add tags to the Spot Instance request using the console.

2. Select your Spot Instance request.
3. Choose the Tags tab and choose Create Tag.

To tag an existing Spot Instance using the console

After your Spot Instance request has launched your Spot Instance, you can add tags to the instance using the console. For more information, see Add and delete tags on an individual resource (p. 1457).

To tag an existing Spot Instance request or Spot Instance using the AWS CLI

Use the create-tags command to tag existing resources. In the following example, the existing Spot Instance request and the Spot Instance are tagged with Key=purpose and Value=test.

```
aws ec2 create-tags \\n    --resources sir-08b93456 i-1234567890abcdef0 \\n    --tags Key=purpose,Value=test
```

View Spot Instance request tags

To view Spot Instance request tags using the console

2. Select your Spot Instance request and choose the Tags tab.

To describe Spot Instance request tags

Use the describe-tags command to view the tags for the specified resource. In the following example, you describe the tags for the specified request.

```
aws ec2 describe-tags \\n    --filters "Name=resource-id,Values=sir-11112222-3333-4444-5555-66666EXAMPLE"
```

You can also view the tags of a Spot Instance request by describing the Spot Instance request.

Use the describe-spot-instance-requests command to view the configuration of the specified Spot Instance request, which includes any tags that were specified for the request.

```
aws ec2 describe-spot-instance-requests \
```
{  
  "SpotInstanceRequests": [  
    {  
      "CreateTime": "2020-06-24T14:22:11+00:00",
      "InstanceId": "i-1234567890EXAMPLE",
      "LaunchSpecification": {  
        "SecurityGroups": [  
          {  
            "GroupName": "launch-wizard-6",
            "GroupId": "sg-1234567890EXAMPLE"
          }
        ],
        "BlockDeviceMappings": [  
          {  
            "DeviceName": "/dev/xvda",
            "Ebs": {  
              "DeleteOnTermination": true,
              "VolumeSize": 8,
              "VolumeType": "gp2"
            }
          }
        ],
        "ImageId": "ami-1234567890EXAMPLE",
        "InstanceType": "t2.micro",
        "KeyName": "my-key-pair",
        "NetworkInterfaces": [  
          {  
            "DeviceIndex": 0,
            "SubnetId": "subnet-11122233"
          }
        ],
        "Placement": {  
          "AvailabilityZone": "eu-west-1c",
          "Tenancy": "default"
        },
        "Monitoring": {  
          "Enabled": false
        }
      },
      "LaunchedAvailabilityZone": "eu-west-1c",
      "ProductDescription": "Linux/UNIX",
      "SpotInstanceRequestId": "sir-1234567890EXAMPLE",
      "SpotPrice": "0.012600",
      "State": "active",
      "Status": {  
        "Code": "fulfilled",
        "Message": "Your spot request is fulfilled.",
        "UpdateTime": "2020-06-25T18:30:21+00:00"
      },
      "Tags": [  
        {  
          "Key": "Environment",
          "Value": "Production"
        },
        {  
          "Key": "Another key",
          "Value": "Another value"
        }
      ],
      "Type": "one-time",
      "InstanceInterruptionBehavior": "terminate"
    }
  ]
}
Cancel a Spot Instance request

If you no longer want your Spot Instance request, you can cancel it. You can only cancel Spot Instance requests that are open, active, or disabled.

- Your Spot Instance request is open when your request has not yet been fulfilled and no instances have been launched.
- Your Spot Instance request is active when your request has been fulfilled and Spot Instances have launched as a result.
- Your Spot Instance request is disabled when you stop your Spot Instance.

If your Spot Instance request is active and has an associated running Spot Instance, canceling the request does not terminate the instance. For more information about terminating a Spot Instance, see Terminate a Spot Instance (p. 306).

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 Instance request.
3. Choose Actions, Cancel request.
4. (Optional) If you are finished with the associated Spot Instances, you can terminate them. In the Cancel Spot request dialog box, select Terminate instances, and then choose Confirm.

To cancel a Spot Instance request (AWS CLI)

- Use the cancel-spot-instance-requests command to cancel the specified Spot Instance request.

```bash
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

Stop a Spot Instance

If you don’t need your Spot Instances now, but you want to restart them later without losing the data persisted in the Amazon EBS volume, you can stop them. The steps for stopping a Spot Instance are similar to the steps for stopping an On-Demand Instance.

Note

While a Spot Instance is stopped, you can modify some of its instance attributes, but not the instance type.
We don’t charge usage for a stopped Spot Instance, or data transfer fees, but we do charge for the storage for any Amazon EBS volumes.

Limitations

- You can only stop a Spot Instance if the Spot Instance was launched from a persistent Spot Instance request.
- You can’t stop a Spot Instance if the associated Spot Instance request is cancelled. When the Spot Instance request is cancelled, you can only terminate the Spot Instance.
- You can’t stop a Spot Instance if it is part of a fleet or launch group, or Availability Zone group.
New console

**To stop a Spot Instance (console)**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and select the Spot Instance.
3. Choose **Instance state, Stop instance**.
4. When prompted for confirmation, choose **Stop**.

Old console

**To stop a Spot Instance (console)**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and select the Spot Instance.
3. Choose **Actions, Instance State, Stop**.

AWS CLI

**To stop a Spot Instance (AWS CLI)**

- Use the `stop-instances` command to manually stop one or more Spot Instances.

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

**Start a Spot Instance**

You can start a Spot Instance that you previously stopped. The steps for starting a Spot Instance are similar to the steps for starting an On-Demand Instance.

**Prerequisites**

You can only start a Spot Instance if:

- You manually stopped the Spot Instance.
- The Spot Instance is an EBS-backed instance.
- Spot Instance capacity is available.
- The Spot price is lower than your maximum price.

**Limitations**

- You can't start a Spot Instance if it is part of fleet or launch group, or Availability Zone group.

New console

**To start a Spot Instance (console)**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and select the Spot Instance.
3. Choose **Instance state, Start instance**.
Old console

**To start a Spot Instance (console)**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Instances** and select the Spot Instance.
3. Choose **Actions**, **Instance State**, **Start**.

AWS CLI

**To start a Spot Instance (AWS CLI)**

- Use the `start-instances` command to manually start one or more Spot Instances.

```bash
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

**Terminate a Spot Instance**

If you terminate a running or stopped Spot Instance that was launched by a persistent Spot Instance request, the Spot Instance request transitions to the **open** state so that a new Spot Instance can be launched. To ensure that no new Spot Instance is launched, you must first cancel the Spot Instance request.

If you cancel an **active** Spot Instance request that has a running Spot Instance, the running Spot Instance is not automatically terminated; you must manually terminate the Spot Instance.

If you cancel a **disabled** Spot Instance request that has a stopped Spot Instance, the stopped Spot Instance is automatically terminated by the Amazon EC2 Spot service. There might be a short lag between when you cancel the Spot Instance request and when the Spot service terminates the Spot Instance.

For information about canceling a Spot Instance request, see [Cancel a Spot Instance request](p. 304).

New console

**To manually terminate a Spot Instance using the console**

1. Before you terminate an instance, verify that you won't lose any data by checking that your Amazon EBS volumes won't be deleted on termination and that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.
2. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
3. In the navigation pane, choose **Instances**.
4. To confirm that the instance is a Spot Instance, check that **spot** appears in the **Instance lifecycle** column.
5. Select the instance, and choose **Actions**, **Instance state**, **Terminate instance**.
6. Choose **Terminate** when prompted for confirmation.

Old console

**To manually terminate a Spot Instance using the console**

1. Before you terminate an instance, verify that you won't lose any data by checking that your Amazon EBS volumes won't be deleted on termination and that you've copied any data that you
need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.

2. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
3. In the navigation pane, choose **Instances**.
4. To confirm that the instance is a Spot Instance, check that **spot** appears in the **Lifecycle** column.
5. Select the instance, and choose **Actions, Instance State, Terminate**.
6. Choose **Yes, Terminate** when prompted for confirmation.

**AWS CLI**

**To manually terminate a Spot Instance using the AWS CLI**

- Use the `terminate-instances` command to manually terminate Spot Instances.

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7
```

**Spot Instance request example launch specifications**

The following examples show launch configurations that you can use with the `request-spot-instances` command to create a Spot Instance request. For more information, see Create a Spot Instance request (p. 295).

1. Launch Spot Instances (p. 307)
2. Launch Spot Instances in the specified Availability Zone (p. 307)
3. Launch Spot Instances in the specified subnet (p. 308)
4. Launch a Dedicated Spot Instance (p. 308)

**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.

```json
{
    "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.

```json
{
    "ImageId": "ami-1a2b3c4d",
    "KeyName": "my-key-pair",
    "SecurityGroupIds": [ "sg-1a2b3c4d" ]
}
```
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.

```json
{
    "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.

```json
{
    "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.

```json
{
    "ImageId": "ami-1a2b3c4d",
    "KeyName": "my-key-pair",
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],
    "InstanceType": "c3.8xlarge",
    "SubnetId": "subnet-1a2b3c4d",
    "Placement": {
        "Tenancy": "dedicated"
    }
}
```
Spot request status

To help you track your Spot Instance requests and plan your use of Spot Instances, use the request status provided by Amazon EC2. For example, the request status can provide the reason why your Spot request isn't fulfilled yet, or list the constraints that are preventing the fulfillment of your Spot request.

At each step of the process—also called the Spot request lifecycle—specific events determine successive request states.

Contents

- Lifecycle of a Spot request (p. 309)
- Get request status information (p. 312)
- Spot request status codes (p. 313)

Lifecycle of a Spot request

The following diagram shows you the paths that your Spot request can follow throughout its lifecycle, from submission to termination. Each step is depicted as a node, and the status code for each node describes the status of the Spot request and Spot Instance.

Pending evaluation

As soon as you create a Spot Instance request, it goes into the pending-evaluation state unless one or more request parameters are not valid (bad-parameters).
### Holding

If one or more request constraints are valid but can't be met yet, or if there is not enough capacity, the request goes into a holding state waiting for the constraints to be met. The request options affect the likelihood of the request being fulfilled. For example, if you specify a maximum price below the current Spot price, your request stays in a holding state until the Spot price goes below your maximum price. If you specify an Availability Zone group, the request stays in a holding state until the Availability Zone constraint is met.

In the event of an outage of one of the Availability Zones, there is a chance that the spare EC2 capacity available for Spot Instance requests in other Availability Zones can be affected.

<table>
<thead>
<tr>
<th>Status code</th>
<th>Request state</th>
<th>Instance state</th>
</tr>
</thead>
<tbody>
<tr>
<td>pending-evaluation</td>
<td>open</td>
<td>n/a</td>
</tr>
<tr>
<td>bad-parameters</td>
<td>closed</td>
<td>n/a</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Status code</th>
<th>Request state</th>
<th>Instance state</th>
</tr>
</thead>
<tbody>
<tr>
<td>schedule-expired</td>
<td>cancelled</td>
<td>n/a</td>
</tr>
<tr>
<td>canceled-before-fulfillment*</td>
<td>cancelled</td>
<td>n/a</td>
</tr>
<tr>
<td>bad-parameters</td>
<td>failed</td>
<td>n/a</td>
</tr>
<tr>
<td>system-error</td>
<td>closed</td>
<td>n/a</td>
</tr>
</tbody>
</table>

* If you cancel the request.

### Pending fulfillment
When the constraints you specified (if any) are met and your maximum price is equal to or higher than the current Spot price, your Spot request goes into the **pending-fulfillment** state.

At this point, Amazon EC2 is getting ready to provision the instances that you requested. If the process stops at this point, it is likely to be because it was canceled by the user before a Spot Instance was launched. It might also be because an unexpected system error occurred.

<table>
<thead>
<tr>
<th>Status code</th>
<th>Request state</th>
<th>Instance state</th>
</tr>
</thead>
<tbody>
<tr>
<td>pending-fulfillment</td>
<td>open</td>
<td>n/a</td>
</tr>
</tbody>
</table>

**Fulfilled**

When all the specifications for your Spot Instances are met, your Spot request is fulfilled. Amazon EC2 launches the Spot Instances, which can take a few minutes. If a Spot Instance is hibernated or stopped when interrupted, it remains in this state until the request can be fulfilled again or the request is canceled.

<table>
<thead>
<tr>
<th>Status code</th>
<th>Request state</th>
<th>Instance state</th>
</tr>
</thead>
<tbody>
<tr>
<td>fulfilled</td>
<td>active</td>
<td>pending → running</td>
</tr>
<tr>
<td>fulfilled</td>
<td>active</td>
<td>stopped → running</td>
</tr>
</tbody>
</table>

If you stop a Spot Instance, your Spot request goes into the **marked-for-stop** or **instance-stopped-by-user** state until the Spot Instance can be started again or the request is cancelled.

<table>
<thead>
<tr>
<th>Status code</th>
<th>Request state</th>
<th>Instance state</th>
</tr>
</thead>
<tbody>
<tr>
<td>marked-for-stop</td>
<td>active</td>
<td>stopping</td>
</tr>
<tr>
<td>instance-stopped-by-user*</td>
<td>disabled or cancelled**</td>
<td>stopped</td>
</tr>
</tbody>
</table>

* A Spot Instance goes into the **instance-stopped-by-user** state if you stop the instance or run the shutdown command from the instance. After you've stopped the instance, you can start it again. On restart, the Spot Instance request returns to the **pending-evaluation** state and then Amazon EC2 launches a new Spot Instance when the constraints are met.

** The Spot request state is **disabled** if you stop the Spot Instance but do not cancel the request. The request state is **cancelled** if your Spot Instance is stopped and the request expires.

**Fulfilled-terminal**

Your Spot Instances continue to run as long as your maximum price is at or above the Spot price, there is available capacity for your instance type, and you don’t terminate the instance. If a change in the Spot price or available capacity requires Amazon EC2 to terminate your Spot Instances, the Spot request goes into a terminal state. A request also goes into the terminal state if you cancel the Spot request or terminate the Spot Instances.

<table>
<thead>
<tr>
<th>Status code</th>
<th>Request state</th>
<th>Instance state</th>
</tr>
</thead>
<tbody>
<tr>
<td>request-canceled-and-instance-running</td>
<td>cancelled</td>
<td>running</td>
</tr>
</tbody>
</table>
### Table: Status codes, request states, and instance states

<table>
<thead>
<tr>
<th>Status code</th>
<th>Request state</th>
<th>Instance state</th>
</tr>
</thead>
<tbody>
<tr>
<td>marked-for-stop</td>
<td>active</td>
<td>running</td>
</tr>
<tr>
<td>marked-for-termination</td>
<td>active</td>
<td>running</td>
</tr>
<tr>
<td>instance-stopped-by-price</td>
<td>disabled</td>
<td>stopped</td>
</tr>
<tr>
<td>instance-stopped-by-user</td>
<td>disabled</td>
<td>stopped</td>
</tr>
<tr>
<td>instance-stopped-no-capacity</td>
<td>disabled</td>
<td>stopped</td>
</tr>
<tr>
<td>instance-terminated-by-price</td>
<td>closed (one-time), open(persistent)</td>
<td>terminated</td>
</tr>
<tr>
<td>instance-terminated-by-service</td>
<td>cancelled</td>
<td>terminated</td>
</tr>
<tr>
<td>instance-terminated-by-user</td>
<td>closed or cancelled*</td>
<td>terminated</td>
</tr>
<tr>
<td>instance-terminated-no-capacity</td>
<td>closed (one-time), open (persistent)</td>
<td>terminated</td>
</tr>
<tr>
<td>instance-terminated-launch-group-constraint</td>
<td>closed (one-time), open (persistent)</td>
<td>terminated</td>
</tr>
</tbody>
</table>

* 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.

### Get 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/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose Spot Requests and select the Spot request.
3. To check the status, on the Description tab, check the Status field.

**To get request status information using the command line**

You can use one of the following commands. For more information about these command line interfaces, see Access Amazon EC2 (p. 3).
Spot request status codes

Spot request status information is composed of a status code, the update time, and a status message. Together, these help you determine the disposition of your Spot request.

The following are the Spot request status codes:

- **az-group-constraint**
  
  Amazon EC2 cannot launch all the instances you requested in the same Availability Zone.

- **bad-parameters**
  
  One or more parameters for your Spot request are not valid (for example, the AMI you specified does not exist). The status message indicates which parameter is not valid.

- **canceled-before-fulfillment**
  
  The user canceled the Spot request before it was fulfilled.

- **capacity-not-available**
  
  There is not enough capacity available for the instances that you requested.

- **constraint-not-fulfillable**
  
  The Spot request can't be fulfilled because one or more constraints are not valid (for example, the Availability Zone does not exist). The status message indicates which constraint is not valid.

- **fulfilled**
  
  The Spot request is active, and Amazon EC2 is launching your Spot Instances.

- **instance-stopped-by-price**
  
  Your instance was stopped because the Spot price exceeded your maximum price.

- **instance-stopped-by-user**
  
  Your instance was stopped because a user stopped the instance or ran the shutdown command from the instance.

- **instance-stopped-no-capacity**
  
  Your instance was stopped due to EC2 capacity management needs.

- **instance-terminated-by-price**
  
  Your instance was terminated because the Spot price exceeded your maximum price. If your request is persistent, the process restarts, so your request is pending evaluation.

- **instance-terminated-by-schedule**
  
  Your Spot Instance was terminated at the end of its scheduled duration.

- **instance-terminated-by-service**
  
  Your instance was terminated from a stopped state.

- **instance-terminated-by-user or spot-instance-terminated-by-user**
  
  You terminated a Spot Instance that had been fulfilled, so the request state is closed (unless it's a persistent request) and the instance state is terminated.

- **instance-terminated-launch-group-constraint**
  
  One or more of the instances in your launch group was terminated, so the launch group constraint is no longer fulfilled.
instance-terminated-no-capacity

Your instance was terminated due to standard capacity management processes.

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.

EC2 instance rebalance recommendations

An EC2 Instance rebalance recommendation is a new signal that notifies you when a Spot instance is at elevated risk of interruption. The signal can arrive sooner than the two-minute Spot Instance interruption notice (p. 322), giving you the opportunity to proactively manage the Spot Instance. You can decide to rebalance your workload to new or existing Spot Instances that are not at an elevated risk of interruption.
It is not always possible for Amazon EC2 to send the rebalance recommendation signal before the two-minute Spot Instance interruption notice. Therefore, the rebalance recommendation signal can arrive along with the two-minute interruption notice.

**Note**
Rebalance recommendations are only supported for Spot Instances that are launched after November 5, 2020 00:00 UTC.

### Topics
- Rebalance actions you can take (p. 315)
- Monitor rebalance recommendation signals (p. 315)
- Services that use the rebalance recommendation signal (p. 317)

### Rebalance actions you can take
These are some of the possible rebalancing actions that you can take:

**Graceful shutdown**
When you receive the rebalance recommendation signal for a Spot Instance, you can start your instance shutdown procedures, which might include ensuring that processes are completed before stopping them. For example, you can upload system or application logs to Amazon Simple Storage Service (Amazon S3), you can shut down Amazon SQS workers, or you can complete deregistration from the Domain Name System (DNS). You can also save your work in external storage and resume it at a later time.

**Prevent new work from being scheduled**
When you receive the rebalance recommendation signal for a Spot Instance, you can prevent new work from being scheduled on the instance, while continuing to use the instance until the scheduled work is completed.

**Proactively launch new replacement instances**
You can configure Auto Scaling groups, EC2 Fleet, or Spot Fleet to automatically launch replacement Spot Instances when a rebalance recommendation signal is emitted. For more information, see Amazon EC2 Auto Scaling Capacity Rebalancing in the *Amazon EC2 Auto Scaling User Guide*, and Capacity Rebalancing (p. 688) for EC2 Fleet and Capacity Rebalancing (p. 714) for Spot Fleet in this user guide.

### Monitor rebalance recommendation signals
You can monitor the rebalance recommendation signal so that, when it is emitted, you can take the actions that are specified in the preceding section. The rebalance recommendation signal is made available as an event that is sent to Amazon EventBridge (formerly known as Amazon CloudWatch Events) and as instance metadata on the Spot Instance.

**Monitor rebalance recommendation signals:**
- Use Amazon EventBridge (p. 315)
- Use instance metadata (p. 317)

**Use Amazon EventBridge**
When the rebalance recommendation signal is emitted for a Spot Instance, the event for the signal is sent to Amazon EventBridge. If EventBridge detects an event pattern that matches a pattern defined in a rule, EventBridge invokes a target (or targets) specified in the rule.
The following is an example event for the rebalance recommendation signal.

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Instance Rebalance Recommendation",
    "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"
    }
}
```

The following fields form the event pattern that is defined in the rule:

"detail-type": "EC2 Instance Rebalance Recommendation"

Identifies that the event is a rebalance recommendation event

"source": "aws.ec2"

Identifies that the event is from Amazon EC2

**Create an EventBridge rule**

You can write an EventBridge rule and automate what actions to take when the event pattern matches the rule.

The following example creates an EventBridge rule to send an email, text message, or mobile push notification every time Amazon EC2 emits a rebalance recommendation signal. The signal is emitted as an EC2 Instance Rebalance Recommendation event, which triggers the action defined by the rule.

**To create an EventBridge rule for a rebalance recommendation event**

1. Open the Amazon EventBridge console at https://console.aws.amazon.com/events/.
2. Choose **Create rule**.
3. Enter a **Name** for the rule, and, optionally, a description.
   
   A rule can't have the same name as another rule in the same Region and on the same event bus.
4. For **Define pattern**, choose **Event pattern**.
5. Under **Event matching pattern**, choose **Custom pattern**.
6. In the **Event pattern** box, add the following pattern to match the EC2 Instance Rebalance Recommendation event, and then choose **Save**.
   
   ```
   {  
       "source": [ "aws.ec2" ],  
       "detail-type": [ "EC2 Instance Rebalance Recommendation" ]
   }
   ```
7. For **Select event bus**, choose **AWS default event bus**. When an AWS service in your account emits an event, it always goes to your account's default event bus.
8. Confirm that **Enable the rule on the selected event bus** is toggled on.
9. For **Target**, choose **SNS topic** to send an email, text message, or mobile push notification when the event occurs.
10. For **Topic**, choose an existing topic. You first need to create an Amazon SNS topic using the Amazon SNS console. For more information, see Using Amazon SNS for application-to-person (A2P) messaging in the Amazon Simple Notification Service Developer Guide.

11. For **Configure input**, choose the input for the email, text message, or mobile push notification.

12. Choose **Create**.

For more information, see Creating a rule for an AWS service and Event Patterns in the Amazon EventBridge User Guide

**Use instance metadata**

The instance metadata category events/recommendations/rebalance provides the approximate time, in UTC, when the rebalance recommendation signal was emitted for a Spot Instance.

We recommend that you check for rebalance recommendation signals every 5 seconds so that you don't miss an opportunity to act on the rebalance recommendation.

If a Spot Instance receives a rebalance recommendation, the time that the signal was emitted is present in the instance metadata. You can retrieve the time that the signal was emitted as follows.

```
```

The following is example output, which indicates the time, in UTC, that the rebalance recommendation signal was emitted for the Spot Instance.

```
{"noticeTime": "2020-10-27T08:22:00Z"}
```

If the signal has not been emitted for the instance, events/recommendations/rebalance is not present and you receive an HTTP 404 error when you try to retrieve it.

**Services that use the rebalance recommendation signal**

Amazon EC2 Auto Scaling, EC2 Fleet, and Spot Fleet use the rebalance recommendation signal to make it easy for you to maintain workload availability by proactively augmenting your fleet with a new Spot Instance before a running instance receives the two-minute Spot Instance interruption notice. You can have these services monitor and respond proactively to changes affecting the availability of your Spot Instances. For more information, see the following:

- Amazon EC2 Auto Scaling Capacity Rebalancing in the Amazon EC2 Auto Scaling User Guide
- Capacity Rebalancing (p. 688) in the EC2 Fleet topic in this user guide
- Capacity Rebalancing (p. 714) in the Spot Fleet topic in this user guide

**Spot Instance interruptions**

You can launch Spot Instances on spare EC2 capacity for steep discounts in exchange for returning them when Amazon EC2 needs the capacity back. When Amazon EC2 reclaims a Spot Instance, we call this event a **Spot Instance interruption**.

Demand for Spot Instances can vary significantly from moment to moment, and the availability of Spot Instances can also vary significantly depending on how many unused EC2 instances are available. It is always possible that your Spot Instance might be interrupted. Therefore, you must ensure that your application is prepared for a Spot Instance interruption.
An On-Demand Instance specified in an EC2 Fleet or Spot Fleet cannot be interrupted.

Contents

- Reasons for interruption (p. 318)
- Interruption behaviors (p. 318)
- Specify the interruption behavior (p. 321)
- Prepare for interruptions (p. 321)
- Prepare for instance hibernation (p. 321)
- Spot Instance interruption notices (p. 322)
- Find interrupted Spot Instances (p. 323)
- Determine whether Amazon EC2 interrupted a Spot Instance (p. 324)
- Billing for interrupted Spot Instances (p. 324)

Reasons for interruption

The following are the possible reasons that Amazon EC2 might interrupt your Spot Instances:

- Price – The Spot price is greater than your maximum price.
- Capacity – Amazon EC2 can interrupt your Spot Instance when it needs it back. EC2 reclaims your instance mainly to repurpose capacity, but it can also occur for other reasons such as host maintenance or hardware decommission.
- 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.

You can see the historical interruption rates for your instance type in the Spot Instance Advisor.

Interruption behaviors

You can specify that Amazon EC2 must do one of the following when it interrupts a Spot Instance:

- Stop interrupted Spot Instances (p. 318)
- Hibernate interrupted Spot Instances (p. 319)
- Terminate interrupted Spot Instances (this is the default behavior)

To change the interruption behavior, see Specify the interruption behavior (p. 321).

Stop interrupted Spot Instances

Prerequisites

You can specify the interruption behavior so that Amazon EC2 stops Spot Instances when they are interrupted if the following prerequisites are met.

- **Spot Instance request type** – must be persistent. You can't specify a launch group in the Spot Instance request.
- **EC2 Fleet or Spot Fleet request type** – must be maintain
- **Root volume type** – must be an EBS volume, not an instance store volume

After a Spot Instance is stopped by the Spot service, only the Spot service can restart the Spot Instance, and the same launch specification must be used.
For a Spot Instance launched by a persistent Spot Instance request, the Spot service restarts the
stopped instance when capacity is available in the same Availability Zone and for the same instance type
as the stopped instance.

If instances in an EC2 Fleet or Spot Fleet are stopped and the fleet is of type maintain, the Spot
service launches replacement instances to maintain the target capacity. The Spot service finds the
best Spot capacity pools based on the specified allocation strategy (lowestPrice, diversified, or
InstancePoolsToUseCount); it does not prioritize the pool with the earlier stopped instances. Later, if
the allocation strategy leads to a pool containing the earlier stopped instances, the Spot service restarts
the stopped instances to meet the target capacity.

For example, consider a Spot Fleet with the lowestPrice allocation strategy. At initial launch, a
c3.large pool meets the lowestPrice criteria for the launch specification. Later, when the c3.large
instances are interrupted, the Spot service stops the instances and replenishes capacity from another
pool that fits the lowestPrice strategy. This time, the pool happens to be a c4.large pool and the
Spot service launches c4.large instances to meet the target capacity. Similarly, Spot Fleet could move
to a c5.large pool the next time. In each of these transitions, the Spot service does not prioritize pools
with earlier stopped instances, but rather prioritizes purely on the specified allocation strategy. The
lowestPrice strategy can lead back to pools with earlier stopped instances. For example, if instances
are interrupted in the c5.large pool and the lowestPrice strategy leads it back to the c3.large or
c4.large pools, the earlier stopped instances are restarted to fulfill target capacity.

While a Spot Instance is stopped, you can modify some of its instance attributes, but not the instance
type. If you detach or delete an EBS volume, it is not attached when the Spot Instance is started. If you
detach the root volume and the Spot service attempts to start the Spot Instance, instance start fails and
the Spot service terminates the stopped instance.

You can terminate a Spot Instance while it is stopped. If you cancel a Spot Instance request, an EC2 Fleet,
or a Spot Fleet, the Spot service terminates any associated Spot Instances that are stopped.

While a Spot Instance is stopped, you are charged only for the EBS volumes, which are preserved. With
EC2 Fleet and Spot Fleet, if you have many stopped instances, you can exceed the limit on the number of
EBS volumes for your account.

Hibernate interrupted Spot Instances

Hibernate prerequisites

You can specify the interruption behavior so that Amazon EC2 hibernates Spot Instances when they are
interrupted if the following prerequisites are met.

- **Spot Instance request type** – must be persistent. You can't specify a launch group in the Spot
  Instance request.
- **EC2 Fleet or Spot Fleet request type** – must be maintain
- **Supported instance families** – C3, C4, C5, M4, M5, R3, R4
- **Instance RAM size** – must be less than 100 GB
- **Supported operating systems** (You must install the hibernation agent on a supported operating
  system. Alternatively, use a supported AMI, which already includes the agent.):
  - Amazon Linux 2
  - Amazon Linux AMI
  - Ubuntu with an AWS-tuned Ubuntu kernel (linux-aws) greater than 4.4.0-1041
  - Windows Server 2008 R2 and later
- **Supported AMIs** (the following supported AMIs include the hibernation agent):
  - Amazon Linux 2
  - Amazon Linux AMI 2017.09.1 or later
• Ubuntu Xenial 16.04 20171121 or later
• Windows Server 2008 R2 AMI 2017.11.19 or later
• Windows Server 2012 or Windows Server 2012 R2 AMI 2017.11.19 or later
• Windows Server 2016 AMI 2017.11.19 or later
• Windows Server 2019

• Root volume type – must be an EBS volume, not an instance store volume, and it must be large enough to store the instance memory (RAM) during hibernation
• Start the hibernation agent – We recommend that you use user data to start the agent on instance startup. Alternatively, you could start the agent manually.

**Recommendation**

• We strongly recommend that you use an encrypted Amazon EBS volume as the root volume, because instance memory is stored on the root volume during hibernation. This ensures that the contents of memory (RAM) are encrypted when the data is at rest on the volume and when data is moving between the instance and volume. Use one of the following three options to ensure that the root volume is an encrypted Amazon EBS volume:
  • EBS “single-step” encryption: In a single run-instances API call, you can launch encrypted EBS-backed EC2 instances from an unencrypted AMI. For more information, see Use encryption with EBS-backed AMIs (p. 127).
  • EBS encryption by default: You can enable EBS encryption by default to ensure all new EBS volumes created in your AWS account are encrypted. For more information, see Encryption by default (p. 1331).
  • Encrypted AMI: You can enable EBS encryption by using an encrypted AMI to launch your instance. If your AMI does not have an encrypted root snapshot, you can copy it to a new AMI and request encryption. For more information, see Encrypt an unencrypted image during copy (p. 132) and Copy an AMI (p. 117).

When a Spot Instance is hibernated by the Spot service, the EBS volumes are preserved and instance memory (RAM) is preserved on the root volume. The private IP addresses of the instance are also preserved. Instance storage volumes and public IP addresses, other than Elastic IP addresses, are not preserved. While the instance is hibernating, you are charged only for the EBS volumes. With EC2 Fleet and Spot Fleet, if you have many hibernated instances, you can exceed the limit on the number of EBS volumes for your account.

The agent prompts the operating system to hibernate when the instance receives a signal from the Spot service. If the agent is not installed, the underlying operating system doesn't support hibernation, or there isn't enough volume space to save the instance memory, hibernation fails and the Spot service stops the instance instead.

When the Spot service hibernates a Spot Instance, you receive an interruption notice, but you do not have two minutes before the Spot Instance is interrupted. Hibernation begins immediately. While the instance is in the process of hibernating, instance health checks might fail. When the hibernation process completes, the state of the instance is `stopped`.

**Resuming a hibernated Spot Instance**

After a Spot Instance is hibernated by the Spot service, it can only be resumed by the Spot service. The Spot service resumes the instance when capacity becomes available with a Spot price that is less than your specified maximum price.

For more information, see Prepare for instance hibernation (p. 321).

For information about hibernating On-Demand Instances, see Hibernate your On-Demand or Reserved Windows instance (p. 428).
Specify the interruption behavior

If you do not specify an interruption behavior, the default is to terminate Spot Instances when they are interrupted. You can specify the interruption behavior when you create a Spot Instance request. The way in which you specify the interruption behavior is different depending on how you request Spot Instances.

If you request Spot Instances using the launch instance wizard (p. 392), you can specify the interruption behavior as follows: Select the Persistent request check box and then, from Interruption behavior, choose an interruption behavior.

If you request Spot Instances using the Spot console (p. 725), you can specify the interruption behavior as follows: Select the Maintain target capacity check box and then, from Interruption behavior, choose an interruption behavior.

If you configure Spot Instances in a launch template (p. 399), you can specify the interruption behavior as follows: In the launch template, expand Advanced details and select the Request Spot Instances check box. Choose Customize and then, from Interruption behavior, choose an interruption behavior.

If you configure Spot Instances in a launch configuration when using the request-spot-fleet CLI, you can specify the interruption behavior as follows: For InstanceInterruptionBehavior, specify an interruption behavior.

If you configure Spot Instances using the request-spot-instances CLI, you can specify the interruption behavior as follows: For --instance-interruption-behavior, specify an interruption behavior.

Prepare 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.
- Amazon EC2 emits a rebalance recommendation signal to the Spot Instance when the instance is at an elevated risk of interruption. You can rely on the rebalance recommendation to proactively manage Spot Instance interruptions without having to wait for the two-minute Spot Instance interruption notice. For more information, see EC2 instance rebalance recommendations (p. 314).
- Use the two-minute Spot Instance interruption notices to monitor the status of your Spot Instances. For more information, see Spot Instance interruption notices (p. 322).
- While we make every effort to provide these warnings as soon as possible, it is possible that your Spot Instance is interrupted before the warnings can be made available. Test your application to ensure that it handles an unexpected instance interruption gracefully, even if you are monitoring for rebalance recommendation signals and interruption notices. You can do so by running the application using an On-Demand Instance and then terminating the On-Demand Instance yourself.

Prepare 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 Prepare 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 for you to gracefully handle Spot Instance interruptions is to architect your application to be fault-tolerant. To accomplish this, you can take advantage of Spot Instance interruption notices. A Spot Instance interruption notice is a warning that is issued two minutes before Amazon EC2 stops or terminates your Spot Instance. If you specify hibernation as the interruption behavior, you receive an interruption notice, but you do not receive a two-minute warning because the hibernation process begins immediately.

We recommend that you check for these interruption notices every 5 seconds.

The interruption notices are made available as a CloudWatch event and as items in the instance metadata (p. 579) on the Spot Instance. Events are emitted on a best effort basis.

EC2 Spot Instance interruption notice

When Amazon EC2 is going to interrupt your Spot Instance, it emits an event two minutes prior to the actual interruption (except for hibernation, which gets the interruption notice, but not two minutes in advance, because hibernation begins immediately). This event can be detected by Amazon CloudWatch Events. For more information about CloudWatch events, see the Amazon CloudWatch Events User Guide. For a detailed example that walks you through how to create and use event rules, see Taking Advantage of Amazon EC2 Spot Instance Interruption Notices.

The following is an example of the event for Spot Instance interruption. The possible values for instance-action are hibernate, stop, and terminate.

```
{
   "version": "0",
   "id": "12345678-1234-1234-1234-123456789012",
   "detail-type": "EC2 Spot Instance Interruption Warning",
   "source": "aws.ec2",
   "account": "123456789012",
   "time": "yyyy-mm-ddThh:mm:ssZ",
   "region": "us-east-2",
   "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-1234567890abcdef0"],
   "detail": {
       "instance-id": "i-1234567890abcdef0",
       "instance-action": "action"
   }
}
```

instance-action

If your Spot Instance is marked to be stopped or terminated by the Spot service, the instance-action item is present in your instance metadata (p. 579). Otherwise, it is not present. You can retrieve instance-action as follows.
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.

```json
{"action": "stop", "time": "2017-09-18T08:22:00Z"}
```

The following example indicates the time at which this instance will be terminated.

```json
{"action": "terminate", "time": "2017-09-18T08:22:00Z"}
```

If Amazon EC2 is not preparing to stop or terminate the instance, or if you terminated the instance yourself, `instance-action` is not present and you receive an HTTP 404 error when you try to retrieve it.

**termination-time**

This item is maintained for backward compatibility; you should use `instance-action` instead.

If your Spot Instance is marked for termination by the Spot service, the `termination-time` item is present in your instance metadata. Otherwise, it is not present. You can retrieve `termination-time` as follows.

```powershell
```

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.

**Find interrupted Spot Instances**

In the console, the **Instances** pane displays all instances, including Spot Instances. You can identify a Spot Instance from the `spot` value in the **Instance lifecycle** column. The **Instance state** column indicates whether the instance is pending, running, stopping, stopped, shutting-down, or terminated. For a hibernated Spot Instance, the **instance state** is `stopped`.

**To find an interrupted Spot Instance (console)**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Instances**. In the top right corner, choose the settings icon (🔧), and under **Attribute columns**, select **Instance lifecycle**. For Spot Instances, **Instance lifecycle** is `spot`. 

323
Alternatively, in the navigation pane, choose **Spot Requests**. You can see both Spot Instance requests and Spot Fleet requests. To view the IDs of the instances, select a Spot Instance request or a Spot Fleet request and choose the **Instances** tab. Choose an instance ID to display the instance in the **Instances** pane.

3. For each Spot Instance, you can view its state in the **Instance State** column.

**To find interrupted Spot Instances (AWS CLI)**

You can list your interrupted Spot Instances using the `describe-instances` command with the `--filters` parameter. To list only the instance IDs in the output, add the `--query` parameter.

```
aws ec2 describe-instances \
  --filters Name=instance-lifecycle,Values=spot Name=instance-state-name,Values=terminated,stopped \
  --query "Reservations[*].Instances[*].InstanceId"
```

**Determine whether Amazon EC2 interrupted a Spot Instance**

If a Spot Instance is stopped, hibernated, or terminated, you can use CloudTrail to see whether Amazon EC2 interrupted the Spot Instance. In AWS CloudTrail, the event name `BidEvictedEvent` indicates that Amazon EC2 interrupted the Spot Instance.

**To view BidEvictedEvent events in CloudTrail**

1. Open the CloudTrail console at https://console.aws.amazon.com/cloudtrail/.
2. In the navigation pane, choose **Event history**.
3. In the filter drop-down, choose **Event name**, and then in the filter field to the right, enter `BidEvictedEvent`.
4. Choose `BidEvictedEvent` in the resulting list to view its details. Under **Event record**, you can find the instance ID.

For more information about using CloudTrail, see [Log Amazon EC2 and Amazon EBS API calls with AWS CloudTrail](p. 876).

**Billing for interrupted Spot Instances**

When a Spot Instance is interrupted, you’re charged as follows.

<table>
<thead>
<tr>
<th>Who interrupts the Spot Instance</th>
<th>Operating system</th>
<th>Interrupted in the first hour</th>
<th>Interrupted in any hour after the first hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you stop or terminate the Spot Instance</td>
<td>Windows and Linux (excluding RHEL and SUSE)</td>
<td>Charged for the seconds used</td>
<td>Charged for the seconds used</td>
</tr>
<tr>
<td></td>
<td>RHEL and SUSE</td>
<td>Charged for the full hour even if you used a partial hour</td>
<td>Charged for the full hours used, and charged a full hour for the interrupted partial hour</td>
</tr>
<tr>
<td>If the Amazon EC2 Spot service interrupts the Spot Instance</td>
<td>Windows and Linux (excluding RHEL and SUSE)</td>
<td>No charge</td>
<td>Charged for the seconds used</td>
</tr>
</tbody>
</table>
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. 325)
- Amazon S3 bucket requirements (p. 326)
- Subscribe to your Spot Instance data feed (p. 326)
- Describe your Spot Instance data feed (p. 327)
- Delete your Spot Instance data feed (p. 327)

### Data feed file name and format

The Spot Instance data feed file name uses the following format (with the date and hour in UTC):

```
bucket-name.s3.amazonaws.com/optional-prefix/aws-account-id.YYYY-MM-DD-HH.n.unique-id.gz
```

For example, if your bucket name is `my-bucket-name` and your prefix is `my-prefix`, your file names are similar to the following:

```
my-bucket-name.s3.amazonaws.com/my-prefix/111122223333.2019-03-17-20.001.pwBdGTJG.gz
```

For more information about bucket names, see Rules for bucket naming in the Amazon Simple Storage Service Developer Guide.

The Spot Instance data feed files are tab-delimited. Each line in the data file corresponds to one instance hour and contains the fields listed in the following table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>The timestamp used to determine the price charged for this instance usage.</td>
</tr>
<tr>
<td>UsageType</td>
<td>The type of usage and instance type being charged for. For m1.small Spot Instances, this field is set to SpotUsage. For all other instance types, this field is set to SpotUsage:{instance-type}. For example, SpotUsage:c1.medium.</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>The product being charged for. For Linux Spot Instances, this field is set to RunInstances. For Windows Spot Instances, this field is set to RunInstances:0002. Spot usage is grouped according to Availability Zone.</td>
</tr>
<tr>
<td>InstanceID</td>
<td>The ID of the Spot Instance that generated this instance usage.</td>
</tr>
<tr>
<td>MyBidID</td>
<td>The ID for the Spot Instance request that generated this instance usage.</td>
</tr>
<tr>
<td>MyMaxPrice</td>
<td>The maximum price specified for this Spot Instance request.</td>
</tr>
<tr>
<td>MarketPrice</td>
<td>The Spot price at the time specified in the Timestamp field.</td>
</tr>
<tr>
<td>Charge</td>
<td>The price charged for this instance usage.</td>
</tr>
<tr>
<td>Version</td>
<td>The version included in the data feed file name for this record.</td>
</tr>
</tbody>
</table>

### 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.

### Subscribe to your Spot Instance data feed

To subscribe to your data feed, use the `create-spot-datafeed-subscription` command.

```bash
aws ec2 create-spot-datafeed-subscription \
  --bucket my-bucket-name \
  [--prefix my-prefix]
```

The following is example output:

```json
{
  "SpotDatafeedSubscription": {
```
Describe your Spot Instance data feed

To describe your data feed subscription, use the `describe-spot-datafeed-subscription` command.

```bash
aws ec2 describe-spot-datafeed-subscription
```

The following is example output:

```
{
  "SpotDatafeedSubscription": {
    "OwnerId": "123456789012",
    "Prefix": "spotdata",
    "Bucket": "my-s3-bucket",
    "State": "Active"
  }
}
```

Delete your Spot Instance data feed

To delete your data feed, use the `delete-spot-datafeed-subscription` command.

```bash
aws ec2 delete-spot-datafeed-subscription
```

Spot Instance limits

There is a limit on the number of running and requested Spot Instances per AWS account per Region. Spot Instance limits are managed in terms of the number of virtual central processing units (vCPUs) that your running Spot Instances are either using or will use pending the fulfillment of open Spot Instance requests. If you terminate your Spot Instances but do not cancel the Spot Instance requests, the requests count against your Spot Instance vCPU limit until Amazon EC2 detects the Spot Instance terminations and closes the requests.

There are six Spot Instance limits:

- All Standard (A, C, D, H, I, M, R, T, Z) Spot Instance Requests
- All F Spot Instance Requests
- All G Spot Instance Requests
- All Inf Spot Instance Requests
- All P Spot Instance Requests
- All X Spot Instance Requests

Each limit specifies the vCPU limit for one or more instance families. For information about the different instance families, generations, and sizes, see [Amazon EC2 Instance Types](https://aws.amazon.com/ec2/instance-types/).

With vCPU limits, you can use your limit in terms of the number of vCPUs that are required to launch any combination of instance types that meet your changing application needs. For example, say your All Standard Spot Instance Requests limit is 256 vCPUs, you could request 32 m5.2xlarge Spot Instances...
(32 x 8 vCPUs) or 16 c5.4xlarge Spot Instances (16 x 16 vCPUs), or a combination of any Standard Spot Instance types and sizes that total 256 vCPUs.

Topics
- Monitor Spot Instance limits and usage (p. 328)
- Request a Spot Instance limit increase (p. 328)

Monitor Spot Instance limits and usage

You can view and manage your Spot Instance limits using the following:
- The Limits page in the Amazon EC2 console
- The Amazon EC2 Services quotas page in the Service Quotas console
- The get-service-quota AWS CLI

For more information, see Amazon EC2 service quotas (p. 1463) in the Amazon EC2 User Guide for Linux Instances and Viewing a Service Quota in the Service Quotas User Guide.

With Amazon CloudWatch metrics integration, you can monitor EC2 usage against limits. You can also configure alarms to warn about approaching limits. For more information, see Using Amazon CloudWatch Alarms in the Service Quotas User Guide.

Request a Spot Instance limit increase

Even though Amazon EC2 automatically increases your Spot Instance limits based on your usage, you can request a limit increase if necessary. For example, if you intend to launch more Spot Instances than your current limit allows, you can request a limit increase. You can also request a limit increase if you submit a Spot Instance request and you receive the error "Max spot instance count exceeded."

To request a Spot Instance limit increase

2. For Limit type, choose EC2 Spot Instances.
3. For Region, select the required Region.
4. For Primary instance type, select the Spot Instance limit for which you want to request a limit increase.
5. For New limit value, enter the total number of vCPUs that you want to run concurrently. To determine the total number of vCPUs that you need, see Amazon EC2 Instance Types to find the number of vCPUs of each instance type.
6. (Conditional) You must create a separate limit request for each Spot Instance limit. To request an increase for another Spot Instance limit, choose Add another request and repeat steps 4 and 5 in this procedure.
7. For Use case description, enter your use case, and then choose Submit.

For more information about viewing limits and requesting a limit increase, see Amazon EC2 service quotas (p. 1463).

Burstable performance instances

If you launch your Spot Instances using a burstable performance instance type (p. 160), and if you plan to use your burstable performance Spot Instances immediately and for a short duration, with no idle
time for accruing CPU credits, we recommend that you launch them in Standard mode (p. 175) to avoid paying higher costs. If you launch burstable performance Spot Instances in Unlimited mode (p. 168) and burst CPU immediately, you'll spend surplus credits for bursting. If you use the instance for a short duration, the instance doesn't have time to accrue CPU credits to pay down the surplus credits, and you are charged for the surplus credits when you terminate the instance.

Unlimited mode is suitable for burstable performance Spot Instances only if the instance runs long enough to accrue CPU credits for bursting. Otherwise, paying for surplus credits makes burstable performance Spot Instances more expensive than using other instances. For more information, see When to use unlimited mode versus fixed CPU (p. 169).

Launch credits are meant to provide a productive initial launch experience for T2 instances by providing sufficient compute resources to configure the instance. Repeated launches of T2 instances to access new launch credits is not permitted. If you require sustained CPU, you can earn credits (by idling over some period), use Unlimited mode (p. 168) for T2 Spot Instances, or use an instance type with dedicated CPU.

Dedicated Hosts

An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated to your use. Dedicated Hosts allow you to use your existing per-socket, per-core, or per-VM software licenses, including Windows Server, Microsoft SQL Server, SUSE, and Linux Enterprise Server.

For information about the configurations supported on Dedicated Hosts, see Dedicated Hosts Configuration.

Contents

- Differences between Dedicated Hosts and Dedicated Instances (p. 329)
- Bring your own license (p. 330)
- Dedicated Host instance capacity (p. 330)
- Dedicated Hosts restrictions (p. 331)
- Pricing and billing (p. 331)
- Work with Dedicated Hosts (p. 333)
- Work with shared Dedicated Hosts (p. 350)
- Host recovery (p. 355)
- Track configuration changes (p. 359)

Differences between Dedicated Hosts and Dedicated Instances

Dedicated Hosts and Dedicated Instances can both be used to launch Amazon EC2 instances onto physical servers that are dedicated for your use.

There are no performance, security, or physical differences between Dedicated Instances and instances on Dedicated Hosts. However, there are some differences between the two. The following table highlights some of the key differences between Dedicated Hosts and Dedicated Instances:

<table>
<thead>
<tr>
<th></th>
<th>Dedicated Host</th>
<th>Dedicated Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Billing</td>
<td>Per-host billing</td>
<td>Per-instance billing</td>
</tr>
<tr>
<td>Visibility of sockets, cores, and host ID</td>
<td>Provides visibility of the number of sockets and physical cores</td>
<td>No visibility</td>
</tr>
</tbody>
</table>
Amazon Elastic Compute Cloud
User Guide for Windows Instances
Dedicated Hosts

<table>
<thead>
<tr>
<th></th>
<th>Dedicated Host</th>
<th>Dedicated Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host and instance affinity</td>
<td>Allows you to consistently deploy your instances to the same physical server over time</td>
<td>Not supported</td>
</tr>
<tr>
<td>Targeted instance placement</td>
<td>Provides additional visibility and control over how instances are placed on a physical server</td>
<td>Not supported</td>
</tr>
<tr>
<td>Automatic instance recovery</td>
<td>Supported. For more information, see Host recovery (p. 355).</td>
<td>Supported</td>
</tr>
<tr>
<td>Bring Your Own License (BYOL)</td>
<td>Supported</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

Bring your own license

Dedicated Hosts allow you to use your existing per-socket, per-core, or per-VM software licenses. When you bring your own license, you are responsible for managing your own licenses. However, Amazon EC2 has features that help you maintain license compliance, such as instance affinity and targeted placement.

These are the general steps to follow in order to bring your own volume licensed machine image into Amazon EC2.

1. Verify that the license terms controlling the use of your machine images allow usage in a virtualized cloud environment. 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 import your machine image, you can launch instances from it onto active Dedicated Hosts in your account.

4. When you run these instances, depending on the operating system, you might be required to activate these instances against your own KMS server (for example, Windows Server or Windows SQL Server). You can’t 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 Track configuration changes (p. 359).

Dedicated Host instance capacity

Support for multiple instance sizes on the same Dedicated Host is available for the following instance families: A1, C5, M5, R5, C5n, R5n, and M5n. Other instance families support only a single instance size on the same Dedicated Host.

For example, when you allocate an R5 Dedicated Host, it has 2 sockets and 48 physical cores on which you can run different instance sizes, such as r5.2xlarge and r5.4xlarge, up to the core capacity associated with the host. However, for each instance family, there is a limit on the number of instances that can be run for each instance size. For example, an R5 Dedicated Host supports up to 2 r5.8xlarge instances, which uses 32 of the physical cores. Additional R5 instances of another size can then be used.
to fill the host to core capacity. For the supported number of instance sizes for each instance family, see [Dedicated Hosts Configuration](#).

The following table shows examples of different instance size combinations that you can run on a Dedicated Host.

<table>
<thead>
<tr>
<th>Instance family</th>
<th>Example instance size combinations</th>
</tr>
</thead>
</table>
| R5              | • Example 1: 4 x r5.4xlarge + 4 x r5.2xlarge  
|                 | • Example 2: 1 x r5.12xlarge + 1 x r5.4xlarge + 1 x r5.2xlarge + 5 x r5.xlarge + 2 x r5.large  
| C5              | • Example 1: 1 x c5.9xlarge + 2 x c5.4xlarge + 1 x c5.xlarge  
|                 | • Example 2: 4 x c5.4xlarge + 1 x c5.xlarge + 2 x c5.large  
| M5              | • Example 1: 4 x m5.4xlarge + 4 x m5.2xlarge  
|                 | • Example 2: 1 x m5.12xlarge + 1 x m5.4xlarge + 1 x m5.2xlarge + 5 x m5.xlarge + 2 x m5.large  

For more information about the instance families and instance size configurations supported on Dedicated Hosts, see the [Dedicated Hosts Configuration Table](#).

### Dedicated Hosts restrictions

Before you allocate Dedicated Hosts, take note of the following limitations and restrictions:

- To run RHEL, SUSE Linux, and SQL Server on Dedicated Hosts, you must bring your own AMIs. RHEL, SUSE Linux, and SQL Server AMIs that are offered by AWS or that are available on AWS Marketplace can’t be used with Dedicated Hosts. For more information on how to create your own AMI, see [Bring your own license](#).

  This restriction does not apply to hosts allocated for high memory instances (u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, u-18tb1.metal, and u-24tb1.metal). RHEL and SUSE Linux AMIs that are offered by AWS or that are available on AWS Marketplace can be used with these hosts.

- Up to two On-Demand Dedicated Hosts per instance family, per Region can be allocated. It is possible to request a limit increase: [Request to Raise Allocation Limit on Amazon EC2 Dedicated Hosts](#).

- The instances that run on a Dedicated Host can only be launched in a VPC.

- Auto Scaling groups are supported when using a launch template that specifies a host resource group. For more information, see [Creating a Launch Template for an Auto Scaling Group](#) in the *Amazon EC2 Auto Scaling User Guide*.

- Amazon RDS instances are not supported.

- The AWS Free Usage tier is not available for Dedicated Hosts.

- Instance placement control refers to managing instance launches onto Dedicated Hosts. You cannot launch Dedicated Hosts into placement groups.

### Pricing and billing

The price for a Dedicated Host varies by payment option.

#### Payment Options

- [On-Demand Dedicated Hosts](#)
On-Demand Dedicated Hosts

On-Demand billing is automatically activated when you allocate a Dedicated Host to your account.

The On-Demand price for a Dedicated Host varies by instance family and Region. You pay per second (with a minimum of 60 seconds) for active Dedicated Host, regardless of the quantity or the size of instances that you choose to launch on it. For more information about On-Demand pricing, see Amazon EC2 Dedicated Hosts On-Demand Pricing.

You can release an On-Demand Dedicated Host at any time to stop accruing charges for it. For information about releasing a Dedicated Host, see Release Dedicated Hosts (p. 347).

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 in one-year and three-year terms. Only some instance families support the three-year term for No Upfront Reservations.
- **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 can cover one or more hosts that support the same instance family in a single Availability Zone. 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 Availability Zone of the reservation must match that of the Dedicated Hosts you want to associate it with.

When a reservation is associated with a Dedicated Host, the Dedicated Host can’t be released until the reservation’s term is over.

For more information about reservation pricing, see Amazon EC2 Dedicated Hosts Pricing.

Savings Plans

Savings Plans are a flexible pricing model that offers significant savings over On-Demand Instances. With Savings Plans, you make a commitment to a consistent amount of usage, in USD per hour, for a term of one or three years. This provides you with the flexibility to use the Dedicated Hosts that best meet your needs and continue to save money, instead of making a commitment to a specific Dedicated Host. For more information, see the AWS Savings Plans User Guide.

Pricing for Windows Server on Dedicated Hosts

Subject to Microsoft licensing terms, you can bring your existing Windows Server and SQL Server licenses to Dedicated Hosts. There is no additional charge for software usage if you choose to bring your own licenses.
In addition, you can also use Windows Server AMIs provided by Amazon to run the latest versions of Windows Server on Dedicated Hosts. This is common for scenarios where you have existing SQL Server licenses eligible to run on Dedicated Hosts, but need Windows Server to run the SQL Server workload. Windows Server AMIs provided by Amazon are supported on current generation instance types (p. 142) only. For more information, see Amazon EC2 Dedicated Hosts Pricing.

Work with Dedicated Hosts

To use a Dedicated Host, you first allocate hosts for use in your account. You then launch instances onto the hosts by specifying host tenancy for the instance. You must select a specific host for the instance to launch on to, or you can allow it to launch on to any host that has auto-placement enabled and matches its instance type. When an instance is stopped and restarted, the Host affinity setting determines whether it's restarted on the same, or a different, host.

If you no longer need an On-Demand host, you can stop the instances running on the host, direct them to launch on a different host, and then release the host.

Dedicated Hosts are also integrated with AWS License Manager. With License Manager, you can create a host resource group, which is a collection of Dedicated Hosts that are managed as a single entity. When creating a host resource group, you specify the host management preferences, such as auto-allocate and auto-release, for the Dedicated Hosts. This allows you to launch instances onto Dedicated Hosts without manually allocating and managing those hosts. For more information, see Host Resource Groups in the AWS License Manager User Guide.

Contents

- Allocate Dedicated Hosts (p. 333)
- Launch instances onto a Dedicated Host (p. 336)
- Launch instances into a host resource group (p. 337)
- Understand auto-placement and affinity (p. 339)
- Modify Dedicated Host auto-placement (p. 339)
- Modify the supported instance types (p. 340)
- Modify instance tenancy and affinity (p. 342)
- View Dedicated Hosts (p. 343)
- Tag Dedicated Hosts (p. 344)
- Monitor Dedicated Hosts (p. 346)
- Release Dedicated Hosts (p. 347)
- Purchase Dedicated Host Reservations (p. 348)
- View Dedicated Host reservations (p. 349)
- Tag Dedicated Host Reservations (p. 350)

Allocate Dedicated Hosts

To begin using Dedicated Hosts, you must allocate Dedicated Hosts in your account using the Amazon EC2 console or the command line tools. After you allocate the Dedicated Host, the Dedicated Host capacity is made available in your account immediately and you can start launching instances onto the Dedicated Host.

Support for multiple instance sizes of the same instance family on the same Dedicated Host is available for the following instance families: c5, m5, r5, c5n, r5n, and m5n. Other instance families support only one instance size on the same Dedicated Host.

Due to a hardware limitation with N-type Dedicated Hosts, such as C5n, M5n, and R5n, you cannot mix smaller instance sizes (large, xlarge, and 2xlarge) with larger instance sizes (4xlarge, 9xlarge,
You can allocate a Dedicated Host using the following methods.

**New console**

**To allocate a Dedicated Host**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Dedicated Hosts** and then choose **Allocate Dedicated Host**.
3. For **Instance family**, choose the instance family for the Dedicated Host.
4. Specify whether the Dedicated Host supports multiple instance sizes within the selected instance family, or a specific instance type only. Do one of the following.
   - To configure the Dedicated Host to support multiple instance types in the selected instance family, for **Support multiple instance types**, choose **Enable**. Enabling this allows you to launch different instance sizes from the same instance family onto the Dedicated Host. For example, if you choose the m5 instance family and choose this option, you can launch m5.xlarge and m5.4xlarge instances onto the Dedicated Host.
   - To configure the Dedicated Host to support a single instance type within the selected instance family, clear **Support multiple instance types**, and then for **Instance type**, choose the instance type to support. This allows you to launch a single instance type on the Dedicated Host. For example, if you choose this option and specify m5.4xlarge as the supported instance type, you can launch only m5.4xlarge instances onto the Dedicated Host.
5. For **Availability Zone**, choose the Availability Zone in which to allocate the Dedicated Host.
6. To allow the Dedicated Host to accept untargeted instance launches that match its instance type, for **Instance auto-placement**, choose **Enable**. For more information about auto-placement, see **Understand auto-placement and affinity** (p. 339).
7. To enable host recovery for the Dedicated Host, for **Host recovery**, choose **Enable**. For more information, see **Host recovery** (p. 355).
8. For **Quantity**, enter the number of Dedicated Hosts to allocate.
9. (Optional) Choose **Add new tag** and enter a tag key and a tag value.
10. Choose **Allocate**.

**Old console**

**To allocate a Dedicated Host**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Dedicated Hosts**, **Allocate Dedicated Host**.
3. For **Instance family**, choose the instance family for the Dedicated Host.
4. Specify whether the Dedicated Host supports multiple instance sizes within the selected instance family, or a specific instance type only. Do one of the following.
   - To configure the Dedicated Host to support multiple instance types in the selected instance family, select **Support multiple instance types**. Enabling this allows you to launch different instance sizes from the same instance family onto the Dedicated Host. For example, if you choose the m5 instance family and choose this option, you can launch m5.xlarge and m5.4xlarge instances onto the Dedicated Host. The instance family must be powered by the Nitro System.
   - To configure the Dedicated Host to support a single instance type within the selected instance family, clear **Support multiple instance types**, and then for **Instance type**, choose the
instance type to support. This allows you to launch a single instance type on the Dedicated Host. For example, if you choose this option and specify m5.4xlarge as the supported instance type, you can launch only m5.4xlarge instances onto the Dedicated Host.

5. For **Availability Zone**, choose the Availability Zone in which to allocate the Dedicated Host.

6. To allow the Dedicated Host to accept untargeted instance launches that match its instance type, for **Instance auto-placement**, choose **Enable**. For more information about auto-placement, see **Understand auto-placement and affinity** (p. 339).

7. To enable host recovery for the Dedicated Host, for **Host recovery** choose **Enable**. For more information, see **Host recovery** (p. 355).

8. For **Quantity**, enter the number of Dedicated Hosts to allocate.

9. (Optional) Choose **Add Tag** and enter a tag key and a tag value.

10. Choose **Allocate host**.

**AWS CLI**

**To allocate a Dedicated Host**

Use the `allocate-hosts` AWS CLI command. The following command allocates a Dedicated Host that supports multiple instance types from the m5 instance family in us-east-1a Availability Zone. The host also has host recovery enabled and it has auto-placement disabled.

```
aws ec2 allocate-hosts --instance-family "m5" --availability-zone "us-east-1a" --auto-placement "off" --host-recovery "on" --quantity 1
```

The following command allocates a Dedicated Host that supports untargeted m4.large instance launches in the eu-west-1a Availability Zone, enables host recovery, and applies a tag with a key of purpose and a value of production.

```
aws ec2 allocate-hosts --instance-type "m4.large" --availability-zone "eu-west-1a" --auto-placement "on" --host-recovery "on" --quantity 1 --tag-specifications 'ResourceType=dedicated-host,Tags=[{Key=purpose,Value=production}]'
```

**PowerShell**

**To allocate a Dedicated Host**

Use the `New-EC2Host` AWS Tools for Windows PowerShell command. The following command allocates a Dedicated Host that supports multiple instance types from the m5 instance family in us-east-1a Availability Zone. The host also has host recovery enabled and it has auto-placement disabled.

```
PS C:\> New-EC2Host -InstanceFamily m5 -AvailabilityZone us-east-1a -AutoPlacement Off -HostRecovery On -Quantity 1
```

The following commands allocate a Dedicated Host that supports untargeted m4.large instance launches in the eu-west-1a Availability Zone, enable host recovery, and apply a tag with a key of purpose and a value of production.

The `TagSpecification` parameter used to tag a Dedicated Host on creation requires an object that specifies the type of resource to be tagged, the tag key, and the tag value. The following commands create the required object.

```
PS C:\> $tag = @( Key="purpose"; Value="production" )
```
The following command allocates the Dedicated Host and applies the tag specified in the $tagspec object.

```powershell
PS C:\> New-EC2Host -InstanceType m4.large -AvailabilityZone eu-west-1a -AutoPlacement On -HostRecovery On -Quantity 1 -TagSpecification $tagspec
```

**Launch instances onto a Dedicated Host**

After you have allocated a Dedicated Host, you can launch instances onto it. You can't launch instances with host tenancy if you do not have active Dedicated Hosts with enough available capacity for the instance type that you are launching.

**Note**
The instances launched onto Dedicated Hosts can only be launched in a VPC. For more information, see Introduction to VPC.

Before you launch your instances, take note of the limitations. For more information, see Dedicated Hosts restrictions (p. 331).

You can launch an instance onto a Dedicated Host using the following methods.

**Console**

**To launch an instance onto a specific Dedicated Host from the Dedicated Hosts page**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. Choose Dedicated Hosts in the navigation pane.
3. On the Dedicated Hosts page, select a host and choose Actions, Launch Instance(s) onto Host.
4. Select an AMI from the list. SQL Server, SUSE, and RHEL AMIs provided by Amazon EC2 can't be used with Dedicated Hosts.
5. On the Choose an Instance Type page, select the instance type to launch and then choose Next: Configure Instance Details.

If the Dedicated Host supports a single instance type only, the supported instance type is selected by default and can't be changed.

If the Dedicated Host supports multiple instance types, you must select an instance type within the supported instance family based on the available instance capacity of the Dedicated Host. We recommend that you launch the larger instance sizes first, and then fill the remaining instance capacity with the smaller instance sizes as needed.

6. On the Configure Instance Details page, configure the instance settings to suit your needs, and then for Affinity, choose one of the following options:
   - **Off**—The instance launches onto the specified host, but it is not guaranteed to restart on the same Dedicated Host if stopped.
   - **Host**—If stopped, the instance always restarts on this specific host.

For more information about Affinity, see Understand auto-placement and affinity (p. 339).

The Tenancy and Host options are pre-configured based on the host that you selected.
7. Choose **Review and Launch**.
8. On the **Review Instance Launch** page, choose **Launch**.
9. When prompted, select an existing key pair or create a new one, and then choose **Launch Instances**.

**To launch an instance onto a Dedicated Host using the Launch Instance wizard**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Instances, Launch Instance**.
3. Select an AMI from the list. SQL Server, SUSE, and RHEL AMIs provided by Amazon EC2 can't be used with Dedicated Hosts.
4. Select the type of instance to launch and choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, configure the instance settings to suit your needs, and then configure the following settings, which are specific to a Dedicated Host:
   - Tenancy—Choose **Dedicated Host - Launch this instance on a Dedicated Host**.
   - Host—Choose either **Use auto-placement** to launch the instance on any Dedicated Host that has auto-placement enabled, or select a specific Dedicated Host in the list. The list displays only Dedicated Hosts that support the selected instance type.
   - Affinity—Choose one of the following options:
     - **Off**—The instance launches onto the specified host, but it is not guaranteed to restart on it if stopped.
     - **Host**—If stopped, the instance always restarts on the specified host.

   For more information, see Understand auto-placement and affinity (p. 339).

   If you are unable to see these settings, check that you have selected a VPC in the **Network** menu.
6. Choose **Review and Launch**.
7. On the **Review Instance Launch** page, choose **Launch**.
8. When prompted, select an existing key pair or create a new one, and then choose **Launch Instances**.

**AWS CLI**

To launch an instance onto a Dedicated Host

Use the **run-instances** AWS CLI command and specify the instance affinity, tenancy, and host in the **Placement** request parameter.

**PowerShell**

To launch an instance onto a Dedicated Host

Use the **New-EC2Instance** AWS Tools for Windows PowerShell command and specify the instance affinity, tenancy, and host in the **Placement** request parameter.

**Launch instances into a host resource group**

When you launch an instance into a host resource group that has a Dedicated Host with available instance capacity, Amazon EC2 launches the instance onto that host. If the host resource group does not have a host with available instance capacity, Amazon EC2 automatically allocates a new host in the host
resource group, and then launches the instance onto that host. For more information, see Host Resource Groups in the AWS License Manager User Guide.

Requirements and limits

- You must associate a core- or socket-based license configuration with the AMI.
- You can't use SQL Server, SUSE, or RHEL AMIs provided by Amazon EC2 with Dedicated Hosts.
- You can't target a specific host by choosing a host ID, and you can't enable instance affinity when launching an instance into a host resource group.

You can launch an instance into a host resource group using the following methods.

New console

To launch an instance into a host resource group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances, Launch Instances.
3. Select an AMI.
4. Select the type of instance to launch and choose Next: Configure Instance Details.
5. On the Configure Instance Details page, configure the instance settings to suit your needs, and then do the following:
   a. For Tenancy, choose Dedicated Host.
   b. For Host resource group, choose Launch instance into a host resource group.
   c. For Host resource group name, choose the host resource group in which to launch the instance.
8. When prompted, select an existing key pair or create a new one, and then choose Launch Instances.

Old console

To launch an instance into a host resource group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances, Launch Instance.
3. Select an AMI.
4. Select the type of instance to launch and choose Next: Configure Instance Details.
5. On the Configure Instance Details page, configure the instance settings to suit your needs, and then do the following:
   a. For Tenancy, choose Dedicated Host.
   b. For Host resource group, choose Launch instance into a host resource group.
   c. For Host resource group name, choose the host resource group in which to launch the instance.
8. When prompted, select an existing key pair or create a new one, and then choose Launch Instances.
AWS CLI

To launch an instance into a host resource group

Use the `run-instances` AWS CLI command, and in the `Placement` request parameter, omit the Tenancy option and specify the host resource group ARN.

PowerShell

To launch an instance into a host resource group

Use the `New-EC2Instance` AWS Tools for Windows PowerShell command, and in the `Placement` request parameter, omit the Tenancy option and specify the host resource group ARN.

Understand auto-placement and affinity

Placement control for Dedicated Hosts happens on both the instance level and host level.

Auto-placement

Auto-placement is configured at the host level. It allows you to manage whether instances that you launch are launched onto a specific host, or onto any available host that has matching configurations.

When the auto-placement of a Dedicated Host is disabled, it only accepts Host tenancy instance launches that specify its unique host ID. This is the default setting for new Dedicated Hosts.

When the auto-placement of a Dedicated Host is enabled, it accepts any untargeted instance launches that match its instance type configuration.

When launching an instance, you need to configure its tenancy. Launching an instance onto a Dedicated Host without providing a specific `HostId` enables it to launch on any Dedicated Host that has auto-placement enabled and that matches its instance type.

Host affinity

Host affinity is configured at the instance level. It establishes a launch relationship between an instance and a Dedicated Host.

When affinity is set to Host, an instance launched onto a specific host always restarts on the same host if stopped. This applies to both targeted and untargeted launches.

When affinity is set to Off, and you stop and restart the instance, it can be restarted on any available host. However, it tries to launch back onto the last Dedicated Host on which it ran (on a best-effort basis).

Modify Dedicated Host auto-placement

You can modify the auto-placement settings of a Dedicated Host after you have allocated it to your AWS account, using one of the following methods.

New console

To modify the auto-placement of a Dedicated Host

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Dedicated Hosts.
3. Select a host and choose Actions, Modify host.
4. For **Instance auto-placement**, choose **Enable** to enable auto-placement, or clear **Enable** to disable auto-placement. For more information, see **Understand auto-placement and affinity (p. 339)**.

5. Choose **Save**.

**Old console**

**To modify the auto-placement of a Dedicated Host**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Dedicated Hosts** in the navigation pane.
3. On the **Dedicated Hosts** page, select a host and choose **Actions, Modify Auto-Placement**.
4. On the Modify Auto-placement window, for **Allow instance auto-placement**, choose **Yes** to enable auto-placement, or choose **No** to disable auto-placement. For more information, see **Understand auto-placement and affinity (p. 339)**.
5. Choose **Save**.

**AWS CLI**

**To modify the auto-placement of a Dedicated Host**

Use the `modify-hosts` AWS CLI command. The following example enables auto-placement for the specified Dedicated Host.

```bash
aws ec2 modify-hosts --auto-placement on --host-ids h-012a3456b7890cdef
```

**PowerShell**

**To modify the auto-placement of a Dedicated Host**

Use the `Edit-EC2Host` AWS Tools for Windows PowerShell command. The following example enables auto-placement for the specified Dedicated Host.

```powershell
PS C:\> Edit-EC2Host --AutoPlacement 1 --HostId h-012a3456b7890cdef
```

**Modify the supported instance types**

Support for multiple instance types on the same Dedicated Host is available for the following instance families: `c5`, `m5`, `r5`, `c5n`, `r5n`, and `m5n`. Other instance families support only a single instance type on the same Dedicated Host.

You can allocate a Dedicated Host using the following methods.

You can modify a Dedicated Host to change the instance types that it supports. If it currently supports a single instance type, you can modify it to support multiple instance types within that instance family. Similarly, if it currently supports multiple instance types, you can modify it to support a specific instance type only.

To modify a Dedicated Host to support multiple instance types, you must first stop all running instances on the host. The modification takes approximately 10 minutes to complete. The Dedicated Host transitions to the **pending** state while the modification is in progress. You can't start stopped instances or launch new instances on the Dedicated Host while it is in the **pending** state.
To modify a Dedicated Host that supports multiple instance types to support only a single instance type, the host must either have no running instances, or the running instances must be of the instance type that you want the host to support. For example, to modify a host that supports multiple instance types in the m5 instance family to support only m5.large instances, the Dedicated Host must either have no running instances, or it must have only m5.large instances running on it.

You can modify the supported instance types using one of the following methods.

New console

To modify the supported instance types for a Dedicated Host

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the Navigation pane, choose Dedicated Host.
3. Select the Dedicated Host to modify and choose Actions, Modify host.
4. Do one of the following, depending on the current configuration of the Dedicated Host:
   - If the Dedicated Host currently supports a specific instance type, **Support multiple instance types** is not enabled, and **Instance type** lists the supported instance type. To modify the host to support multiple types in the current instance family, for **Support multiple instance types**, choose **Enable**.
     
     You must first stop all instances running on the host before modifying it to support multiple instance types.
   - If the Dedicated Host currently supports multiple instance types in an instance family, **Enabled** is selected for **Support multiple instance types**. To modify the host to support a specific instance type, for **Support multiple instance types**, clear **Enable**, and then for **Instance type**, select the specific instance type to support.
     
     You can't change the instance family supported by the Dedicated Host.

5. Choose **Save**.

Old console

To modify the supported instance types for a Dedicated Host

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the Navigation pane, choose Dedicated Host.
3. Select the Dedicated Host to modify and choose Actions, Modify Supported Instance Types.
4. Do one of the following, depending on the current configuration of the Dedicated Host:
   - If the Dedicated Host currently supports a specific instance type, **No** is selected for **Support multiple instance types**. To modify the host to support multiple types in the current instance family, for **Support multiple instance types**, select **Yes**.
     
     You must first stop all instances running on the host before modifying it to support multiple instance types.
   - If the Dedicated Host currently supports multiple instance types in an instance family, **Yes** is selected for **Support multiple instance types**, and **Instance family** displays the supported instance family. To modify the host to support a specific instance type, for **Support multiple instance types**, select **No**, and then for **Instance type**, select the specific instance type to support.
     
     You can't change the instance family supported by the Dedicated Host.

5. Choose **Save**.
AWS CLI

**To modify the supported instance types for a Dedicated Host**

Use the `modify-hosts` AWS CLI command.

The following command modifies a Dedicated Host to support multiple instance types within the m5 instance family.

```
aws ec2 modify-hosts --instance-family m5 --host-ids h-012a3456b7890cdef
```

The following command modifies a Dedicated Host to support m5.xlarge instances only.

```
aws ec2 modify-hosts --instance-type m5.xlarge --instance-family --host-ids h-012a3456b7890cdef
```

PowerShell

**To modify the supported instance types for a Dedicated Host**

Use the `Edit-EC2Host` AWS Tools for Windows PowerShell command.

The following command modifies a Dedicated Host to support multiple instance types within the m5 instance family.

```
PS C:\> Edit-EC2Host --InstanceFamily m5 --HostId h-012a3456b7890cdef
```

The following command modifies a Dedicated Host to support m5.xlarge instances only.

```
PS C:\> Edit-EC2Host --InstanceType m5.xlarge --HostId h-012a3456b7890cdef
```

**Modify instance tenancy and affinity**

You can change the tenancy of an instance from dedicated to host, or from host to dedicated, after you have launched it. You can also modify the affinity between the instance and the host. To modify either instance tenancy or affinity, the instance must be in the stopped state.

You can modify an instance's tenancy and affinity using the following methods.

**Console**

**To modify instance tenancy or affinity**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Instances**, and select the instance to modify.
3. Choose **Instance state, 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. 360).
• Launch the instance on a Dedicated Host—Launches the instance onto a Dedicated Host with configurable affinity.

• **Affinity**—Choose one of the following:
  - This instance can run on any one of my hosts—The instance launches onto any available Dedicated Host in your account that supports its instance type.
  - This instance can only run on the selected host—The instance is only able to run on the Dedicated Host selected for **Target Host**.

• **Target Host**—Select the Dedicated Host that the instance must run on. If no target host is listed, you might not have available, compatible Dedicated Hosts in your account.

For more information, see Understand auto-placement and affinity (p. 339).

6. Choose **Save**.

**AWS CLI**

**To modify instance tenancy or affinity**

Use the `modify-instance-placement` AWS CLI command. The following example changes the specified instance's affinity from `default` to `host`, and specifies the Dedicated Host that the instance has affinity with.

```
aws ec2 modify-instance-placement --instance-id i-1234567890abcdef0 --affinity host --host-id h-012a3456b7890cdef
```

**PowerShell**

**To modify instance tenancy or affinity**

Use the `Edit-EC2InstancePlacement` AWS Tools for Windows PowerShell command. The following example changes the specified instance's affinity from `default` to `host`, and specifies the Dedicated Host that the instance has affinity with.

```
PS C:\> Edit-EC2InstancePlacement -InstanceId i-1234567890abcdef0 -Affinity host -HostId h-012a3456b7890cdef
```

**View Dedicated Hosts**

You can view details about a Dedicated Host and the individual instances on it using the following methods.

**New console**

**To view the details of a Dedicated Host**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Dedicated Hosts**.
3. On the **Dedicated Hosts** page, select a host.
4. For information about the host, choose **Details**.

**Available vCPUs** indicates the vCPUs that are available on the Dedicated Host for new instance launches. For example, a Dedicated Host that supports multiple instance types within the c5 instance family, and that has no instances running on it, has 72 available vCPUs. This means that
you can launch different combinations of instance types onto the Dedicated Host to consume
the 72 available vCPUs.

For information about instances running on the host, choose Running instances.

Old console

To view the details of a Dedicated Host

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Dedicated Hosts.
3. On the Dedicated Hosts page, select a host.
4. For information about the host, choose Description. Available vCPUs indicates the vCPUs that
   are available on the Dedicated Host for new instance launches. For example, a Dedicated Host
   that supports multiple instance types within the c5 instance family, and that has no instances
   running on it, has 72 available vCPUs. This means that you can launch different combinations of
   instance types onto the Dedicated Host to consume the 72 available vCPUs.

   For information about instances running on the host, choose Instances.

AWS CLI

To view the capacity of a Dedicated Host

Use the describe-hosts AWS CLI command.

The following example uses the describe-hosts (AWS CLI) command to view the available instance
capacity for a Dedicated Host that supports multiple instance types within the c5 instance family.
The Dedicated Host already has two c5.4xlarge instances and four c5.2xlarge instances running
on it.

C:\> aws ec2 describe-hosts --host-id h-012a3456b7890cdef

"AvailableInstanceCapacity": [
   { "AvailableCapacity": 2,
     "InstanceType": "c5.xlarge",
     "TotalCapacity": 18 },
   { "AvailableCapacity": 4,
     "InstanceType": "c5.large",
     "TotalCapacity": 36 }
],
"AvailableVCpus": 8

PowerShell

To view the instance capacity of a Dedicated Host


PS C:\> Get-EC2Host -HostId h-012a3456b7890cdef

Tag Dedicated Hosts

You can assign custom tags to your existing Dedicated Hosts to categorize them in different ways, for
example, by purpose, owner, or environment. This helps you to quickly find a specific Dedicated Host
based on the custom tags that you assigned. Dedicated Host tags can also be used for cost allocation tracking.

You can also apply tags to Dedicated Hosts at the time of creation. For more information, see Allocate Dedicated Hosts (p. 333).

You can tag a Dedicated Host using the following methods.

New console

**To tag a Dedicated Host**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Dedicated Hosts.
3. Select the Dedicated Host to tag, and then choose Actions, Manage tags.
4. In the Manage tags screen, choose Add tag, and then specify the key and value for the tag.
5. (Optional) Choose Add tag to add additional tags to the Dedicated Host.
6. Choose Save changes.

Old console

**To tag a Dedicated Host**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Dedicated Hosts.
3. Select the Dedicated Host to tag, and then choose Tags.
4. Choose Add/Edit Tags.
5. In the Add/Edit Tags dialog box, choose Create Tag, and then specify the key and value for the tag.
6. (Optional) Choose Create Tag to add additional tags to the Dedicated Host.
7. Choose Save.

AWS CLI

**To tag a Dedicated Host**

Use the `create-tags` AWS CLI command.

The following command tags the specified Dedicated Host with `Owner=TeamA`.

```
aws ec2 create-tags --resources h-abc12345678909876 --tags Key=Owner,Value=TeamA
```

PowerShell

**To tag a Dedicated Host**

Use the `New-EC2Tag` AWS Tools for Windows PowerShell command.

The `New-EC2Tag` command needs a `Tag` object, which specifies the key and value pair to be used for the Dedicated Host tag. The following commands create a `Tag` object named `$tag`, with a key and value pair of `Owner` and `TeamA` respectively.

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag
PS C:\> $tag.Key = "Owner"
```
Monitor Dedicated Hosts

Amazon EC2 constantly monitors the state of your Dedicated Hosts. Updates are communicated on the Amazon EC2 console. You can view information about a Dedicated Host using the following methods.

**Console**

**To view the state of a Dedicated Host**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Dedicated Hosts.
3. Locate the Dedicated Host in the list and review the value in the State column.

**AWS CLI**

**To view the state of a Dedicated Host**

Use the describe-hosts AWS CLI command and then review the state property in the hostSet response element.

```bash
aws ec2 describe-hosts --host-id h-012a3456b7890cdef
```

**PowerShell**

**To view the state of a Dedicated Host**

Use the Get-EC2Host AWS Tools for Windows PowerShell command and then review the state property in the hostSet response element.

```powershell
PS C:\> Get-EC2Host -HostId h-012a3456b7890cdef
```

The following table explains the possible Dedicated Host states.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>available</td>
<td>AWS hasn't detected an issue with the Dedicated Host. No maintenance or repairs are scheduled. Instances can be launched onto this Dedicated Host.</td>
</tr>
<tr>
<td>released</td>
<td>The Dedicated Host has been released. The host ID is no longer in use. Released hosts can't be reused.</td>
</tr>
<tr>
<td>under-assessment</td>
<td>AWS is exploring a possible issue with the Dedicated Host. If action must be taken, you are notified via the AWS Management Console or email. Instances can't be launched onto a Dedicated Host in this state.</td>
</tr>
<tr>
<td>pending</td>
<td>The Dedicated Host cannot be used for new instance launches. It is either being modified to support multiple instance types (p. 340), or a host recovery (p. 355) is in progress.</td>
</tr>
</tbody>
</table>
### State Description

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>permanent-failure</td>
<td>An unrecoverable failure has been detected. You receive an eviction notice through your instances and by email. Your instances might continue to run. If you stop or terminate all instances on a Dedicated Host with this state, AWS retires the host. AWS does not restart instances in this state. Instances can’t be launched onto Dedicated Hosts in this state.</td>
</tr>
<tr>
<td>released-permanent-failure</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Release Dedicated Hosts**

Any running instances on the Dedicated Host must be stopped before you can release the host. These instances can be migrated to other Dedicated Hosts in your account so that you can continue to use them. These steps apply only to On-Demand Dedicated Hosts.

You can release a Dedicated Host using the following methods.

**New console**

**To release a Dedicated Host**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose Dedicated Hosts.
3. On the Dedicated Hosts page, select the Dedicated Host to release.
5. To confirm, choose Release.

**Old console**

**To release a Dedicated Host**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](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.
5. Choose Release to confirm.

**AWS CLI**

**To release a Dedicated Host**

Use the release-hosts AWS CLI command.

```
aws ec2 release-hosts --host-ids h-012a3456b7890cdef
```

**PowerShell**

**To release a Dedicated Host**

After you release a Dedicated Host, you can't reuse the same host or host ID again, and you are no longer charged On-Demand billing rates for it. The state of the Dedicated Host is changed to released, and you are not able to launch any instances onto that host.

**Note**
If you have recently released Dedicated Hosts, it can take some time for them to stop counting towards your limit. During this time, you might experience LimitExceeded errors when trying to allocate new Dedicated Hosts. If this is the case, try allocating new hosts again after a few minutes.

The instances that were stopped are still available for use and are listed on the **Instances** page. They retain their host tenancy setting.

**Purchase Dedicated Host Reservations**

You can purchase reservations using the following methods:

**Console**

To purchase reservations

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose Dedicated Hosts, Dedicated Host Reservations, Purchase Dedicated Host Reservation.
3. On the Purchase Dedicated Host Reservation screen, you can search for available offerings using the default settings, or you can specify custom values for the following:
   - **Host instance family**—The options listed correspond with the Dedicated Hosts in your account that are not already assigned to a reservation.
   - **Availability Zone**—The Availability Zone of the Dedicated Hosts in your account that aren't already assigned to a reservation.
   - **Payment option**—The payment option for the offering.
   - **Term**—The term of the reservation, which can be one or three years.
4. Choose Find offering and select an offering that matches your requirements.
5. Choose the Dedicated Hosts to associate with the reservation, and then choose **Review**.
6. Review your order and choose **Order**.

**AWS CLI**

To purchase reservations

1. Use the describe-host-reservation-offerings AWS CLI command to list the available offerings that match your needs. The following example lists the offerings that support instances in the m4 instance family and have a one-year term.

   **Note**
   The term is specified in seconds. A one-year term includes 31,536,000 seconds, and a three-year term includes 94,608,000 seconds.

   ```bash
   aws ec2 describe-host-reservation-offerings --filter Name=instance-family,Values=m4 --max-duration 31536000
   ```

   The command returns a list of offerings that match your criteria. Note the offeringId of the offering to purchase.
2. Use the `purchase-host-reservation` AWS CLI command to purchase the offering and provide the `offeringId` noted in the previous step. The following example purchases the specified reservation and associates it with a specific Dedicated Host that is already allocated in the AWS account, and it applies a tag with a key of `purpose` and a value of `production`.

```
aws ec2 purchase-host-reservation --offering-id hro-03f707bf363b6b324 --host-id-set h-013abcd2a00cbd123 --tag-specifications 'ResourceType=host-reservation,Tags={Key=purpose,Value=production}'
```

### PowerShell

#### To purchase reservations

1. Use the `Get-EC2HostReservationOffering` AWS Tools for Windows PowerShell command to list the available offerings that match your needs. The following examples list the offerings that support instances in the `m4` instance family and have a one-year term.

   **Note**
   The term is specified in seconds. A one-year term includes 31,536,000 seconds, and a three-year term includes 94,608,000 seconds.

   ```
   PS C:\> $filter = @{Name="instance-family"; Value="m4"}
   PS C:\> Get-EC2HostReservationOffering -filter $filter -MaxDuration 31536000
   ```

   The command returns a list of offerings that match your criteria. Note the `offeringId` of the offering to purchase.

2. Use the `New-EC2HostReservation` AWS Tools for Windows PowerShell command to purchase the offering and provide the `offeringId` noted in the previous step. The following example purchases the specified reservation and associates it with a specific Dedicated Host that is already allocated in the AWS account.

   ```
   PS C:\> New-EC2HostReservation -OfferingId hro-03f707bf363b6b324 -HostIdSet h-013abcd2a00cbd123
   ```

### View Dedicated Host reservations

You can view information about the Dedicated Hosts that are associated with your reservation, including:

- The term of the reservation
- The payment option
- The start and end dates

You can view details of your Dedicated Host reservations using the following methods.

#### Console

**To view the details of a Dedicated Host reservation**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Dedicated Hosts** in the navigation pane.
3. On the **Dedicated Hosts** page, choose **Dedicated Host Reservations**, and then select the reservation from the list provided.
4. Choose **Details** for information about the reservation.
5. Choose **Hosts** for information about the Dedicated Hosts with which the reservation is associated.

**AWS CLI**

*To view the details of a Dedicated Host reservation*

Use the `describe-host-reservations` AWS CLI command.

```
aws ec2 describe-host-reservations
```

**PowerShell**

*To view the details of a Dedicated Host reservation*

Use the `Get-EC2HostReservation` AWS Tools for Windows PowerShell command.

```
PS C:\> Get-EC2HostReservation
```

**Tag Dedicated Host Reservations**

You can assign custom tags to your Dedicated Host Reservations to categorize them in different ways, for example, by purpose, owner, or environment. This helps you to quickly find a specific Dedicated Host Reservation based on the custom tags that you assigned.

You can tag a Dedicated Host Reservation using the command line tools only.

**AWS CLI**

*To tag a Dedicated Host Reservation*

Use the `create-tags` AWS CLI command.

```
aws ec2 create-tags --resources hr-1234563a4ffc669ae --tags Key=Owner,Value=TeamA
```

**PowerShell**

*To tag a Dedicated Host Reservation*

Use the `New-EC2Tag` AWS Tools for Windows PowerShell command.

The `New-EC2Tag` command needs a `Tag` parameter, which specifies the key and value pair to be used for the Dedicated Host Reservation tag. The following commands create the `Tag` parameter.

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag
PS C:\> $tag.Key = "Owner"
PS C:\> $tag.Value = "TeamA"

PS C:\> New-EC2Tag -Resource hr-1234563a4ffc669ae -Tag $tag
```

**Work with shared Dedicated Hosts**

Dedicated Host sharing enables Dedicated Host owners to share their Dedicated Hosts with other AWS accounts or within an AWS organization. This enables you to create and manage Dedicated Hosts centrally, and share the Dedicated Host across multiple AWS accounts or within your AWS organization.
In this model, the AWS account that owns the Dedicated Host (owner) shares it with other AWS accounts (consumers). Consumers can launch instances onto Dedicated Hosts that are shared with them in the same way that they would launch instances onto Dedicated Hosts that they allocate in their own account. The owner is responsible for managing the Dedicated Host and the instances that they launch onto it. Owners can't modify instances that consumers launch onto shared Dedicated Hosts. Consumers are responsible for managing the instances that they launch onto Dedicated Hosts shared with them. Consumers can't view or modify instances owned by other consumers or by the Dedicated Host owner, and they can't modify Dedicated Hosts that are shared with them.

A Dedicated Host owner can share a Dedicated Host with:

- Specific AWS accounts inside or outside of its AWS organization
- An organizational unit inside its AWS organization
- Its entire AWS organization

### Contents
- Prerequisites for sharing Dedicated Hosts (p. 351)
- Limitations for sharing Dedicated Hosts (p. 351)
- Related services (p. 351)
- Share across Availability Zones (p. 352)
- Share a Dedicated Host (p. 352)
- Unshare a shared Dedicated Host (p. 353)
- Identify a shared Dedicated Host (p. 353)
- View instances running on a shared Dedicated Host (p. 354)
- Shared Dedicated Host permissions (p. 354)
- Billing and metering (p. 355)
- Dedicated Host limits (p. 355)
- Host recovery and Dedicated Host sharing (p. 355)

### Prerequisites for sharing Dedicated Hosts

- To share a Dedicated Host, you must own it in your AWS account. You can't share a Dedicated Host that has been shared with you.
- To share a Dedicated Host with your AWS organization or an organizational unit in your AWS organization, you must enable sharing with AWS Organizations. For more information, see Enable Sharing with AWS Organizations in the AWS RAM User Guide.

### Limitations for sharing Dedicated Hosts

You can't share Dedicated Hosts that have been allocated for the following instance types: u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, u-18tb1.metal, and u-24tb1.metal.

### Related services

**AWS Resource Access Manager**

Dedicated Host sharing integrates with AWS Resource Access Manager (AWS RAM). AWS RAM is a service that enables you to share your AWS resources with any AWS account or through AWS Organizations. With AWS RAM, you share resources that you own by creating a resource share. A resource share specifies the resources to share, and the consumers with whom to share them. Consumers can be individual AWS accounts, or organizational units or an entire organization from AWS Organizations.
Share across Availability Zones

To ensure that resources are distributed across the Availability Zones for a Region, we independently map Availability Zones to names for each account. This could lead to Availability Zone naming differences across accounts. For example, the Availability Zone `us-east-1a` for your AWS account might not have the same location as `us-east-1a` for another AWS account.

To identify the location of your Dedicated Hosts relative to your accounts, you must use the Availability Zone ID (AZ ID). The Availability Zone ID is a unique and consistent identifier for an Availability Zone across all AWS accounts. For example, `use1-az1` is an Availability Zone ID for the `us-east-1` Region and it is the same location in every AWS account.

To view the Availability Zone IDs for the Availability Zones in your account

2. The Availability Zone IDs for the current Region are displayed in the *Your AZ ID* panel on the right-hand side of the screen.

Share a Dedicated Host

When an owner shares a Dedicated Host, it enables consumers to launch instances on the host. Consumers can launch as many instances onto the shared host as its available capacity allows.

**Important**

Note that you are responsible for ensuring that you have appropriate license rights to share any BYOL licenses on your Dedicated Hosts.

If you share a Dedicated Host with auto-placement enabled, keep the following in mind as it could lead to unintended Dedicated Host usage:

- If consumers launch instances with Dedicated Host tenancy and they do not have capacity on a Dedicated Host that they own in their account, the instance is automatically launched onto the shared Dedicated Host.

To share a Dedicated Host, you must add it to a resource share. A resource share is an AWS RAM resource that lets you share your resources across AWS accounts. A resource share specifies the resources to share, and the consumers with whom they are shared. You can add the Dedicated Host to an existing resource, or you can add it to a new resource share.

If you are part of an organization in AWS Organizations and sharing within your organization is enabled, consumers in your organization are automatically granted access to the shared Dedicated Host. Otherwise, consumers receive an invitation to join the resource share and are granted access to the shared Dedicated Host after accepting the invitation.

**Note**

After you share a Dedicated Host, it could take a few minutes for consumers to have access to it.

You can share a Dedicated Host that you own by using one of the following methods.

Amazon EC2 console

**To share a Dedicated Host that you own using the Amazon EC2 console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Dedicated Hosts.
3. Choose the Dedicated Host to share and choose Actions, Share host.
4. Select the resource share to which to add the Dedicated Host and choose **Share host**.

It could take a few minutes for consumers to get access to the shared host.

**AWS RAM console**

**To share a Dedicated Host that you own using the AWS RAM console**


**AWS CLI**

**To share a Dedicated Host that you own using the AWS CLI**

Use the `create-resource-share` command.

---

**Unshare a shared Dedicated Host**

The Dedicated Host owner can unshare a shared Dedicated Host at any time. When you unshare a shared Dedicated Host, the following rules apply:

- Consumers with whom the Dedicated Host was shared can no longer launch new instances onto it.
- Instances owned by consumers that were running on the Dedicated Host at the time of unsharing continue to run but are scheduled for **retirement**. Consumers receive retirement notifications for the instances and they have two weeks to take action on the notifications. However, if the Dedicated Host is reshared with the consumer within the retirement notice period, the instance retirements are cancelled.

To unshare a shared Dedicated Host that you own, you must remove it from the resource share. You can do this by using one of the following methods.

**Amazon EC2 console**

**To unshare a shared Dedicated Host that you own using the Amazon EC2 console**

1. Open the Amazon EC2 console at `https://console.aws.amazon.com/ec2/`.
2. In the navigation pane, choose **Dedicated Hosts**.
3. Choose the Dedicated Host to unshare and choose the **Sharing** tab.
4. The **Sharing** tab lists the resource shares to which the Dedicated Host has been added. Select the resource share from which to remove the Dedicated Host and choose **Remove host from resource share**.

**AWS RAM console**

**To unshare a shared Dedicated Host that you own using the AWS RAM console**


**Command line**

**To unshare a shared Dedicated Host that you own using the AWS CLI**

Use the `disassociate-resource-share` command.

---

**Identify a shared Dedicated Host**

Owners and consumers can identify shared Dedicated Hosts using one of the following methods.
Amazon Elastic Compute Cloud
User Guide for Windows Instances
Dedicated Hosts

Amazon EC2 console

To identify a shared Dedicated Host using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Dedicated Hosts. The screen lists Dedicated Hosts that you own and Dedicated Hosts that are shared with you. The Owner column shows the AWS account ID of the Dedicated Host owner.

Command line

To identify a shared Dedicated Host using the AWS CLI

Use the describe-hosts command. The command returns the Dedicated Hosts that you own and Dedicated Hosts that are shared with you.

View instances running on a shared Dedicated Host

Owners and consumers can view the instances running on a shared Dedicated Host at any time using one of the following methods.

Amazon EC2 console

To view the instances running on a shared Dedicated Host using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Dedicated Hosts.
3. Select the Dedicated Host for which to view the instances and choose Instances. The tab lists the instances that are running on the host. Owners see all of the instances running on the host, including instances launched by consumers. Consumers only see running instances that they launched onto the host. The Owner column shows the AWS account ID of the account that launched the instance.

Command line

To view the instances running on a shared Dedicated Host using the AWS CLI

Use the describe-hosts command. The command returns the instances running on each Dedicated Host. Owners see all of the instances running on the host. Consumers only see running instances that they launched on the shared hosts. InstanceOwnerId shows the AWS account ID of the instance owner.

Shared Dedicated Host permissions

Permissions for owners

 Owners are responsible for managing their shared Dedicated Hosts and the instances that they launch onto them. Owners can view all instances running on the shared Dedicated Host, including those launched by consumers. However, owners can't take any action on running instances that were launched by consumers.

Permissions for consumers

Consumers are responsible for managing the instances that they launch onto a shared Dedicated Host. Consumers can't modify the shared Dedicated Host in any way, and they can't view or modify instances that were launched by other consumers or the Dedicated Host owner.
Billing and metering

There are no additional charges for sharing Dedicated Hosts.

Owners are billed for Dedicated Hosts that they share. Consumers are not billed for instances that they launch onto shared Dedicated Hosts.

Dedicated Host Reservations continue to provide billing discounts for shared Dedicated Hosts. Only Dedicated Host owners can purchase Dedicated Host Reservations for shared Dedicated Hosts that they own.

Dedicated Host limits

Shared Dedicated Hosts count towards the owner’s Dedicated Hosts limits only. Consumer’s Dedicated Hosts limits are not affected by Dedicated Hosts that have been shared with them. Similarly, instances that consumers launch onto shared Dedicated Hosts do not count towards their instance limits.

Host recovery and Dedicated Host sharing

Host recovery recovers instances launched by the Dedicated Host owner and the consumers with whom it has been shared. The replacement Dedicated Host is allocated to the owner’s account. It is added to the same resource shares as the original Dedicated Host, and it is shared with the same consumers.

For more information, see Host recovery (p. 355).

Host recovery

Host recovery automatically restarts your instances on to a new replacement host if failures are detected on your Dedicated Host. Host recovery reduces the need for manual intervention and lowers the operational burden if there is an unexpected Dedicated Host failure.

Additionally, built-in integration with AWS License Manager automates the tracking and management of your licenses if a host recovery occurs.

Note
AWS License Manager integration is supported only in Regions in which AWS License Manager is available.

Contents
- Host recovery basics (p. 355)
- Supported instance types (p. 356)
- Configure host recovery (p. 357)
- Host recovery states (p. 358)
- Manually recover unsupported instances (p. 358)
- Related services (p. 359)
- Pricing (p. 359)

Host recovery basics

Host recovery uses host-level health checks to assess Dedicated Host availability and to detect underlying system failures. Examples of problems that can cause host-level health checks to fail include:

- Loss of network connectivity
- Loss of system power
• Hardware or software issues on the physical host

When a system failure is detected on your Dedicated Host, host recovery is initiated and Amazon EC2 automatically allocates a replacement Dedicated Host. The replacement Dedicated Host receives a new host ID, but retains the same attributes as the original Dedicated Host, including:

• Availability Zone
• Instance type
• Tags
• Auto placement settings

After the replacement Dedicated Host is allocated, the instances are recovered on to the replacement Dedicated Host. The recovered instances retain the same attributes as the original instances, including:

• Instance ID
• Private IP addresses
• Elastic IP addresses
• EBS volume attachments
• All instance metadata

If instances have a host affinity relationship with the impaired Dedicated Host, the recovered instances establish host affinity with the replacement Dedicated Host.

When all of the instances have been recovered on to the replacement Dedicated Host, the impaired Dedicated Host is released, and the replacement Dedicated Host becomes available for use.

When host recovery is initiated, the AWS account owner is notified by email and by an AWS Personal Health Dashboard event. A second notification is sent after the host recovery has been successfully completed.

Stopped instances are not recovered on to the replacement Dedicated Host. If you attempt to start a stopped instance that targets the impaired Dedicated Host, the instance start fails. We recommend that you modify the stopped instance to either target a different Dedicated Host, or to launch on any available Dedicated Host with matching configurations and auto-placement enabled.

Instances with instance storage are not recovered on to the replacement Dedicated Host. As a remedial measure, the impaired Dedicated Host is marked for retirement and you receive a retirement notification after the host recovery is complete. Follow the remedial steps described in the retirement notification within the specified time period to manually recover the remaining instances on the impaired Dedicated Host.

If you are using AWS License Manager to track your licenses, AWS License Manager allocates new licenses for the replacement Dedicated Host based on the license configuration limits. If the license configuration has hard limits that will be breached as a result of the host recovery, the recovery process is not allowed and you are notified of the host recovery failure through an Amazon SNS notification. If the license configuration has soft limits that will be breached as a result of the host recovery, the recovery is allowed to continue and you are notified of the limit breach through an Amazon SNS notification. For more information, see Using License Configurations in the AWS License Manager User Guide.

Supported instance types

Host recovery is supported for the following instance families: A1, C3, C4, C5, C5n, M3, M4, M5, M5n, P3, R3, R4, R5, R5n, X1, X1e, u-6tb1, u-9tb1, u-12tb1, u-18tb1, and u-24tb1.
To recover instances that are not supported, see Manually recover unsupported instances (p. 358).

Configure host recovery

You can configure host recovery at the time of Dedicated Host allocation, or after allocation using the Amazon EC2 console or AWS Command Line Interface (CLI).

Contents

- Enable host recovery (p. 357)
- Disable host recovery (p. 357)
- View the host recovery configuration (p. 358)

Enable host recovery

You can enable host recovery at the time of Dedicated Host allocation or after allocation.

For more information about enabling host recovery at the time of Dedicated Host allocation, see Allocate Dedicated Hosts (p. 333).

To enable host recovery after allocation using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Dedicated Hosts.
3. Select the Dedicated Host for which to enable host recovery, and then choose Actions, Modify Host Recovery.
4. For Host recovery, choose Enable, and then choose Save.

To enable host recovery after allocation using the AWS CLI

Use the modify-hosts command and specify the host-recovery parameter.

```
$ aws ec2 modify-hosts --host-recovery on --host-ids h-012a3456b7890cdef
```

Disable host recovery

You can disable host recovery at any time after the Dedicated Host has been allocated.

To disable host recovery after allocation using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Dedicated Hosts.
3. Select the Dedicated Host for which to disable host recovery, and then choose Actions, Modify Host Recovery.
4. For Host recovery, choose Disable, and then choose Save.

To disable host recovery after allocation using the AWS CLI

Use the modify-hosts command and specify the host-recovery parameter.

```
$ aws ec2 modify-hosts --host-recovery off --host-ids h-012a3456b7890cdef
```
View the host recovery configuration

You can view the host recovery configuration for a Dedicated Host at any time.

To view the host recovery configuration for a Dedicated Host using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Dedicated Hosts.
3. Select the Dedicated Host, and in the Description tab, review the Host Recovery field.

To view the host recovery configuration for a Dedicated Host using the AWS CLI

Use the describe-hosts command.

```
$ aws ec2 describe-hosts --host-ids h-012a3456b7890cdef
```

The HostRecovery response element indicates whether host recovery is enabled or disabled.

Host recovery states

When a Dedicated Host failure is detected, the impaired Dedicated Host enters the under-assessment state, and all of the instances enter the impaired state. You can’t launch instances on to the impaired Dedicated Host while it is in the under-assessment state.

After the replacement Dedicated Host is allocated, it enters the pending state. It remains in this state until the host recovery process is complete. You can’t launch instances on to the replacement Dedicated Host while it is in the pending state. Recovered instances on the replacement Dedicated Host remain in the impaired state during the recovery process.

After the host recovery is complete, the replacement Dedicated Host enters the available state, and the recovered instances return to the running state. You can launch instances on to the replacement Dedicated Host after it enters the available state. The original impaired Dedicated Host is permanently released and it enters the released-permanent-failure state.

If the impaired Dedicated Host has instances that do not support host recovery, such as instances with instance store-backed volumes, the Dedicated Host is not released. Instead, it is marked for retirement and enters the permanent-failure state.

Manually recover unsupported instances

Host recovery does not support recovering instances that use instance store volumes. Follow the instructions below to manually recover any of your instances that could not be automatically recovered.

**Warning**

Data on instance store volumes is lost when an instance is stopped, hibernated, or terminated. This includes instance store volumes that are attached to an instance that has an EBS volume as the root device. To protect data from instance store volumes, back it up to persistent storage before the instance is stopped or terminated.

Manually recover EBS-backed instances

For EBS-backed instances that could not be automatically recovered, we recommend that you manually stop and start the instances to recover them onto a new Dedicated Host. For more information about stopping your instance, and about the changes that occur in your instance configuration when it's stopped, see Stop and start your instance (p. 425).
Manually recover instance store-backed instances

For instance store-backed instances that could not be automatically recovered, we recommend that you do the following:

1. Launch a replacement instance on a new Dedicated Host from your most recent AMI.
2. Migrate all of the necessary data to the replacement instance.
3. Terminate the original instance on the impaired Dedicated Host.

Related services

Dedicated Host integrates with the following services:

- AWS License Manager—Tracks licenses across your Amazon EC2 Dedicated Hosts (supported only in Regions in which AWS License Manager is available). For more information, see the AWS License Manager User Guide.

Pricing

There are no additional charges for using host recovery, but the usual Dedicated Host charges apply. For more information, see Amazon EC2 Dedicated Hosts Pricing.

As soon as host recovery is initiated, you are no longer billed for the impaired Dedicated Host. Billing for the replacement Dedicated Host begins only after it enters the available state.

If the impaired Dedicated Host was billed using the On-Demand rate, the replacement Dedicated Host is also billed using the On-Demand rate. If the impaired Dedicated Host had an active Dedicated Host Reservation, it is transferred to the replacement Dedicated Host.

Track configuration changes

You can use AWS Config to record configuration changes for Dedicated Hosts, and for instances that are launched, stopped, or terminated on them. You can then use the information captured by AWS Config as a data source for license reporting.

AWS Config records configuration information for Dedicated Hosts and instances individually, and pairs this information through relationships. There are three reporting conditions:

- AWS Config recording status—When On, AWS Config is recording one or more AWS resource types, which can include Dedicated Hosts and Dedicated Instances. To capture the information required for license reporting, verify that hosts and instances are being recorded with the following fields.

- Host recording status—When Enabled, the configuration information for Dedicated Hosts is recorded.

- Instance recording status—When Enabled, the configuration information for Dedicated Instances is recorded.

If any of these three conditions are disabled, the icon in the Edit Config Recording button is red. To derive the full benefit of this tool, ensure that all three recording methods are enabled. When all three are enabled, the icon is green. To edit the settings, choose Edit Config Recording. You are directed to the Set up AWS Config page in the AWS Config console, where you can set up AWS Config and start recording for your hosts, instances, and other supported resource types. For more information, see Setting up AWS Config using the Console in the AWS Config Developer Guide.

Note

AWS Config records your resources after it discovers them, which might take several minutes.
After AWS Config starts recording configuration changes to your hosts and instances, you can get the configuration history of any host that you have allocated or released and any instance that you have launched, stopped, or terminated. For example, at any point in the configuration history of a Dedicated Host, you can look up how many instances are launched on that host, along with the number of sockets and cores on the host. For any of those instances, you can also look up the ID of its Amazon Machine Image (AMI). You can use this information to report on licensing for your own server-bound software that is licensed per-socket or per-core.

You can view configuration histories in any of the following ways:

- By using the AWS Config console. For each recorded resource, you can view a timeline page, which provides a history of configuration details. To view this page, choose the gray icon in the **Config Timeline** column of the **Dedicated Hosts** page. For more information, see Viewing Configuration Details in the AWS Config Console in the AWS Config Developer Guide.
- By running AWS CLI commands. First, you can use the list-discovered-resources command to get a list of all hosts and instances. Then, you can use the get-resource-config-history command to get the configuration details of a host or instance for a specific time interval. For more information, see View Configuration Details Using the CLI in the AWS Config Developer Guide.
- By using the AWS Config API in your applications. First, you can use the ListDiscoveredResources action to get a list of all hosts and instances. Then, you can use the GetResourceConfigHistory action to get the configuration details of a host or instance for a specific time interval.

For example, to get a list of all of your Dedicated Hosts from AWS Config, run a CLI command such as the following.

```
aws configservice list-discovered-resources --resource-type AWS::EC2::Host
```

To obtain the configuration history of a Dedicated Host from AWS Config, run a CLI command such as the following.

```
aws configservice get-resource-config-history --resource-type AWS::EC2::Instance --resource-id i-1234567890abcdef0
```

**To manage AWS Config settings using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the Dedicated Hosts page, choose Edit Config Recording.
3. In the AWS Config console, follow the steps provided to turn on recording. For more information, see Setting up AWS Config using the Console.

For more information, see Viewing Configuration Details in the AWS Config Console.

**To activate AWS Config using the command line or API**

- AWS CLI: Viewing Configuration Details (AWS CLI) in the AWS Config Developer Guide.
- Amazon EC2 API: GetResourceConfigHistory.

**Dedicated Instances**

Dedicated Instances are Amazon EC2 instances that run in a virtual private cloud (VPC) on hardware that’s dedicated to a single customer. Dedicated Instances that belong to different AWS accounts are physically isolated at a hardware level, even if those accounts are linked to a single payer account.
However, Dedicated Instances may share hardware with other instances from the same AWS account that are not Dedicated Instances.

**Note**

A *Dedicated Host* is also a physical server that's dedicated for your use. With a Dedicated Host, you have visibility and control over how instances are placed on the server. For more information, see Dedicated Hosts (p. 329).

### Dedicated Instance basics

Each instance that you launch into a VPC has a tenancy attribute. This attribute has the following values.

- **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 Change the tenancy of an instance (p. 365).

Each VPC has a related instance tenancy attribute. This attribute has the following values.

- **default**
  
  An instance launched into the VPC runs on shared hardware by default, unless you explicitly specify a different tenancy during instance launch.

- **dedicated**
  
  An instance launched into the VPC is a Dedicated Instance by default, unless you explicitly specify a tenancy of `host` during instance launch. You cannot specify a tenancy of `default` during instance launch.

You can change the instance tenancy of a VPC from `dedicated` to `default` after you create it. You cannot change the instance tenancy of a VPC from `default` to `dedicated` after it is created.

To create Dedicated Instances, you can do the following:

- Create the VPC with the instance tenancy set to `dedicated` (all instances launched into this VPC are Dedicated Instances).
- Create the VPC with the instance tenancy set to `default`, and specify a tenancy of `dedicated` for any instances when you launch them.
Dedicated Instances limitations

Some AWS services or their features won't work with a VPC with the instance tenancy set to dedicated. Check the service's documentation to confirm if there are any limitations.

Some instance types cannot be launched into a VPC with the instance tenancy set to dedicated. For more information about supported instances types, see Amazon EC2 Dedicated Instances.

Amazon EBS with Dedicated Instances

When you launch an Amazon EBS-backed Dedicated Instance, the EBS volume doesn't run on single-tenant hardware.

Reserved Instances with dedicated tenancy

To guarantee that sufficient capacity is available to launch Dedicated Instances, you can purchase Dedicated Reserved Instances. For more information, see Reserved Instances (p. 245).

When you purchase a Dedicated Reserved Instance, you are purchasing the capacity to launch a Dedicated Instance into a VPC at a much reduced usage fee; the price break in the usage charge applies only if you launch an instance with dedicated tenancy. When you purchase a Reserved Instance with default tenancy, it applies only to a running instance with default tenancy; it would not apply to a running instance with dedicated tenancy.

You can't use the modification process to change the tenancy of a Reserved Instance after you've purchased it. However, you can exchange a Convertible Reserved Instance for a new Convertible Reserved Instance with a different tenancy.

Automatic scaling of Dedicated Instances

You can use Amazon EC2 Auto Scaling to launch Dedicated Instances. For more information, see Launching Auto Scaling Instances in a VPC in the Amazon EC2 Auto Scaling User Guide.

Automatic recovery of Dedicated Instances

You can configure automatic recovery for a Dedicated Instances if it becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair. For more information, see Recover your instance (p. 447).

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 Specify a tenancy for your Spot Instances (p. 293).

Pricing for Dedicated Instances

Pricing for Dedicated Instances is different to pricing for On-Demand Instances. For more information, see the Amazon EC2 Dedicated Instances product page.

Burstable performance instances with Dedicated Instances

You can leverage the benefits of running on dedicated tenancy hardware with the section called “Burstable performance instances” (p. 160). T3 Dedicated Instances launch in unlimited mode by default, and they provide a baseline level of CPU performance with the ability to burst to a higher CPU level when required by your workload. The T3 baseline performance and ability to burst are governed by CPU credits. Because of the burstable nature of the T3 instance types, we recommend that you monitor how your T3 instances use the CPU resources of the dedicated hardware for the best performance. T3 Dedicated Instances are intended for customers with diverse workloads that display random CPU behavior, but that ideally have average CPU usage at or below the baseline usages. For more information, see the section called “Key concepts” (p. 162).
Amazon EC2 has systems in place to identify and correct variability in performance. However, it is still possible to experience short term variability if you launch multiple T3 Dedicated Instances that have correlated CPU usage patterns. For these more demanding or correlated workloads, we recommend using M5 or M5a Dedicated Instances rather than T3 Dedicated Instances.

**Work 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**
- Create a VPC with an instance tenancy of dedicated (p. 363)
- Launch Dedicated Instances into a VPC (p. 363)
- Display tenancy information (p. 364)
- Change the tenancy of an instance (p. 365)
- Change the tenancy of a VPC (p. 365)

**Create 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/](https://console.aws.amazon.com/vpc/).
2. From the dashboard, choose *Launch VPC Wizard*.
3. Select a VPC configuration, and then choose *Select*.
4. For *Hardware tenancy*, choose *Dedicated*.
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/](https://console.aws.amazon.com/vpc/).
2. In the navigation pane, choose *Your VPCs*, and then choose *Create VPC*.
3. For *Tenancy*, choose *Dedicated*. Specify the CIDR block, and choose *Create VPC*.

**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.

**Launch 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/](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. For **Tenancy**, choose **Dedicated - Run a dedicated instance**, and then choose **Next: Add Storage**.
6. Continue as prompted by the wizard. When you've finished reviewing your options on the **Review Instance Launch** page, choose **Launch** to choose a key pair and launch the Dedicated Instance.

For more information about launching an instance with a tenancy of **host**, see Launch instances onto a Dedicated Host (p. 336).

**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)

**Display 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 the settings icon (⚙️) in the top-right corner, toggle on **Tenancy**, and choose **Confirm**.

**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 the settings icon (⚙️) in the top-right corner, toggle on **Tenancy**, and choose **Confirm**.
   - Select the instance. On the **Details** tab near the bottom of the page, under **Host and placement group**, check the value for **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)
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)

Change 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 Work with Dedicated Hosts (p. 333). 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 **Instance state**, **Stop instance**, **Stop**.
4. Choose **Actions**, **Instance settings**, **Modify instance placement**.
5. For **Tenancy**, 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)

Change the tenancy of a VPC

You can change the instance tenancy attribute of a VPC from `dedicated` to `default`. Modifying the instance tenancy of the VPC does not affect the tenancy of any existing instances in the VPC. The next time you launch an instance in the VPC, it has a tenancy of `default`, unless you specify otherwise during launch.

You cannot change the tenancy attribute of a VPC from `default` to `dedicated` after it is created.

You can modify the instance tenancy attribute of a VPC using the AWS CLI, an AWS SDK, or the Amazon EC2 API only.

To modify the instance tenancy attribute of a VPC using the AWS CLI

- Use the `modify-vpc-tenancy` command to specify the ID of the VPC and instance tenancy value. The only supported value is `default`.
On-Demand Capacity Reservations

On-Demand Capacity Reservations enable you to reserve compute capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. This gives you the ability to create and manage Capacity Reservations independently from the billing discounts offered by Savings Plans or regional Reserved Instances.

By creating Capacity Reservations, you ensure that you always have access to EC2 capacity when you need it, for as long as you need it. You can create Capacity Reservations at any time, without entering into a one-year or three-year term commitment, and the capacity is available immediately. Billing starts as soon as the capacity is provisioned and the Capacity Reservation enters the active state. When you no longer need it, cancel the Capacity Reservation to stop incurring charges.

When you create a Capacity Reservation, you specify:

- The Availability Zone in which to reserve the capacity
- The number of instances for which to reserve capacity
- The instance attributes, including the instance type, tenancy, and platform/OS

Capacity Reservations can only be used by instances that match their attributes. By default, they are automatically used by running instances that match the attributes. If you don't have any running instances that match the attributes of the Capacity Reservation, it remains unused until you launch an instance with matching attributes.

In addition, you can use Savings Plans and regional Reserved Instances with your Capacity Reservations to benefit from billing discounts. AWS automatically applies your discount when the attributes of a Capacity Reservation match the attributes of a Savings Plan or regional Reserved Instance. For more information, see Billing discounts (p. 369).

Contents

- Differences between Capacity Reservations, Reserved Instances, and Savings Plans (p. 366)
- Supported platforms (p. 367)
- Capacity Reservation limits (p. 367)
- Capacity Reservation limitations and restrictions (p. 368)
- Capacity Reservation pricing and billing (p. 368)
- Work with Capacity Reservations (p. 369)
- Capacity Reservations in Local Zones (p. 378)
- Capacity Reservations in Wavelength Zones (p. 379)
- Capacity Reservations on AWS Outposts (p. 379)
- Work with shared Capacity Reservations (p. 380)
- CloudWatch metrics for On-Demand Capacity Reservations (p. 384)

Differences between Capacity Reservations, Reserved Instances, and Savings Plans

The following table highlights key differences between Capacity Reservations, Reserved Instances, and Savings Plans:

```bash
code
aws ec2 modify-vpc-tenancy --vpc-id vpc-1a2b3c4d --instance-tenancy default
```
Amazon Elastic Compute Cloud
User Guide for Windows Instances
On-Demand Capacity Reservations

<table>
<thead>
<tr>
<th>Capacity Reservations</th>
<th>Zonal Reserved Instances</th>
<th>Regional Reserved Instances</th>
<th>Savings Plans</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Term</strong></td>
<td>No commitment required. Can be created and canceled as needed.</td>
<td>Requires a fixed one-year or three-year commitment</td>
<td></td>
</tr>
<tr>
<td><strong>Capacity benefit</strong></td>
<td>Capacity reserved in a specific Availability Zone.</td>
<td>No capacity reserved.</td>
<td></td>
</tr>
<tr>
<td><strong>Billing discount</strong></td>
<td>No billing discount. †</td>
<td>Provides a billing discount.</td>
<td></td>
</tr>
<tr>
<td><strong>Instance Limits</strong></td>
<td>Your On-Demand Instance limits per Region apply.</td>
<td>Default is 20 per Availability Zone. You can request a limit increase.</td>
<td>Default is 20 per Region. You can request a limit increase.</td>
</tr>
</tbody>
</table>

† You can combine Capacity Reservations with Savings Plans or Regional Reserved Instances to receive a discount.

For more information, see the following:

- Reserved Instances (p. 245)
- Savings Plans User Guide

**Supported platforms**

You must create the Capacity Reservation with the correct platform to ensure that it properly matches with your instances. Capacity Reservations support the following platforms:

- Windows
- Windows with SQL Server
- Windows with SQL Server Web
- Windows with SQL Server Standard
- Windows with SQL Server Enterprise

When you purchase a Capacity Reservation, you must specify the *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 the specific platform.
- For all other Windows versions, excluding BYOL which is not supported, choose the *Windows* platform.

For more information about the supported Linux platforms, see Supported platforms in the Amazon EC2 User Guide for Linux Instances.

**Capacity Reservation limits**

The number of instances for which you are allowed to reserve capacity is based on your account's On-Demand Instance limit. You can reserve capacity for as many instances as that limit allows, minus the number of instances that are already running.

367
Capacity Reservation limitations and restrictions

Before you create Capacity Reservations, take note of the following limitations and restrictions.

• Active and unused Capacity Reservations count toward your On-Demand Instance limits.
• Capacity Reservations are not transferable from one AWS account to another. However, you can share Capacity Reservations with other AWS accounts. For more information, see Work with shared Capacity Reservations (p. 380).
• Zonal Reserved Instance billing discounts do not apply to Capacity Reservations.
• Capacity Reservations can’t be created in placement groups.
• Capacity Reservations can’t be used with Dedicated Hosts.
• Capacity Reservations can’t be used with Bring Your Own License (BYOL).
• Capacity Reservations do not ensure that a hibernated instance can resume after you try to start it.

Capacity Reservation pricing and billing

The price for a Capacity Reservation varies by payment option.

Pricing

When the Capacity Reservation enters the active state, you are charged the equivalent On-Demand rate whether you run instances in the reserved capacity or not. If you do not use the reservation, this shows up as unused reservation on your EC2 bill. When you run an instance that matches the attributes of a reservation, you just pay for the instance and nothing for the reservation. There are no upfront or additional charges.

For example, if you create a Capacity Reservation for 20 m4.large Linux instances and run 15 m4.large Linux instances in the same Availability Zone, you will be charged for 15 active instances and for 5 unused instances in the reservation.

Billing discounts for Savings Plans and regional Reserved Instances apply to Capacity Reservations. For more information, see Billing discounts (p. 369).

For more information, see Amazon EC2 Pricing.

Billing

Billing starts as soon as the capacity is provisioned and the Capacity Reservation enters the active state, and it continues while the Capacity Reservation remains in the active state.

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 canceled 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.
Billing discounts

Billing discounts for Savings Plans and regional Reserved Instances apply to Capacity Reservations. AWS automatically applies these discounts to Capacity Reservations that have matching attributes. When a Capacity Reservation is used by an instance, the discount is applied to the instance. Discounts are preferentially applied to instance usage before covering unused Capacity Reservations.

Billing discounts for zonal Reserved Instances do not apply to Capacity Reservations.

For more information, see the following:

- Reserved Instances (p. 245)
- Savings Plans User Guide

Viewing your bill

You can review the charges and fees to your account on the AWS Billing and Cost Management console.

- The Dashboard displays a spend summary for your account.
- On the Bills page, under Details, expand the Elastic Compute Cloud section and the Region to get billing information about your Capacity Reservations.

You can view the charges online, or you can download a CSV file. For more information, see Capacity Reservation Line Items in the AWS Billing and Cost Management User Guide.

Work with Capacity Reservations

To start using Capacity Reservations, you create the capacity reservation in the required Availability Zone. Then, you can launch instances into the reserved capacity, view its capacity utilization in real time, and increase or decrease its capacity as needed.

By default, Capacity Reservations automatically match new instances and running instances that have matching attributes (instance type, platform, and Availability Zone). This means that any instance with matching attributes automatically runs in the Capacity Reservation. However, you can also target a Capacity Reservation for specific workloads. This enables you to explicitly control which instances are allowed to run in that reserved capacity.

You can specify how the reservation ends. You can choose to cancel the Capacity Reservation or end it automatically at a specified time. If you specify an end time, the Capacity Reservation is canceled within an hour of the specified time. For example, if you specify 5/31/2019, 13:30:55, the Capacity Reservation is guaranteed to end between 13:30:55 and 14:30:55 on 5/31/2019. After a reservation ends, you can no longer target instances to the Capacity Reservation. Instances running in the reserved capacity continue to run uninterrupted. If instances targeting a Capacity Reservation are stopped, you cannot restart them until you remove their Capacity Reservation targeting preference or configure them to target a different Capacity Reservation.
Create a Capacity Reservation

After you create the Capacity Reservation, the capacity is available immediately. The capacity remains reserved for your use as long as the Capacity Reservation is active, and you can launch instances into it at any time. If the Capacity Reservation is open, new instances and existing instances that have matching attributes automatically run in the capacity of the Capacity Reservation. If the Capacity Reservation is targeted, instances must specifically target it to run in the reserved capacity.

Your request to create a Capacity Reservation could fail if one of the following is true:

• Amazon EC2 does not have sufficient capacity to fulfill the request. Either try again at a later time, try a different Availability Zone, or try a smaller capacity. If your application is flexible across instance types and sizes, try different instance attributes.
• The requested quantity exceeds your On-Demand Instance limit for the selected instance family. Increase your On-Demand Instance limit for the instance family and try again. For more information, see On-Demand Instance limits (p. 242).

To create a Capacity Reservation using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose Capacity Reservations, and then choose Create Capacity Reservation.
3. On the Create a Capacity Reservation page, configure the following settings in the Instance details section. The instance type, platform, and Availability Zone of the instances that you launch must match the instance type, platform, and Availability Zone that you specify here or the Capacity Reservation is not applied. For example, if an open Capacity Reservation doesn't match, an instance launch that targets that Capacity Reservation explicitly will fail.
   a. Instance Type—The type of instance to launch into the reserved capacity.
   b. Launch EBS-optimized instances—Specify whether to reserve the capacity for EBS-optimized instances. This option is selected by default for some instance types. For more information about EBS-optimized instances, see Amazon Elastic Block Store (p. 1159).
   c. Attach instance store at launch—Specify whether instances launched into the Capacity Reservation use temporary block-level storage. The data on an instance store volume persists only during the life of the associated instance.
   d. Platform—The operating system for your instances. For more information, see Supported platforms (p. 367). For more information about the supported Linux platforms, see Supported platforms in the Amazon EC2 User Guide for Linux Instances.
   e. Availability Zone—The Availability Zone in which to reserve the capacity.
   f. Tenancy—Specify whether to run on shared hardware (default) or a dedicated instance.
   g. Quantity—The number of instances for which to reserve capacity. If you specify a quantity that exceeds your remaining On-Demand Instance limit for the selected instance type, the request is denied.
4. Configure the following settings in the Reservation details section:
On-Demand Capacity Reservations

a. **Reservation Ends**—Choose one of the following options:
   - **Manually**—Reserve the capacity until you explicitly cancel it.
   - **Specific time**—Cancel the capacity reservation automatically at the specified date and time.

b. **Instance eligibility**—Choose one of the following options:
   - **open**—(Default) The Capacity Reservation matches any instance that has matching attributes (instance type, platform, and Availability Zone). If you launch an instance with matching attributes, it is placed into the reserved capacity automatically.
   - **targeted**—The Capacity Reservation only accepts instances that have matching attributes (instance type, platform, and Availability Zone), and that explicitly target the reservation.

5. Choose *Request reservation*.

To create a Capacity Reservation using the AWS CLI

Use the `create-capacity-reservation` command. For more information, see Supported platforms (p. 367). For more information about the supported Linux platforms, see Supported platforms in the *Amazon EC2 User Guide for Linux Instances*.

For example, the following command creates a Capacity Reservation that reserves capacity for three m5.2xlarge instances running Windows with SQL Server AMIs in the us-east-1a Availability Zone.

```
aws ec2 create-capacity-reservation --instance-type m5.2xlarge --instance-platform Windows with SQL Server --availability-zone us-east-1a --instance-count 3
```

Work with Capacity Reservation groups

You can use AWS Resource Groups to create logical collections of Capacity Reservations, called *resource groups*. A resource group is a logical grouping of AWS resources that are all in the same AWS Region. You can include multiple Capacity Reservations that have different attributes (instance type, platform, and Availability Zone) in a single resource group.

When you create resource groups for your Capacity Reservations, you can target instances to a group of Capacity Reservations instead of an individual Capacity Reservation. Instances that target a group of Capacity Reservations match with any Capacity Reservation in the group that has matching attributes (instance type, platform, and Availability Zone) and available capacity. If the group does not have a Capacity Reservation with matching attributes and available capacity, the instances run using On-Demand capacity. If a matching Capacity Reservation is added to the targeted group at a later stage, the instance is automatically matched with and moved into its reserved capacity.

To prevent unintended use of Capacity Reservations in a group, configure the Capacity Reservations in the group to accept only instances that explicitly target the capacity reservation. To do this, set **Instance eligibility** to **targeted** (old console) or **Only instances that specify this reservation** (new console) when creating the Capacity Reservation using the Amazon EC2 console. When using the AWS CLI, specify `--instance-match-criteria targeted` when creating the Capacity Reservation. Doing this ensures that only instances that explicitly target the group, or a Capacity Reservation in the group, can run in the group.

If a Capacity Reservation in a group is canceled or expires while it has running instances, the instances are automatically moved to another Capacity Reservation in the group that has matching attributes and available capacity. If there are no remaining Capacity Reservations in the group that have matching attributes and available capacity, the instances run in On-Demand capacity. If a matching Capacity Reservation is added to the targeted group at a later stage, the instance is automatically moved into its reserved capacity.

**To create a group for your Capacity Reservations**
Use the `create-group` AWS CLI command. For **name**, provide a descriptive name for the group, and for **configuration**, specify two **Type** request parameters:

- AWS::EC2::CapacityReservationPool to ensure that the resource group can be targeted for instance launches
- AWS::ResourceGroups::Generic with **allowed-resource-types** set to AWS::EC2::CapacityReservation to ensure that the resource group accepts Capacity Reservations only

For example, the following command creates a group named `MyCRGroup`.

```bash
C:\> aws resource-groups create-group --name MyCRGroup --configuration '{"Type":"AWS::EC2::CapacityReservationPool"}', '{"Type":"AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-resource-types", "Values": ["AWS::EC2::CapacityReservation"]}]}
```

The following shows example output.

```json
{
  "GroupConfiguration": {
    "Status": "UPDATE_COMPLETE",
    "Configuration": [
      {
        "Type": "AWS::EC2::CapacityReservationPool"
      },
      {
        "Type": "AWS::ResourceGroups::Generic",
        "Parameters": [
          {
            "Values": [
              "AWS::EC2::CapacityReservation"
            ],
            "Name": "allowed-resource-types"
          }
        ]
      }
    ],
    "Group": {
      "GroupArn": "arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup",
      "Name": "MyCRGroup"
    }
  }
}
```

**To add a Capacity Reservation to a group**

Use the `group-resources` AWS CLI command. For **group**, specify the name of the group to which to add the Capacity Reservations, and for **resources**, specify ARNs of the Capacity Reservations to add. To add multiple Capacity Reservations, separate the ARNs with a space. To get the ARNs of the Capacity Reservations to add, use the `describe-capacity-reservations` AWS CLI command and specify the IDs of the Capacity Reservations.

For example, the following command adds two Capacity Reservations to a group named `MyCRGroup`.

```bash
```

The following shows example output.
To view the Capacity Reservations in a specific group

Use the list-group-resources AWS CLI command. For group, specify the name of the group.

For example, the following command lists the Capacity Reservations in a group named MyCRGroup.

C:\> aws resource-groups list-group-resources --group MyCRGroup

The following shows example output.

```
{
   "QueryErrors": [],
   "ResourceIdentifiers": [
      {
         "ResourceType": "AWS::EC2::CapacityReservation",
      },
      {
         "ResourceType": "AWS::EC2::CapacityReservation",
      }
   ]
}
```

To view the groups to which a specific Capacity Reservation has been added (AWS CLI)

Use the get-groups-for-capacity-reservation AWS CLI command.

For example, the following command lists the groups to which Capacity Reservation cr-1234567890abcdef1 has been added.

C:\> aws ec2 get-groups-for-capacity-reservation --capacity-reservation-id cr-1234567890abcdef1

The following shows example output.

```
{
   "CapacityReservationGroups": [
      {
         "OwnerId": "123456789012",
         "GroupArn": "arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup"
      }
   ]
}
```

To view the groups to which a specific Capacity Reservation has been added (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Capacity Reservations**, select the Capacity Reservation to view, and then choose **View**.

   The groups to which the Capacity Reservation has been added are listed in the **Groups** card.

**To remove a Capacity Reservation from a group**

Use the `ungroup-resources` AWS CLI command. For `group`, specify the ARN of the group from which to remove the Capacity Reservation, and for `resources` specify the ARNs of the Capacity Reservations to remove. To remove multiple Capacity Reservations, separate the ARNs with a space.

The following example removes two Capacity Reservations from a group named `MyCRGroup`.

```
```

The following shows example output.

```
{
  "Failed": [],
  "Succeeded": [
    "arn:aws:ec2:sa-east-1:123456789012:capacity-reservation/cr-0e154d26a16094dd",
  ]
}
```

**To delete a group**

Use the `delete-group` AWS CLI command. For `group`, provide the name of the group to delete.

For example, the following command deletes a group named `MyCRGroup`.

```
C:\> aws resource-groups delete-group --group MyCRGroup
```

The following shows example output.

```
{
  "Group": {
    "GroupArn": "arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup",
    "Name": "MyCRGroup"
  }
}
```

**Launch instances into an existing Capacity Reservation**

When you launch an instance, you can specify whether to launch the instance into any **open** Capacity Reservation, into a specific Capacity Reservation, or into a group of Capacity Reservations. You can only launch an instance into a Capacity Reservation that has matching attributes (instance type, platform, and Availability Zone) and sufficient capacity. Alternatively, you can configure the instance to avoid running in a Capacity Reservation, even if you have an **open** Capacity Reservation that has matching attributes and available capacity.

Launching an instance into a Capacity Reservation reduces its available capacity by the number of instances launched. For example, if you launch three instances, the available capacity of the Capacity Reservation is reduced by three.
To launch instances into an existing Capacity Reservation using the console

1. Open the Launch Instance wizard by choosing Launch Instances from Dashboard or Instances.
2. Select an Amazon Machine Image (AMI) and an instance type.
3. Complete the Configure Instance Details page. For Capacity Reservation, choose one of the following options:
   - **None** — Prevents the instances from launching into a Capacity Reservation. The instances run in On-Demand capacity.
   - **Open** — Launches the instances into any Capacity Reservation that has matching attributes and sufficient capacity for the number of instances you selected. If there is no matching Capacity Reservation with sufficient capacity, the instance uses On-Demand capacity.
   - **Target by ID** — Launches the instances into the selected Capacity Reservation. If the selected Capacity Reservation does not have sufficient capacity for the number of instances you selected, the instance launch fails.
   - **Target by group** — Launches the instances into any Capacity Reservation with matching attributes and available capacity in the selected Capacity Reservation group. If the selected group does not have a Capacity Reservation with matching attributes and available capacity, the instances launch into On-Demand capacity.
4. Complete the remaining steps to launch the instances.

To launch an instance into an existing Capacity Reservation using the AWS CLI

Use the run-instances command and specify the --capacity-reservation-specification parameter.

The following example launches a `t2.micro` instance into any open Capacity Reservation that has matching attributes and available capacity:

```bash
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --key-name MyKeyPair --subnet-id subnet-1234567890abcdef1 --capacity-reservation-specification CapacityReservationPreference=open
```

The following example launches a `t2.micro` instance into a targeted Capacity Reservation:

```bash
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --key-name MyKeyPair --subnet-id subnet-1234567890abcdef1 --capacity-reservation-specification CapacityReservationTarget={CapacityReservationId=cr-a1234567}
```

The following example launches a `t2.micro` instance into a Capacity Reservation group:

```bash
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --key-name MyKeyPair --subnet-id subnet-1234567890abcdef1 --capacity-reservation-specification CapacityReservationTarget={CapacityReservationResourceId=cr-a1234567 CapacityReservationResourceGroupId=arn:aws:resource-groups:us-west-1:123456789012:group/my-cr-group}
```

Modify a Capacity Reservation

You can change the attributes of an active Capacity Reservation after you have created it. You cannot modify a Capacity Reservation after it has expired or after you have explicitly canceled it.

When modifying a Capacity Reservation, you can only increase or decrease the quantity and change the way in which it is released. You cannot change the instance type, EBS optimization, instance store settings, platform, Availability Zone, or instance eligibility of a Capacity Reservation. If you need to modify any of these attributes, we recommend that you cancel the reservation, and then create a new one with the required attributes.
If you specify a new quantity that exceeds your remaining On-Demand Instance limit for the selected instance type, the update fails.

**To modify a Capacity Reservation using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. Choose *Capacity Reservations*, select the Capacity Reservation to modify, and then choose *Edit*.
3. Modify the *Quantity* or *Reservation ends* options as needed, and choose *Save changes*.

**To modify a Capacity Reservation using the AWS CLI**

Use the `modify-capacity-reservations` command:

For example, the following command modifies a Capacity Reservation to reserve capacity for eight instances.

```bash
aws ec2 modify-capacity-reservation --capacity-reservation-id cr-1234567890abcdef0 --instance-count 8
```

**Modify an instance's Capacity Reservation settings**

You can modify the following Capacity Reservation settings for a stopped instance at any time:

- Start in any Capacity Reservation that has matching attributes (instance type, platform, and Availability Zone) and available capacity.
- Start the instance in a specific Capacity Reservation.
- Start the in any Capacity Reservation that has matching attributes and available capacity in a Capacity Reservation group
- Prevent the instance from starting in a Capacity Reservation.

**To modify an instance's Capacity Reservation settings using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. Choose *Instances* and select the instance to modify. Stop the instance if it is not already stopped.
3. Choose *Actions, Modify Capacity Reservation Settings*.
4. For *Capacity Reservation*, choose one of the following options:

   - **Open** — Launches the instances into any Capacity Reservation that has matching attributes and sufficient capacity for the number of instances you selected. If there is no matching Capacity Reservation with sufficient capacity, the instance uses On-Demand capacity.
   - **None** — Prevents the instances from launching into a Capacity Reservation. The instances run in On-Demand capacity.
   - **Specify Capacity Reservation** — Launches the instances into the selected Capacity Reservation. If the selected Capacity Reservation does not have sufficient capacity for the number of instances you selected, the instance launch fails.
   - **Specify Capacity Reservation group** — Launches the instances into any Capacity Reservation with matching attributes and available capacity in the selected Capacity Reservation group. If the selected group does not have a Capacity Reservation with matching attributes and available capacity, the instances launch into On-Demand capacity.

**To modify an instance's Capacity Reservation settings using the AWS CLI**

Use the `modify-instance-capacity-reservation-attributes` command.
For example, the following command changes an instance's Capacity Reservation setting to open or none.

```
aws ec2 modify-instance-capacity-reservation-attributes --instance-id i-1234567890abcdef0
   --capacity-reservation-specification CapacityReservationPreference=none|open
```

For example, the following command modifies an instance to target a specific Capacity Reservation.

```
aws ec2 modify-instance-capacity-reservation-attributes --instance-id
   i-1234567890abcdef0 --capacity-reservation-specification
   CapacityReservationTarget={CapacityReservationId=cr-1234567890abcdef0}
```

For example, the following command modifies an instance to target a specific Capacity Reservation group.

```
aws ec2 modify-instance-capacity-reservation-attributes --instance-id
   i-1234567890abcdef0 --capacity-reservation-specification
   CapacityReservationTarget={CapacityReservationResourceGroupArn=arn:aws:resource-groups:us-west-1:123456789012:group/my-cr-group}
```

View a Capacity Reservation

Capacity Reservations have the following possible states:

- **active**—The capacity is available for use.
- **expired**—The Capacity Reservation expired automatically at the date and time specified in your reservation request. The reserved capacity is no longer available for your use.
- **cancelled**—The Capacity Reservation was canceled. The reserved capacity is no longer available for your use.
- **pending**—The Capacity Reservation request was successful but the capacity provisioning is still pending.
- **failed**—The Capacity Reservation request has failed. A request can fail due to invalid request parameters, capacity constraints, or instance limit constraints. You can view a failed request for 60 minutes.

To view your Capacity Reservations using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose Capacity Reservations and select a Capacity Reservation to view.
3. Choose View launched instances for this reservation.

To view your Capacity Reservations using the AWS CLI

Use the describe-capacity-reservations command:

```
aws ec2 describe-capacity-reservations
```

Cancel a Capacity Reservation

You can cancel a Capacity Reservation at any time if you no longer need the reserved capacity. When you cancel a Capacity Reservation, the capacity is released immediately, and it is no longer reserved for your use.
You can cancel empty Capacity Reservations and Capacity Reservations that have running instances. If you cancel a Capacity Reservation that has running instances, the instances continue to run normally outside of the capacity reservation at standard On-Demand Instance rates or at a discounted rate if you have a matching Savings Plan or regional Reserved Instance.

After you cancel a Capacity Reservation, instances that target it can no longer launch. Modify these instances so that they either target a different Capacity Reservation, launch into any open Capacity Reservation with matching attributes and sufficient capacity, or avoid launching into a Capacity Reservation. For more information, see Modify an instance's Capacity Reservation settings (p. 376).

To cancel a Capacity Reservation using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose Capacity Reservations and select the Capacity Reservation to cancel.
3. Choose Cancel reservation, Cancel reservation.

To cancel a Capacity Reservation using the AWS CLI

Use the cancel-capacity-reservation command:

For example, the following command cancels a Capacity Reservation with an ID of cr-1234567890abcdef0.

```bash
aws ec2 cancel-capacity-reservation --capacity-reservation-id cr-1234567890abcdef0
```

Capacity Reservations in Local Zones

A Local Zone is an extension of an AWS Region that is geographically close to your users. Resources created in a Local Zone can serve local users with very low-latency communications. For more information, see AWS Local Zones.

You can extend a VPC from its parent AWS Region into a Local Zone by creating a new subnet in that Local Zone. When you create a subnet in a Local Zone, your VPC is extended to that Local Zone. The subnet in the Local Zone operates the same as the other subnets in your VPC.

By using Local Zones, you can place Capacity Reservations in multiple locations that are closer to your users. You create and use Capacity Reservations in Local Zones in the same way that you create and use Capacity Reservations in regular Availability Zones. The same features and instance matching behavior apply. For more information about the pricing models that are supported in Local Zones, see AWS Local Zones FAQs.

Considerations

You can't use Capacity Reservation groups in a Local Zone.

To use a Capacity Reservation in a Local Zone

1. Enable the Local Zone for use in your AWS account. For more information, see Enable Local Zones in the Amazon EC2 User Guide for Linux Instances.
2. Create a Capacity Reservation in the Local Zone. For Availability Zone, choose the Local Zone. The Local Zone is represented by an AWS Region code followed by an identifier that indicates the location, for example us-west-2-lax-1a. For more information, see Create a Capacity Reservation (p. 370).
3. Create a subnet in the Local Zone. For Availability Zone, choose the Local Zone. For more information, see Creating a subnet in your VPC in the Amazon VPC User Guide.
4. Launch an instance. For **Subnet**, choose the subnet in the Local Zone (for example subnet-123abc | us-west-2-lax-1a), and for **Capacity Reservation**, choose the specification (either open or target it by ID) that's required for the Capacity Reservation that you created in the Local Zone. For more information, see Launch instances into an existing Capacity Reservation (p. 374).

### Capacity Reservations in Wavelength Zones

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

When you create On-Demand Capacity Reservations, you can choose the Wavelength Zone and you can launch instances into a Capacity Reservation in a Wavelength Zone by specifying the subnet associated with the Wavelength Zone. A Wavelength Zone is represented by an AWS Region code followed by an identifier that indicates the location, for example `us-east-1-w11-bos-wlz-1`.

Wavelength Zones are not available in every Region. For information about the Regions that support Wavelength Zones, see Available Wavelength Zones in the AWS Wavelength Developer Guide.

**Considerations**

You can't use Capacity Reservation groups in a Wavelength Zone.

**To use a Capacity Reservation in a Wavelength Zone**

1. Enable the Wavelength Zone for use in your AWS account. For more information, see Enable Wavelength Zones in the Amazon EC2 User Guide for Linux Instances.
2. Create a Capacity Reservation in the Wavelength Zone. For **Availability Zone**, choose the Wavelength. The Wavelength is represented by an AWS Region code followed by an identifier that indicates the location, for example `us-east-1-w11-bos-wlz-1`. For more information, see Create a Capacity Reservation (p. 370).
3. Create a subnet in the Wavelength Zone. For **Availability Zone**, choose the Wavelength Zone. For more information, see Creating a subnet in your VPC in the Amazon VPC User Guide.
4. Launch an instance. For **Subnet**, choose the subnet in the Wavelength Zone (for example subnet-123abc | us-east-1-w11-bos-wlz-1), and for **Capacity Reservation**, choose the specification (either open or target it by ID) that's required for the Capacity Reservation that you created in the Wavelength. For more information, see Launch instances into an existing Capacity Reservation (p. 374).

### Capacity Reservations on AWS Outposts

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

An Outpost is a pool of AWS compute and storage capacity deployed at a customer site. AWS operates, monitors, and manages this capacity as part of an AWS Region.

You can create Capacity Reservations on Outposts that you have created in your account. This allows you to reserve compute capacity on an Outpost at your site. You create and use Capacity Reservations on
Outposts in the same way that you create and use Capacity Reservations in regular Availability Zones. The same features and instance matching behavior apply.

You can also share Capacity Reservations on Outposts with other AWS accounts within your organization using AWS Resource Access Manager. For more information about sharing Capacity Reservations, see Work with shared Capacity Reservations (p. 380).

Prerequisite

You must have an Outpost installed at your site. For more information, see Create an Outpost and order Outpost capacity in the AWS Outposts User Guide.

Considerations

- You can't use Capacity Reservation groups on an Outpost.

**To use a Capacity Reservation on an Outpost**

1. Create a subnet on the Outpost. For more information, see Create a subnet in the AWS Outposts User Guide.
2. Create a Capacity Reservation on the Outpost.
   a. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
   b. In the navigation pane, choose **Outposts**, and then choose **Actions, Create Capacity Reservation**.
   c. Configure the Capacity Reservation as needed and then choose **Create**. For more information, see Create a Capacity Reservation (p. 370).

   **Note**

   The **Instance Type** drop-down lists only instance types that are supported by the selected Outpost, and the **Availability Zone** drop-down lists only the Availability Zone with which the selected Outpost is associated.

3. Launch an instance into the Capacity Reservation. For **Subnet** choose the subnet that you created in Step 1, and for **Capacity Reservation**, select the Capacity Reservation that you created in Step 2. For more information, see Launch an instance on the Outpost in the AWS Outposts User Guide.

**Work with shared Capacity Reservations**

Capacity Reservation sharing enables Capacity Reservation owners to share their reserved capacity with other AWS accounts or within an AWS organization. This enables you to create and manage Capacity Reservations centrally, and share the reserved capacity across multiple AWS accounts or within your AWS organization.

In this model, the AWS account that owns the Capacity Reservation (owner) shares it with other AWS accounts (consumers). Consumers can launch instances into Capacity Reservations that are shared with them in the same way that they launch instances into Capacity Reservations that they own in their own account. The Capacity Reservation owner is responsible for managing the Capacity Reservation and the instances that they launch into it. Owners cannot modify instances that consumers launch into Capacity Reservations that they have shared. Consumers are responsible for managing the instances that they launch into Capacity Reservations shared with them. Consumers cannot view or modify instances owned by other consumers or by the Capacity Reservation owner.

A Capacity Reservation owner can share a Capacity Reservation with:

- Specific AWS accounts inside or outside of its AWS organization
- An organizational unit inside its AWS organization

380
• Its entire AWS organization

Contents
• Prerequisites for sharing Capacity Reservations (p. 381)
• Related services (p. 381)
• Share across Availability Zones (p. 381)
• Share a Capacity Reservation (p. 382)
• Stop sharing a Capacity Reservation (p. 382)
• Identify a shared Capacity Reservation (p. 383)
• View shared Capacity Reservation usage (p. 383)
• Shared Capacity Reservation permissions (p. 384)
• Billing and metering (p. 384)
• Instance limits (p. 384)

Prerequisites for sharing Capacity Reservations

• To share a Capacity Reservation, you must own it in your AWS account. You cannot share a Capacity Reservation that has been shared with you.
• You can only share Capacity Reservations for shared tenancy instances. You cannot share Capacity Reservations for dedicated tenancy instances.
• Capacity Reservation sharing is not available to new AWS accounts or AWS accounts that have a limited billing history.
• To share a Capacity Reservation with your AWS organization or an organizational unit in your AWS organization, you must enable sharing with AWS Organizations. For more information, see Enable Sharing with AWS Organizations in the AWS RAM User Guide.

Related services

Capacity Reservation sharing integrates with AWS Resource Access Manager (AWS RAM). AWS RAM is a service that enables you to share your AWS resources with any AWS account or through AWS Organizations. With AWS RAM, you share resources that you own by creating a resource share. A resource share specifies the resources to share, and the consumers with whom to share them. Consumers can be individual AWS accounts, or organizational units or an entire organization from AWS Organizations.

For more information about AWS RAM, see the AWS RAM User Guide.

Share across Availability Zones

To ensure that resources are distributed across the Availability Zones for a Region, we independently map Availability Zones to names for each account. This could lead to Availability Zone naming differences across accounts. For example, the Availability Zone us-east-1a for your AWS account might not have the same location as us-east-1a for another AWS account.

To identify the location of your Capacity Reservations relative to your accounts, you must use the Availability Zone ID (AZ ID). The AZ ID is a unique and consistent identifier for an Availability Zone across all AWS accounts. For example, use1-az1 is an AZ ID for the us-east-1 Region and it is the same location in every AWS account.

To view the AZ IDs for the Availability Zones in your account

2. The AZ IDs for the current Region are displayed in the **Your AZ ID** panel on the right-hand side of the screen.

### Share a Capacity Reservation

When you share a Capacity Reservation that you own with other AWS accounts, you enable them to launch instances into your reserved capacity. If you share an open Capacity Reservation, keep the following in mind as it could lead to unintended Capacity Reservation usage:

- If consumers have running instances that match the attributes of the Capacity Reservation, have the `CapacityReservationPreference` parameter set to `open`, and are not yet running in reserved capacity, they automatically use the shared Capacity Reservation.
- If consumers launch instances that have matching attributes (instance type, platform, and Availability Zone) and have the `CapacityReservationPreference` parameter set to `open`, they automatically launch into the shared Capacity Reservation.

To share a Capacity Reservation, you must add it to a resource share. A resource share is an AWS RAM resource that lets you share your resources across AWS accounts. A resource share specifies the resources to share, and the consumers with whom they are shared. When you share a Capacity Reservation using the Amazon EC2 console, you add it to an existing resource share. To add the Capacity Reservation to a new resource share, you must create the resource share using the **AWS RAM console**.

If you are part of an organization in AWS Organizations and sharing within your organization is enabled, consumers in your organization are automatically granted access to the shared Capacity Reservation. Otherwise, consumers receive an invitation to join the resource share and are granted access to the shared Capacity Reservation after accepting the invitation.

You can share a Capacity Reservation that you own using the Amazon EC2 console, **AWS RAM console**, or **the AWS CLI**.

#### To share a Capacity Reservation that you own using the Amazon EC2 console

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Capacity Reservations**.
3. Choose the Capacity Reservation to share and choose **Actions, Share reservation**.
4. Select the resource share to which to add the Capacity Reservation and choose **Share Capacity Reservation**.

   It could take a few minutes for consumers to get access to the shared Capacity Reservation.

#### To share a Capacity Reservation that you own using the AWS RAM console


#### To share a Capacity Reservation that you own using the AWS CLI

Use the `create-resource-share` command.

### Stop sharing a Capacity Reservation

The Capacity Reservation owner can stop sharing a Capacity Reservation at any time. The following rules apply:

- Instances owned by consumers that were running in the shared capacity at the time sharing stops continue to run normally outside of the reserved capacity, and the capacity is restored to the Capacity Reservation subject to Amazon EC2 capacity availability.
• Consumers with whom the Capacity Reservation was shared can no longer launch new instances into
the reserved capacity.

To stop sharing a Capacity Reservation that you own, you must remove it from the resource share. You
can do this using the Amazon EC2 console, AWS RAM console, or the AWS CLI.

**To stop sharing a Capacity Reservation that you own using the Amazon EC2 console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Capacity Reservations**.
3. Select the Capacity Reservation and choose the **Sharing** tab.
4. The **Sharing** tab lists the resource shares to which the Capacity Reservation has been added. Select
the resource share from which to remove the Capacity Reservation and choose **Remove from
resource share**.

**To stop sharing a Capacity Reservation that you own using the AWS RAM console**

See Updating a Resource Share in the **AWS RAM User Guide**.

**To stop sharing a Capacity Reservation that you own using the AWS CLI**

Use the **disassociate-resource-share** command.

**Identify a shared Capacity Reservation**

Owners and consumers can identify shared Capacity Reservations using the Amazon EC2 console and
AWS CLI

**To identify a shared Capacity Reservation using the Amazon EC2 console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Capacity Reservations**. The screen lists Capacity Reservations that
you own and Capacity Reservations that are shared with you. The **Owner** column shows the AWS
account ID of the Capacity Reservation owner. (me) next to the AWS account ID indicates that you
are the owner.

**To identify a shared Capacity Reservation using the AWS CLI**

Use the **describe-capacity-reservations** command. The command returns the Capacity Reservations that
you own and Capacity Reservations that are shared with you. **OwnerId** shows the AWS account ID of the
Capacity Reservation owner.

**View shared Capacity Reservation usage**

The owner of a shared Capacity Reservation can view its usage at any time using the Amazon EC2
console and the AWS CLI.

**To view Capacity Reservation usage using the Amazon EC2 console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Capacity Reservations**.
3. Select the Capacity Reservation for which to view the usage and choose the **Usage** tab.

   The **AWS account ID** column shows the account IDs of the consumers currently using the Capacity
Reservation. The **Launched instances** column shows the number of instances each consumer
currently has running in the reserved capacity.
To view Capacity Reservation usage using the AWS CLI

Use the `get-capacity-reservation-usage` command. `AccountId` shows the account ID of the account using the Capacity Reservation. `UsedInstanceCount` shows the number of instances the consumer currently has running in the reserved capacity.

Shared Capacity Reservation permissions

Permissions for owners

Owners are responsible for managing and canceling their shared Capacity Reservations. Owners cannot modify instances running in the shared Capacity Reservation that are owned by other accounts. Owners remain responsible for managing instances that they launch into the shared Capacity Reservation.

Permissions for consumers

Consumers are responsible for managing their instances that are running the shared Capacity Reservation. Consumers cannot modify the shared Capacity Reservation in any way, and they cannot view or modify instances that are owned by other consumers or the Capacity Reservation owner.

Billing and metering

There are no additional charges for sharing Capacity Reservations.

The Capacity Reservation owner is billed for instances that they run inside the Capacity Reservation and for unused reserved capacity. Consumers are billed for the instances that they run inside the shared Capacity Reservation.

Instance limits

All Capacity Reservation usage counts toward the Capacity Reservation owner's On-Demand Instance limits. This includes:

- Unused reserved capacity
- Usage by instances owned by the Capacity Reservation owner
- Usage by instances owned by consumers

Instances launched into the shared capacity by consumers count towards the Capacity Reservation owner's On-Demand Instance limit. Consumers' instance limits are a sum of their own On-Demand Instance limits and the capacity available in the shared Capacity Reservations to which they have access.

CloudWatch metrics for On-Demand Capacity Reservations

With CloudWatch metrics, you can efficiently monitor your Capacity Reservations and identify unused capacity by setting CloudWatch alarms to notify you when usage thresholds are met. This can help you maintain a constant Capacity Reservation volume and achieve a higher level of utilization.

On-Demand Capacity Reservations send metric data to CloudWatch every five minutes. Metrics are not supported for Capacity Reservations that are active for less than five minutes.

For more information about viewing metrics in the CloudWatch console, see Using Amazon CloudWatch Metrics. For more information about creating alarms, see Creating Amazon CloudWatch Alarms.

Contents

- Capacity Reservation usage metrics (p. 385)
Capacity Reservation usage metrics

The AWS/EC2CapacityReservations namespace includes the following usage metrics you can use to monitor and maintain on-demand capacity within thresholds you specify for your reservation.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UsedInstanceCount</td>
<td>The number of instances that are currently in use.</td>
</tr>
<tr>
<td></td>
<td>Unit: Count</td>
</tr>
<tr>
<td>AvailableInstanceCount</td>
<td>The number of instances that are available.</td>
</tr>
<tr>
<td></td>
<td>Unit: Count</td>
</tr>
<tr>
<td>TotalInstanceCount</td>
<td>The total number of instances you have reserved.</td>
</tr>
<tr>
<td></td>
<td>Unit: Count</td>
</tr>
<tr>
<td>InstanceUtilization</td>
<td>The percentage of reserved capacity instances that are currently in use.</td>
</tr>
<tr>
<td></td>
<td>Unit: Percent</td>
</tr>
</tbody>
</table>

Capacity Reservation metric dimensions

You can use the following dimensions to refine the metrics listed in the previous table.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapacityReservationId</td>
<td>This globally unique dimension filters the data you request for the identified capacity reservation only.</td>
</tr>
</tbody>
</table>

View CloudWatch metrics for Capacity Reservations

Metrics are grouped first by the service namespace, and then by the supported dimensions. You can use the following procedures to view the metrics for your Capacity Reservations.

To view Capacity Reservation metrics using the CloudWatch console

2. If necessary, change the Region. From the navigation bar, select the Region where your Capacity Reservation resides. For more information, see Regions and Endpoints.
3. In the navigation pane, choose Metrics.
4. For All metrics, choose EC2 Capacity Reservations.
5. Choose the metric dimension By Capacity Reservation. Metrics will be grouped by CapacityReservationId.
6. To sort the metrics, use the column heading. To graph a metric, select the check box next to the metric.
To view Capacity Reservation metrics (AWS CLI)

Use the following `list-metrics` command:

```bash
aws cloudwatch list-metrics --namespace "AWS/EC2CapacityReservations"
```

 Instance lifecycle

An Amazon EC2 instance transitions through different states from the moment you launch it through to its termination.

The following illustration represents the transitions between instance states.

![Instance lifecycle diagram]

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.

<table>
<thead>
<tr>
<th>Instance state</th>
<th>Description</th>
<th>Instance usage billing</th>
</tr>
</thead>
<tbody>
<tr>
<td>pending</td>
<td>The instance is preparing to enter the running state. An instance enters the pending state.</td>
<td>Not billed</td>
</tr>
</tbody>
</table>
### Instance state | Description | Instance usage billing |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Running</strong></td>
<td>The instance is running and ready for use.</td>
<td>Billed</td>
</tr>
<tr>
<td><strong>Stopping</strong></td>
<td>The instance is preparing to be stopped or stop- hibernated.</td>
<td>Not billed if preparing to stop&lt;br&gt;Billed if preparing to hibernate</td>
</tr>
<tr>
<td><strong>Stopped</strong></td>
<td>The instance is shut down and cannot be used. The instance can be started at any time.</td>
<td>Not billed</td>
</tr>
<tr>
<td><strong>Shutting down</strong></td>
<td>The instance is preparing to be terminated.</td>
<td>Not billed</td>
</tr>
<tr>
<td><strong>Terminated</strong></td>
<td>The instance has been permanently deleted and cannot be started.</td>
<td>Not billed</td>
</tr>
</tbody>
</table>

### Note
Rebooting an instance doesn't start a new instance billing period because the instance stays in the running state.

## Instance launch

When you launch an instance, it enters the pending state. The instance type that you specified at launch determines the hardware of the host computer for your instance. We use the Amazon Machine Image (AMI) you specified at launch to boot the instance. After the instance is ready for you, it enters the running state. You can connect to your running instance and use it the way that you'd use a computer sitting in front of you.

As soon as your instance transitions to the running state, you're billed for each second, with a one- minute minimum, that you keep the instance running, even if the instance remains idle and you don't connect to it.

For more information, see Launch your instance (p. 390) and Connect to your Windows instance (p. 413).

## Instance stop and start (Amazon EBS-backed instances only)

If your instance fails a status check or is not running your applications as expected, and if the root volume of your instance is an Amazon EBS volume, you can stop and start your instance to try to fix the problem.
When you stop your instance, it enters the stopping state, and then the stopped state. We don't charge usage or data transfer fees for your instance after you stop it, but we do charge for the storage for any Amazon EBS volumes. While your instance is in the stopped state, you can modify certain attributes of the instance, including the instance type.

When you start your instance, it enters the pending state, and we move the instance to a new host computer (though in some cases, it remains on the current host). When you stop and start your instance, you lose any data on the instance store volumes on the previous host computer.

Your instance retains its private IPv4 address, which means that an Elastic IP address associated with the private IPv4 address or network interface is still associated with your instance. If your instance has an IPv6 address, it retains its IPv6 address.

Each time you transition an instance from stopped to running, we charge per second when the instance is running, with a minimum of one minute every time you start your instance.

For more information, see Stop and start your instance (p. 425).

### Instance hibernate (Amazon EBS-backed instances only)

When you hibernate an instance, we signal the operating system to perform hibernation (suspend-to-disk), which saves the contents from the instance memory (RAM) to your Amazon EBS root volume. We persist the instance's Amazon EBS root volume and any attached Amazon EBS data volumes. When you start your instance, the Amazon EBS root volume is restored to its previous state and the RAM contents are reloaded. Previously attached data volumes are reattached and the instance retains its instance ID.

When you hibernate your instance, it enters the stopping state, and then the stopped state. We don't charge usage for a hibernated instance when it is in the stopped state, but we do charge while it is in the stopping state, unlike when you stop an instance (p. 387) without hibernating it. We don't charge usage for data transfer fees, but we do charge for the storage for any Amazon EBS volumes, including storage for the RAM data.

When you start your hibernated instance, it enters the pending state, and we move the instance to a new host computer (though in some cases, it remains on the current host).

Your instance retains its private IPv4 address, which means that an Elastic IP address associated with the private IPv4 address or network interface is still associated with your instance. If your instance has an IPv6 address, it retains its IPv6 address.

For more information, see Hibernate your On-Demand or Reserved Windows instance (p. 428).

### Instance reboot

You can reboot your instance using the Amazon EC2 console, a command line tool, and the Amazon EC2 API. We recommend that you use Amazon EC2 to reboot your instance instead of running the operating system reboot command from your instance.

Rebooting an instance is equivalent to rebooting an operating system. The instance remains on the same host computer and maintains its public DNS name, private IP address, and any data on its instance store volumes. It typically takes a few minutes for the reboot to complete, but the time it takes to reboot depends on the instance configuration.

Rebooting an instance doesn't start a new instance billing period; per second billing continues without a further one-minute minimum charge.

For more information, see Reboot your instance (p. 438).
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. 439).

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. 441).

Differences between reboot, stop, hibernate, and terminate

The following table summarizes the key differences between rebooting, stopping, hibernating, and terminating your instance.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Reboot</th>
<th>Stop/start (Amazon EBS-backed instances only)</th>
<th>Hibernate (Amazon EBS-backed instances only)</th>
<th>Terminate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host computer</td>
<td>The instance stays on the same host computer</td>
<td>We move the instance to a new host computer (though in some cases, it remains on the current host).</td>
<td>We move the instance to a new host computer (though in some cases, it remains on the current host).</td>
<td>None</td>
</tr>
<tr>
<td>Private and public IPv4 addresses stay the same</td>
<td>These addresses stay the same</td>
<td>The instance keeps its private IPv4 address. The instance gets a new public IPv4</td>
<td>The instance keeps its private IPv4 address. The instance gets a new public IPv4</td>
<td>None</td>
</tr>
<tr>
<td>Characteristic</td>
<td>Reboot</td>
<td>Stop/start (Amazon EBS-backed instances only)</td>
<td>Hibernate (Amazon EBS-backed instances only)</td>
<td>Terminate</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------</td>
<td>------------------------------------------------</td>
<td>---------------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>address, unless it has an Elastic IP address, which doesn't change during a stop/start.</td>
<td>address, unless it has an Elastic IP address, which doesn't change during a stop/start.</td>
<td></td>
</tr>
<tr>
<td>Elastic IP addresses</td>
<td>The Elastic IP address remains associated with the instance</td>
<td>The Elastic IP address remains associated with the instance</td>
<td>The Elastic IP address remains associated with the instance</td>
<td>The Elastic IP address is disassociated from the instance</td>
</tr>
<tr>
<td>(IPv4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPv6 address</td>
<td>The address stays the same</td>
<td>The instance keeps its IPv6 address</td>
<td>The instance keeps its IPv6 address</td>
<td>None</td>
</tr>
<tr>
<td>Instance store volumes</td>
<td>The data is preserved</td>
<td>The data is erased</td>
<td>The data is erased</td>
<td>The data is erased</td>
</tr>
<tr>
<td>Root device volume</td>
<td>The volume is preserved</td>
<td>The volume is preserved</td>
<td>The volume is preserved</td>
<td>The volume is deleted by default</td>
</tr>
<tr>
<td>RAM (contents of memory)</td>
<td>The RAM is erased</td>
<td>The RAM is erased</td>
<td>The RAM is saved to a file on the root volume</td>
<td>The RAM is erased</td>
</tr>
<tr>
<td>Billing</td>
<td>The instance billing hour doesn't change.</td>
<td>You stop incurring charges for an instance as soon as its state changes to stopping. Each time an instance transitions from stopped to running, we start a new instance billing period, billing a minimum of one minute every time you start your instance.</td>
<td>You incur charges while the instance is in the stopping state, but stop incurring charges when the instance is in the stopped state. Each time an instance transitions from stopped to running, we start a new instance billing period, billing a minimum of one minute every time you start your instance.</td>
<td>You stop incurring charges for an instance as soon as its state changes to shutting-down.</td>
</tr>
</tbody>
</table>

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 Change the instance initiated shutdown behavior (p. 445).

**Launch your instance**

An instance is a virtual server in the AWS Cloud. You launch an instance from an Amazon Machine Image (AMI). The AMI provides the operating system, application server, and applications for your instance.

When you sign up for AWS, you can get started with Amazon EC2 for free using the AWS Free Tier. You can use the free tier to launch and use a t2.micro instance for free for 12 months (in Regions where t2.micro is unavailable, you can use a t3.micro instance under the free tier). If you launch an instance...
that is not within the free tier, you incur the standard Amazon EC2 usage fees for the instance. For more information, see Amazon EC2 pricing.

You can launch an instance using the following methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Amazon EC2 console] Use the launch instance wizard to specify the launch parameters.</td>
<td>Launch an instance using the Launch Instance Wizard (p. 392)</td>
</tr>
<tr>
<td>[Amazon EC2 console] Create a launch template and launch the instance from the launch template.</td>
<td>Launch an instance from a launch template (p. 397)</td>
</tr>
<tr>
<td>[Amazon EC2 console] Use an existing instance as the base.</td>
<td>Launch an instance using parameters from an existing instance (p. 411)</td>
</tr>
<tr>
<td>[Amazon EC2 console] Use an AMI that you purchased from the AWS Marketplace.</td>
<td>Launch an AWS Marketplace instance (p. 412)</td>
</tr>
<tr>
<td>[AWS CLI] Use an AMI that you select.</td>
<td>Using Amazon EC2 through the AWS CLI</td>
</tr>
<tr>
<td>[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.</td>
<td>EC2 Fleet (p. 664)</td>
</tr>
<tr>
<td>[AWS CloudFormation] Use a AWS CloudFormation template to specify an instance.</td>
<td>AWS::EC2::Instance in the AWS CloudFormation User Guide</td>
</tr>
<tr>
<td>[AWS SDK] Use a language-specific AWS SDK to launch an instance.</td>
<td>AWS SDK for .NET</td>
</tr>
<tr>
<td></td>
<td>AWS SDK for C++</td>
</tr>
<tr>
<td></td>
<td>AWS SDK for Go</td>
</tr>
<tr>
<td></td>
<td>AWS SDK for Java</td>
</tr>
<tr>
<td></td>
<td>AWS SDK for JavaScript</td>
</tr>
<tr>
<td></td>
<td>AWS SDK for PHP V3</td>
</tr>
<tr>
<td></td>
<td>AWS SDK for Python</td>
</tr>
<tr>
<td></td>
<td>AWS SDK for Ruby V3</td>
</tr>
</tbody>
</table>

When you launch your instance, you can launch your instance in a subnet that is associated with one of the following resources:

- An Availability Zone - This option is the default.
- A Local Zone - To launch an instance in a Local Zone, you must opt in to the Local Zone, and then create a subnet in the zone. For more information, see Local Zones
- A Wavelength Zone - To launch an instance in a Wavelength Zone, you must opt in to the Wavelength Zone, and then create a subnet in the zone. For information about how to launch an instance in a Wavelength Zone, see Get started with AWS Wavelength in the AWS Wavelength Developer Guide.
• An Outpost - To launch an instance in an Outpost, you must create an Outpost. For information about how to create an Outpost, see Get Started with AWS Outposts in the AWS Outposts User Guide.

After you launch your instance, you can connect to it and use it. To begin, the instance state is pending. When the instance state is running, the instance has started booting. There might be a short time before you can connect to the instance. Note that bare metal instance types might take longer to launch. For more information about bare metal instances, see Instances built on the Nitro System (p. 146).

The instance receives a public DNS name that you can use to contact the instance from the internet. The instance also receives a private DNS name that other instances within the same VPC can use to contact the instance. For more information about connecting to your instance, see Connect to your Windows instance (p. 413).

When you are finished with an instance, be sure to terminate it. For more information, see Terminate your instance (p. 441).

Launch an instance using the Launch Instance Wizard

You can launch an instance using the launch instance wizard. The launch instance wizard specifies all the launch parameters required for launching an instance. Where the launch instance wizard provides a default value, you can accept the default or specify your own value. At the very least, you need to select an AMI and a key pair to launch an instance.

Before you launch your instance, be sure that you are set up. For more information, see Set up to use Amazon EC2 (p. 5).

Important
When you launch an instance that’s not within the AWS Free Tier, you are charged for the time that the instance is running, even if it remains idle.

Steps to launch an instance:

- Initiate instance launch (p. 392)
- Step 1: Choose an Amazon Machine Image (AMI) (p. 392)
- Step 2: Choose an Instance Type (p. 393)
- Step 3: Configure Instance Details (p. 394)
- Step 4: Add Storage (p. 396)
- Step 5: Add Tags (p. 396)
- Step 6: Configure Security Group (p. 397)
- Step 7: Review Instance Launch and Select Key Pair (p. 397)

Initiate instance launch

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation bar at the top of the screen, the current Region is displayed (for example, US East (Ohio)). Select a Region for the instance that meets your needs. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can’t. For more information, see Resource locations (p. 1442).
3. From the Amazon EC2 console dashboard, choose Launch instance.

Step 1: Choose an Amazon Machine Image (AMI)

When you launch an instance, you must select a configuration, known as an Amazon Machine Image (AMI). An AMI contains the information required to create a new instance. For example, an AMI might contain the software required to act as a web server, such as Windows, Apache, and your website.
When you launch an instance, you can either select an AMI from the list, or you can select a Systems Manager parameter that points to an AMI ID. For more information, see Using a Systems Manager parameter to find an AMI.

On the **Choose an Amazon Machine Image (AMI)** page, use one of two options to choose an AMI. Either search the list of AMIs (p. 393), or search by Systems Manager parameter (p. 393).

**By searching the list of AMIs**

1. Select the type of AMI to use in the left pane:
   - **Quick Start**
     A selection of popular AMIs to help you get started quickly. To select an AMI that is eligible for the free tier, choose *Free tier only* in the left pane. These AMIs are marked *Free tier eligible*.
   - **My AMIs**
     The private AMIs that you own, or private AMIs that have been shared with you. To view AMIs that are shared with you, choose *Shared with me* in the left pane.
   - **AWS Marketplace**
     An online store where you can buy software that runs on AWS, including AMIs. For more information about launching an instance from the AWS Marketplace, see Launch an AWS Marketplace instance (p. 412).
   - **Community AMIs**
     The AMIs that AWS community members have made available for others to use. To filter the list of AMIs by operating system, choose the appropriate check box under **Operating system**. You can also filter by architecture and root device type.

2. Check the **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.

3. Check the **Boot mode** listed for each AMI. Notice which AMIs use the boot mode that you need, either *legacy-bios* or *uefi*. For more information, see Boot modes (p. 21).

4. Choose an AMI that meets your needs, and then choose **Select**.

**By Systems Manager parameter**

1. Choose **Search by Systems Manager parameter** (at top right).
2. For **Systems Manager parameter**, select a parameter. The corresponding AMI ID appears next to **Currently resolves to**.
3. Choose **Search**. The AMIs that match the AMI ID appear in the list.
4. Select the AMI from the list, and choose **Select**.

**Step 2: Choose an Instance Type**

On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. Larger instance types have more CPU and memory. For more information, see Instance types (p. 141).

To remain eligible for the free tier, choose the **t2.micro** instance type (or the **t3.micro** instance type in Regions where **t2.micro** is unavailable). For more information, see Burstable performance instances (p. 160).

By default, the wizard displays current generation instance types, and selects the first available instance type based on the AMI that you selected. To view previous generation instance types, choose All generations from the filter list.
Note
To set up an instance quickly for testing purposes, choose **Review and Launch** to accept the default configuration settings, and launch your instance. Otherwise, to configure your instance further, choose **Next: Configure Instance Details**.

**Step 3: Configure Instance Details**

On the **Configure Instance Details** page, change the following settings as necessary (expand **Advanced Details** to see all the settings), and then choose **Next: Add Storage**:

- **Number of instances**: Enter the number of instances to launch.

  **Tip**
  To ensure faster instance launches, break up large requests into smaller batches. For example, create five separate launch requests for 100 instances each instead of one launch request for 500 instances.

- (Optional) To help ensure that you maintain the correct number of instances to handle demand on your application, you can choose **Launch into Auto Scaling Group** to create a launch configuration and an Auto Scaling group. Auto Scaling scales the number of instances in the group according to your specifications. For more information, see the *Amazon EC2 Auto Scaling User Guide*.

  **Note**
  If Amazon EC2 Auto Scaling marks an instance that is in an Auto Scaling group as unhealthy, the instance is automatically scheduled for replacement where it is terminated and another is launched, and you lose your data on the original instance. An instance is marked as unhealthy if you stop or reboot the instance, or if another event marks the instance as unhealthy. For more information, see **Health Checks for Auto Scaling Instances** in the *Amazon EC2 Auto Scaling User Guide*.

- **Purchasing option**: Choose **Request Spot instances** to launch a Spot Instance. This adds and removes options from this page. Set your maximum price, and optionally update the request type, interruption behavior, and request validity. For more information, see Create a Spot Instance request (p. 295).

- **Network**: Select the VPC, or to create a new VPC, choose **Create new VPC** to go to the Amazon VPC console. When you have finished, return to the wizard and choose **Refresh** to load your VPC in the list.

- **Subnet**: You can launch an instance in a subnet associated with an Availability Zone, Local Zone, Wavelength Zone or Outpost.

  To launch the instance in an Availability Zone, select the subnet into which to launch your instance. You can select **No preference** to let AWS choose a default subnet in any Availability Zone. To create a new subnet, choose **Create new subnet** to go to the Amazon VPC console. When you are done, return to the wizard and choose **Refresh** to load your subnet in the list.

  To launch the instance in a Local Zone, select a subnet that you created in the Local Zone.

  To launch an instance in an Outpost, select a subnet in a VPC that you associated with an Outpost.

- **Auto-assigned 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. 894).

- **Auto-assigned 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 after launch. If you select a domain, you must select an IAM role with the required permissions. For more information, see Seamlessly Join a Windows EC2 Instance.
• **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. 975).

• **Capacity Reservation**: Specify whether to launch the instance into shared capacity, any open Capacity Reservation, a specific Capacity Reservation, or a Capacity Reservation group. For more information, see Launch instances into an existing Capacity Reservation (p. 374).

• **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. 1114).

• **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 Optimize CPU options (p. 540).

• **Shutdown behavior**: Select whether the instance should stop or terminate when shut down. For more information, see Change the instance initiated shutdown behavior (p. 445).

• **Stop - Hibernate behavior**: To enable hibernation, select this check box. This option is only available if your instance meets the hibernation prerequisites. For more information, see Hibernate your On-Demand or Reserved Windows instance (p. 428).

• **Enable termination protection**: To prevent accidental termination, select this check box. For more information, see Enable termination protection (p. 443).

• **Monitoring**: Select this check box to enable detailed monitoring of your instance using Amazon CloudWatch. Additional charges apply. For more information, see Monitor your instances using CloudWatch (p. 839).

• **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. 1344).

• **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. 360) and Dedicated Hosts (p. 329).

• **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. 160).

• **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.
  - **Network Card Index**: The index of the network card. The primary network interface must be assigned to network card index 0. Some instance types support multiple network cards.
  - 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. 934). 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 Assign a public IPv4 address during instance launch (p. 898).

• **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.
- **Enclave**: Select **Enable** to enable the instance for AWS Nitro Enclaves. For more information, see What is AWS Nitro Enclaves? in the AWS Nitro Enclaves User Guide.
- **Metadata accessible**: You can enable or disable access to the instance metadata. For more information, see Use IMDSv2 (p. 580).
- **Metadata transport**: You can enable or disable the access method to the instance metadata service that's available for this EC2 instance based on the IP address type (IPv4, IPv6, or IPv4 and IPv6) of the instance. For more information, see Retrieve instance metadata (p. 586).
- **Metadata version**: If you enable access to the instance metadata, you can choose to require the use of Instance Metadata Service Version 2 when requesting instance metadata. For more information, see Configure instance metadata options for new instances (p. 584).
- **Metadata token response hop limit**: If you enable instance metadata, you can set the allowable number of network hops for the metadata token. For more information, see Use IMDSv2 (p. 580).
- **User data**: You can specify user data to configure an instance during launch, or to run a configuration script. To attach a file, select the **As file** option and browse for the file to attach.

**Step 4: Add Storage**

The AMI you selected includes one or more volumes of storage, including the root device volume. On the **Add Storage** page, you can specify additional volumes to attach to the instance by choosing **Add New Volume**. Configure each volume as follows, and then choose **Next: Add Tags**.

- **Type**: Select instance store or Amazon EBS volumes to associate with your instance. The types of volume available in the list depend on the instance type you've chosen. For more information, see Amazon EC2 instance store (p. 1392) and Amazon EBS volumes (p. 1160).
- **Device**: Select from the list of available device names for the volume.
- **Snapshot**: Enter the name or ID of the snapshot from which to restore a volume. You can also search for available shared and public snapshots by typing text into the **Snapshot** field. Snapshot descriptions are case-sensitive.
- **Size**: For EBS volumes, you can specify a storage size. Even if you have selected an AMI and instance that are eligible for the free tier, to stay within the free tier, you must stay under 30 GiB of total storage. For more information, see Constraints on the size and configuration of an EBS volume (p. 1181).
- **Volume Type**: For EBS volumes, select a volume type. For more information, see Amazon EBS volume types (p. 1163).
- **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 Preserve Amazon EBS volumes on instance termination (p. 445).
- **Encrypted**: If the instance type supports EBS encryption, you can specify the encryption state of the volume. If you have enabled encryption by default in this Region, the default customer managed key is selected for you. You can select a different key or disable encryption. For more information, see Amazon EBS encryption (p. 1327).

**Step 5: Add Tags**

On the **Add Tags** page, specify tags (p. 1450) by providing key and value combinations. You can tag the instance, the volumes, or both. For Spot Instances, you can tag the Spot Instance request only. Choose **Add another tag** to add more than one tag to your resources. Choose **Next: Configure Security Group** when you are done.
Step 6: Configure Security Group

On the Configure Security Group page, use a security group to define firewall rules for your instance. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored. (For more information about security groups, see Amazon EC2 security groups for Windows instances (p. 1135).) Select or create a security group as follows, and then choose Review and Launch.

- To select an existing security group, choose Select an existing security group, and select your security group. You can't edit the rules of an existing security group, but you can copy them to a new group by choosing Copy to new. Then you can add rules as described in the next step.
- To create a new security group, choose Create a new security group. The wizard automatically defines the launch-wizard-x security group and creates an inbound rule to allow you to connect to your instance over RDP (port 3389).
- You can add rules to suit your needs. For example, if your instance is a web server, open ports 80 (HTTP) and 443 (HTTPS) to allow internet traffic.

To add a rule, choose Add Rule, select the protocol to open to network traffic, and then specify the source. Choose My IP from the Source list to let the wizard add your computer's public IP address. However, if you are connecting through an ISP or from behind your firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

**Warning**

Rules that enable all IP addresses (0.0.0.0/0) to access your instance over SSH or RDP are acceptable for this short exercise, but are unsafe for production environments. You should authorize only a specific IP address or range of addresses to access your instance.

Step 7: Review Instance Launch and Select Key Pair

On the Review Instance Launch page, check the details of your instance, and make any necessary changes by choosing the appropriate Edit link.

When you are ready, choose Launch.

In the Select an existing key pair or create a new key pair dialog box, you can choose an existing key pair, or create a new one. For example, choose Choose an existing key pair, then select the key pair you created when getting set up. For more information, see Amazon EC2 key pairs and Windows instances (p. 1127).

**Important**

If you choose the Proceed without key pair option, you won't be able to connect to the instance unless you choose an AMI that is configured to allow users another way to log in.

To launch your instance, select the acknowledgment check box, then choose Launch Instances.

(Optional) You can create a status check alarm for the instance (additional fees may apply). (If you're not sure, you can always add one later.) On the confirmation screen, choose Create status check alarms and follow the directions. For more information, see Create and edit status check alarms (p. 813).

If the instance fails to launch or the state immediately goes to terminated instead of running, see Troubleshoot instance launch issues (p. 1465).

Launch an instance from a launch template

You can create a launch template that contains the configuration information to launch an instance. You can use launch templates 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. 398)
- Use launch templates to control launch parameters (p. 399)
- Control the use of launch templates (p. 399)
- Create a launch template (p. 399)
- Modify a launch template (manage launch template versions) (p. 405)
- Launch an instance from a launch template (p. 408)
- Use launch templates with Amazon EC2 Auto Scaling (p. 409)
- Use launch templates with EC2 Fleet (p. 410)
- Use launch templates with Spot Fleet (p. 410)
- Delete a launch template (p. 410)

Launch template restrictions

The following rules apply to launch templates and launch template versions:

- You are limited to creating 5,000 launch templates per Region and 10,000 versions per launch template.
- Launch template parameters are optional. However, you must ensure that your request to launch an instance includes all required parameters. For example, if your launch template does not include an AMI ID, you must specify both the launch template and an AMI ID when you launch an instance.
- Launch template parameters are not fully validated when you create the launch template. If you specify incorrect values for parameters, or if you do not use supported parameter combinations, no instances can launch using this launch template. Ensure that you specify the correct values for the parameters and that you use supported parameter combinations. For example, to launch an instance in a placement group, you must specify a supported instance type.
- You can tag a launch template, but you cannot tag a launch template version.
• Launch templates are immutable. To modify a launch template, you must create a new version of the launch template.
• 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.

Use 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. IAM users 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. 1090).

Control the use of launch templates

By default, IAM users do not have permissions to work with launch templates. You can create an IAM user policy that grants users permissions to create, modify, describe, and delete launch templates and launch template versions. You can also apply resource-level permissions to some launch template actions to control a user's ability to use specific resources for those actions. For more information, see the following example policies: Example: Work with launch templates (p. 1101).

Take care when granting users permissions to use the `ec2:CreateLaunchTemplate` and `ec2:CreateLaunchTemplateVersion` actions. You cannot use resource-level permissions to control which resources users can specify in the launch template. To restrict the resources that are used to launch an instance, ensure that you grant permissions to create launch templates and launch template versions only to appropriate administrators.

Create a launch template

Create a new launch template using parameters that you define, or use an existing launch template or an instance as the basis for a new launch template.

Tasks
• Create a new launch template using parameters you define (p. 399)
• Create a launch template from an existing launch template (p. 403)
• Create a launch template from an instance (p. 404)

Create a new launch template using parameters you define

Console

To create a new launch template using defined parameters using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Launch Templates**, and then choose **Create launch template**.
3. For **Launch template name**, enter a descriptive name for the launch template.
4. For **Template version description**, provide a brief description of the launch template version.
5. To tag the launch template on creation, expand **Template tags**, choose **Add tag**, and then enter a tag key and value pair.
6. For **Launch template contents**, provide the following information:
   - **AMI**: An AMI from which to launch the instance. To search through all available AMIs, choose **Search for AMI**. To select a commonly used AMI, choose **Quick Start**. Or, choose **AWS Marketplace** or **Community AMIs**. You can use an AMI that you own or **find a suitable AMI**.
   - **Instance type**: Ensure that the instance type is compatible with the AMI that you've specified. For more information, see **Instance types** (p. 141).
   - **Key pair name**: The key pair for the instance. For more information, see **Amazon EC2 key pairs and Windows instances** (p. 1127).
   - **Network platform**: If applicable, whether to launch the instance into a VPC or EC2-Classic. If you choose **VPC**, specify the subnet in the **Network interfaces** section. If you choose **Classic**, ensure that the specified instance type is supported in EC2-Classic and specify the Availability Zone for the instance.
   - **Security groups**: One or more security groups to associate with the instance. If you add a network interface to the launch template, omit this setting and specify the security groups as part of the network interface specification. You cannot launch an instance from a launch template that specifies security groups and a network interface. For more information, see **Amazon EC2 security groups for Windows instances** (p. 1135).
7. For **Storage (volumes)**, specify volumes to attach to the instance besides the volumes specified by the AMI (**Volume 1 (AMI Root)**). To add a new volume, choose **Add new volume**.
   - **Volume type**: The instance store or Amazon EBS volumes with which to associate your instance. The type of volume depends on the instance type that you’ve chosen. For more information, see **Amazon EC2 instance store** (p. 1392) and **Amazon EBS volumes** (p. 1160).
   - **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. 1163).
   - **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 **Preserve Amazon EBS volumes on instance termination** (p. 445).
   - **Encrypted**: If the instance type supports EBS encryption, you can enable encryption for the volume. If you have enabled encryption by default in this Region, encryption is enabled for you. For more information, see **Amazon EBS encryption** (p. 1527).
   - **Key**: The customer managed key to use for EBS encryption. You can specify the ARN of any customer managed key that you created using the customer managed key. If you specify a customer managed key, you must also use **Encrypted** to enable encryption.
8. For **Resource tags**, specify **tags** (p. 1450) by providing key and value combinations. You can tag the instance, the volumes, Spot Instance requests, or all three.
9. For **Network interfaces**, you can specify up to two **network interfaces** (p. 934) for the instance.
   - **Device index**: The device number for the network interface, for example, *eth0* for the primary network interface. If you leave the field blank, AWS creates the primary network interface.
• **Network interface**: The ID of the network interface, or leave blank to let AWS create a new network interface.

• **Description**: (Optional) A description for the new network interface.

• **Subnet**: The subnet in which to create a new network interface. For the primary network interface (eth0), this is the subnet in which the instance is launched. If you've entered an existing network interface for eth0, the instance is launched in the subnet in which the network interface is located.

• **Auto-assign public IP**: Whether to automatically assign a public IP address to the network interface with the device index of eth0. This setting can only be enabled for a single, new network interface.

• **Primary IP**: A private IPv4 address from the range of your subnet. Leave blank to let AWS choose a private IPv4 address for you.

• **Secondary IP**: A secondary private IPv4 address from the range of your subnet. Leave blank to let AWS choose one for you.

• (IPv6-only) **IPv6 IPs**: An IPv6 address from the range of the subnet.

• **Security groups**: One or more security groups in your VPC with which to associate the network interface.

• **Delete on termination**: Whether the network interface is deleted when the instance is deleted.

• **Network card index**: The index of the network card. The primary network interface must be assigned to network card index 0. Some instance types support multiple network cards.

10. For **Advanced details**, expand the section to view the fields and specify any additional parameters for the instance.

• **Purchasing option**: The purchasing model. Choose Request Spot Instances to request Spot Instances at the Spot price, capped at the On-Demand price, and choose Customize to change the default Spot Instance settings. If you do not request a Spot Instance, EC2 launches an On-Demand Instance by default. For more information, see Spot Instances (p. 282).

• **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. 1114).

• **Shutdown behavior**: Whether the instance should stop or terminate when shut down. For more information, see Change the instance initiated shutdown behavior (p. 445).

• **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 On-Demand or Reserved Windows instance (p. 428).

• **Termination protection**: Whether to prevent accidental termination. For more information, see Enable termination protection (p. 443).

• **Detailed CloudWatch monitoring**: Whether to enable detailed monitoring of the instance using Amazon CloudWatch. Additional charges apply. For more information, see Monitor your instances using CloudWatch (p. 839).

• **Elastic GPU**: An Elastic Graphics accelerator to attach to the instance. Not all instance types support Elastic Graphics. For more information, see Amazon Elastic Graphics (p. 793).

• **Elastic inference**: An elastic inference accelerator to attach to your EC2 CPU instance. For more information, see Working with Amazon Elastic Inference in the Amazon Elastic Inference Developer Guide.

• **T2/T3 Unlimited**: Whether to enable applications to burst beyond the baseline for as long as needed. This field is only valid for T2, T3, and T3a instances. Additional charges may apply. For more information, see Burstable performance instances (p. 160).

• **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. 975).
• **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. 1344).

• **Capacity Reservation**: Specify whether to launch the instance into any open Capacity Reservation (Open), a specific Capacity Reservation (Target by ID), or a Capacity Reservation group (Target by group). To specify that a Capacity Reservation should not be used, choose None. For more information, see [Launch instances into an existing Capacity Reservation](p. 374).

• **Tenancy**: Choose whether to run your instance on shared hardware (Shared), isolated, dedicated hardware (Dedicated), or on a Dedicated Host (Dedicated host). If you choose to launch the instance on a Dedicated Host, you can specify whether to launch the instance into a host resource group or you can target a specific Dedicated Host. Additional charges may apply. For more information, see [Dedicated Instances](p. 360) and [Dedicated Hosts](p. 329).

• **RAM disk ID**: (Only valid for paravirtual (PV) AMIs) A RAM disk for the instance. If you have specified a kernel, you may need to specify a specific RAM disk with the drivers to support it.

• **Kernel ID**: (Only valid for paravirtual (PV) AMIs) A kernel for the instance.

• **License configurations**: You can launch instances against the specified license configuration to track your license usage. For more information, see [Create a License Configuration](in the AWS License Manager User Guide).

• **Metadata accessible**: Whether to enable or disable access to the instance metadata. For more information, see [Use IMDSv2](p. 580).

• **Metadata version**: If you enable access to the instance metadata, you can choose to require the use of Instance Metadata Service Version 2 when requesting instance metadata. For more information, see [Configure instance metadata options for new instances](p. 584).

• **Metadata response hop limit**: If you enable instance metadata, you can set the allowable number of network hops for the metadata token. For more information, see [Use IMDSv2](p. 580).

• **User data**: You can specify user data to configure an instance during launch, or to run a configuration script. For more information, see [Run commands on your Windows instance at launch](p. 572).

11. Choose **Create launch template**.

**AWS CLI**

**To create a launch template using the AWS CLI**

- Use the `create-launch-template` command. The following example creates a launch template that specifies the following:

  - A tag for the launch template (`purpose=production`)
  - The instance type (`r4.4xlarge`) and AMI (`ami-8c1be5f6`) to launch
  - The number of cores (4) and threads per core (2) for a total of 8 vCPUs (4 cores x 2 threads)
  - The subnet in which to launch the instance (`subnet-7b16de0c`)

The template assigns a public IP address and an IPv6 address to the instance and creates a tag for the instance (`Name=webserver`).

```bash
aws ec2 create-launch-template \
  --launch-template-name TemplateForWebServer \
  --version-description WebVersion1 \
  --tag-specifications 'ResourceType=launch-template,Tags=[(Key=purpose,Value=production)]'
```

402
The following is an example template-data.json file.

```json
{
    "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.

```json
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-01238c059e3466abc",
        "LaunchTemplateName": "TemplateForWebServer",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:root",
        "CreateTime": "2017-11-27T09:13:24.000Z"
    }
}
```

Create a launch template from an existing launch template

To create a launch template from an existing launch template using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Launch Templates, and then choose Create launch template.
3. For Launch template name, enter a descriptive name for the launch template.
4. For Template version description, provide a brief description of the launch template version.
5. To tag the launch template on creation, expand Template tags, choose Add tag, and then enter a tag key and value pair.
6. Expand Source template, and for Launch template name choose a launch template on which to base the new launch template.
7. For Source template version, choose the launch template version on which to base the new launch template.
8. Adjust any launch parameters as required, and then choose Create launch template.
Create a launch template from an instance

Console

To create a launch template from an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance, and choose Actions, Create template from instance.
4. Provide a name, description, and tags, and adjust the launch parameters as required.

   Note
   When you create a launch template from an instance, the instance's network interface IDs and IP addresses are not included in the template.

5. Choose Create launch template.

AWS CLI

You can use the AWS CLI to create a launch template from an existing instance by first getting the launch template data from an instance, and then creating a launch template using the launch template data.

To get launch template data from an instance using the AWS CLI

- Use the get-launch-template-data command and specify the instance ID. You can use the output as a base to create a new launch template or launch template version. By default, the output includes a top-level LaunchTemplateData object, which cannot be specified in your launch template data. Use the --query option to exclude this object.

```bash
aws ec2 get-launch-template-data \
   --instance-id i-0123d646e8048babc \
   --query "LaunchTemplateData"
```

The following is example output.

```json
{
   "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",
         "PrivateIpAddress": ["" ]
      }
   ]
}
```

404
"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:

```bash
aws ec2 get-launch-template-data \
  --instance-id i-0123d646e8048babc \
  --query "LaunchTemplateData" >> instance-data.json
```

To create a launch template using launch template data

Use the `create-launch-template` command to create a launch template using the output from the previous procedure. For more information about creating a launch template using the AWS CLI, see Create a new launch template using parameters you define (p. 399).

Modify a launch template (manage launch template versions)

Launch templates are immutable; after you create a launch template, you can't modify it. Instead, you can create a new version of the launch template that includes any changes you require.

You can create launch template versions for a specific launch template, set the default version, describe a launch template version, and delete versions that you no longer require.

**Tasks**

- Create a launch template version (p. 405)
- Set the default launch template version (p. 406)
- Describe a launch template version (p. 406)
- Delete a launch template version (p. 407)

Create 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 Create a launch template (p. 399).

**Console**

**To create a launch template version using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Launch Templates**.
3. Select a launch template, and then choose **Actions, Modify template (Create new version)**.
4. For **Template version description**, enter a description for the launch template version.
5. (Optional) Expand **Source template** and select a version of the launch template to use as a base for the new launch template version. The new launch template version inherits the launch parameters from this launch template version.
6. Modify the launch parameters as required, and choose **Create launch template**.

**AWS CLI**

**To create a launch template version using the AWS CLI**

- Use the `create-launch-template-version` command. You can specify a source version on which to base the new version. The new version inherits the launch parameters from this version, and you can override parameters using `--launch-template-data`. The following example creates a new version based on version 1 of the launch template and specifies a different AMI ID.

```
aws ec2 create-launch-template-version \
  --launch-template-id lt-0abcd290751193123 \
  --version-description WebVersion2 \
  --source-version 1 \
  --launch-template-data "ImageId=ami-c998b6b2"
```

**Set 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.

**Console**

**To set the default launch template version using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Launch Templates**.
3. Select the launch template and choose **Actions, Set default version**.
4. For **Template version**, select the version number to set as the default version and choose **Set as default version**.

**AWS CLI**

**To set the default launch template version using the AWS CLI**

- Use the `modify-launch-template` command and specify the version that you want to set as the default.

```
aws ec2 modify-launch-template \
  --launch-template-id lt-0abcd290751193123 \
  --default-version 2
```

**Describe a launch template version**

Using the console, you can view all the versions of the selected launch template, or get a list of the launch templates whose latest or default version matches a specific version number. Using the AWS CLI,
you can describe all versions, individual versions, or a range of versions of a specified launch template. You can also describe all the latest versions or all the default versions of all the launch templates in your account.

Console

To describe a launch template version using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Launch Templates.
3. You can view a version of a specific launch template, or get a list of the launch templates whose latest or default version matches a specific version number.
   - To view a version of a launch template: Select the launch template. On the Versions tab, from Version, select a version to view its details.
   - To get a list of all the launch templates whose latest version matches a specific version number: From the search bar, choose Latest version, and then choose a version number.
   - To get a list of all the launch templates whose default version matches a specific version number: From the search bar, choose Default version, and then choose a version number.

AWS CLI

To describe a launch template version using the AWS CLI

- Use the `describe-launch-template-versions` command and specify the version numbers. In the following example, versions 1 and 3 are specified.

```bash
aws ec2 describe-launch-template-versions \
  --launch-template-id lt-0abcd290751193123 \
  --versions 1 3
```

To describe all the latest and default launch template versions in your account using the AWS CLI

- Use the `describe-launch-template-versions` command and specify $Latest, $Default, or both. You must omit the launch template ID and name in the call. You cannot specify version numbers.

```bash
aws ec2 describe-launch-template-versions \
  --versions "$Latest,$Default"
```

Delete 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.

Console

To delete a launch template version using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Launch Templates.
3. Select the launch template and choose Actions, Delete template version.
Launch 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.

Console

To launch an instance from a launch template using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Launch Templates.
3. Select the launch template and choose Actions, Launch instance from template.
4. For Source template version, select the launch template version to use.
5. For Number of instances, specify the number of instances to launch.
6. (Optional) You can override or add launch template parameters by changing and adding parameters in the Instance details section.
7. Choose Launch instance from template.

AWS CLI

To launch an instance from a launch template using the AWS CLI

- Use the `run-instances` command and specify the --launch-template parameter. Optionally specify the launch template version to use. If you don't specify the version, the default version is used.

```
aws ec2 run-instances \
  --launch-template LaunchTemplateId=lt-0abcd290751193123,Version=1
```

- To override a launch template parameter, specify the parameter in the `run-instances` command. The following example overrides the instance type that's specified in the launch template (if any).

```
aws ec2 run-instances \
  --launch-template LaunchTemplateId=lt-0abcd290751193123 \
  --instance-type t2.small
```

- If you specify a nested parameter that's part of a complex structure, the instance is launched using the complex structure as specified in the launch template plus any additional nested parameters that you specify.
In the following example, the instance is launched with the tag \textit{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 \textit{Owner}, the value is replaced with TeamA.

```
aws ec2 run-instances \\
  --launch-template LaunchTemplateId=lt-0abcd290751193123 \\
  --tag-specifications "ResourceType=instance,Tags=[{Key=Owner,Value=TeamA}]
```

In the following example, the instance is launched with a volume with the device name /dev/xvdb as well as any other block device mappings that are specified in the launch template. If the launch template has an existing volume defined for /dev/xvdb, its values are replaced with the specified values.

```
aws ec2 run-instances \\
  --launch-template LaunchTemplateId=lt-0abcd290751193123 \\
  --block-device-mappings "DeviceName=/dev/xvdb,Ebs={VolumeSize=20,VolumeType=gp2}"
```

If the instance fails to launch or the state immediately goes to terminated instead of running, see Troubleshoot instance launch issues (p. 1465).

Use launch templates with Amazon EC2 Auto Scaling

You can create an Auto Scaling group and specify a launch template to use for the group. When Amazon EC2 Auto Scaling launches instances in the Auto Scaling group, it uses the launch parameters defined in the associated launch template. For more information, see Creating an Auto Scaling Group Using a Launch Template in the Amazon EC2 Auto Scaling User Guide.

Before you can create an Auto Scaling group using a launch template, you must create a launch template that includes the parameters required to launch an instance in an Auto Scaling group, such as the ID of the AMI. The console provides guidance to help you create a template that you can use with Auto Scaling.

To create a launch template to use with Auto Scaling using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Launch Templates, and then choose Create launch template.
3. For Launch template name, enter a descriptive name for the launch template.
4. For Template version description, provide a brief description of the launch template version.
5. Under Auto Scaling guidance, select the checkbox to have Amazon EC2 provide guidance to help create a template to use with Auto Scaling.
6. Modify the launch parameters as required. Because you selected Auto Scaling guidance, some fields are required and some fields are not available. For considerations to keep in mind when creating a launch template, and for information about how to configure the launch parameters for Auto Scaling, see Creating a Launch Template for an Auto Scaling Group in the Amazon EC2 Auto Scaling User Guide.
7. Choose Create launch template.
8. (Optional) To create an Auto Scaling group using this launch template, in the Next steps page, choose Create Auto Scaling group.
To create or update an Amazon EC2 Auto Scaling group with a launch template using the AWS CLI

- Use the create-auto-scaling-group or the update-auto-scaling-group command and specify the --launch-template parameter.

Use 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 Create an EC2 Fleet (p. 701).

To create an EC2 Fleet with a launch template using the AWS CLI

- Use the create-fleet command. Use the --launch-template-configs parameter to specify the launch template and any overrides for the launch template.

Use 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. 718).

To create a Spot Fleet request with a launch template using the AWS CLI

- Use the request-spot-fleet command. Use the LaunchTemplateConfigs parameter to specify the launch template and any overrides for the launch template.

Delete a launch template

If you no longer require a launch template, you can delete it. Deleting a launch template deletes all of its versions.

Console

To delete a launch template (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Launch Templates.
3. Select the launch template and choose Actions, Delete template.
4. Enter Delete to confirm deletion, and then choose Delete.

AWS CLI

To delete a launch template (AWS CLI)

- Use the delete-launch-template (AWS CLI) command and specify the launch template.

```
aws ec2 delete-launch-template --launch-template-id lt-01238c059e3466abc
```
Launch 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. 397) 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. 894)**.
- 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 ** Seamlessly Join a Windows EC2 Instance** in the **AWS Directory Service Administration Guide**.

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.

**New console**

**To use your current instance as a template**

1. Open the Amazon EC2 console at **https://console.aws.amazon.com/ec2/**.
2. In the navigation pane, choose **Instances**.
3. Select the instance you want to use, and then choose Actions, Images and templates, Launch more like this.
4. The launch wizard opens on the Review Instance Launch page. You can make any necessary changes by choosing the appropriate Edit link.

   When you are ready, choose Launch to select a key pair and launch your instance.
5. If the instance fails to launch or the state immediately goes to terminated instead of running, see Troubleshoot instance launch issues (p. 1465).

Old console

To use your current instance as a template

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance you want to use, and then choose Actions, Launch More Like This.
4. The launch wizard opens on the Review Instance Launch page. You can make any necessary changes by choosing the appropriate Edit link.

   When you are ready, choose Launch to select a key pair and launch your instance.
5. If the instance fails to launch or the state immediately goes to terminated instead of running, see Troubleshoot instance launch issues (p. 1465).

Launch 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. 109). To cancel your subscription after launch, you first have to terminate all instances running from it. For more information, see Manage your AWS Marketplace subscriptions (p. 112).

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 Launch an instance using the Launch Instance Wizard (p. 392). Choose Next until you reach the Configure Security Group page.

   The wizard creates a new security group according to the vendor's specifications for the product. The security group may include rules that allow all IPv4 addresses (0.0.0.0/0) access on SSH (port 22)
on Linux or RDP (port 3389) on Windows. We recommend that you adjust these rules to allow only a specific address or range of addresses to access your instance over those ports.

When you are ready, choose Review and Launch.

7. On the Review Instance Launch page, check the details of the AMI from which you’re about to launch the instance, as well as the other configuration details you set up in the wizard. When you’re ready, choose Launch to select or create a key pair, and launch your instance.

8. Depending on the product you’ve subscribed to, the instance may take a few minutes or more to launch. You are first subscribed to the product before your instance can launch. If there are any problems with your credit card details, you will be asked to update your account details. When the launch confirmation page displays, choose View Instances to go to the Instances page.

   **Note**
   You are charged the subscription price as long as your instance is running, even if it is idle. If your instance is stopped, you may still be charged for storage.

9. When your instance is in the running state, you can connect to it. To do this, select your instance in the list and choose Connect. Follow the instructions in the dialog. For more information about connecting to your instance, see Connect to your Windows instance (p. 413).

   **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 Manage your AWS Marketplace subscriptions (p. 112).

10. If the instance fails to launch or the state immediately goes to terminated instead of running, see Troubleshoot instance launch issues (p. 1465).

Launch 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:

<table>
<thead>
<tr>
<th>Method</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS CLI</td>
<td>Use the run-instances command, or see the following topic for more information: Launching an Instance.</td>
</tr>
<tr>
<td>AWS Tools for Windows PowerShell</td>
<td>Use the New-EC2Instance command, or see the following topic for more information: Launch an Amazon EC2 Instance Using Windows PowerShell</td>
</tr>
<tr>
<td>Query API</td>
<td>Use the RunInstances request.</td>
</tr>
</tbody>
</table>

Connect to your Windows instance

You can connect to Amazon EC2 instances created from most Windows Amazon Machine Images (AMIs) using Remote Desktop. Remote Desktop uses the Remote Desktop Protocol (RDP) to connect to and use your instance in the same way you use a computer sitting in front of you (local computer). It is available on most editions of Windows and is also available for Mac OS.

The license for the Windows Server operating system allows two simultaneous remote connections for administrative purposes. The license for Windows Server is included in the price of your Windows instance. If you require 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 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. 414)
- Connect to your Windows instance using RDP (p. 414)
- Connect to a Windows instance using its IPv6 address (p. 420)
- Connect to a Windows instance using Session Manager (p. 422)
- Configure your accounts (p. 422)
- Transfer files to Windows instances (p. 423)

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 Remmina.
- Locate the private key
  Get the fully-qualified path to the location on your computer of the .pem file for the key pair that you specified when you launched the instance. For more information, see Identify the key pair that was specified at launch. If you can't find your private key file, see Connect to your Windows instance if you lose your private key.
- Enable inbound RDP traffic from your IP address to your instance
  Ensure that the security group associated with your instance allows incoming RDP traffic (port 3389) from your IP address. The default security group does not allow incoming RDP traffic by default. For more information, see Authorize inbound traffic for your Windows instances (p. 1123).

Connect to your Windows instance using RDP

To connect to a Windows instance, you must retrieve the initial administrator password and then enter this password when you connect to your instance using Remote Desktop. It takes a few minutes after instance launch before this password is available.

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.

If you receive an error while attempting to connect to your instance, see Remote Desktop can't connect to the remote computer (p. 1469).
New console

To connect to your Windows instance using an RDP client

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, select Instances. Select the instance and then choose Connect.
3. On the Connect to instance page, choose the RDP client tab, and then choose Get password.

4. Choose Browse and navigate to the private key (\.pem) file you created when you launched the instance. Select the file and choose Open to copy the entire contents of the file to this window.

5. Choose Decrypt Password. The console displays the default administrator password for the instance under Password, replacing the Get password link shown previously. Save the password in a safe place. This password is required to connect to the instance.
6. Choose **Download remote desktop file**. Your browser prompts you to either open or save the RDP shortcut file. When you have finished downloading the file, choose **Cancel** to return to the **Instances** page.

   - 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. Choose **Connect** to continue to connect to your instance.
8. The administrator account is chosen by default. Copy and paste the password that you saved previously.

   **Tip**
   If you receive a "Password Failed" error, try entering the password manually. Copying and pasting content can corrupt it.

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** (Windows) or **Continue** (Mac OS X) if you trust the certificate.
a. If you are using Remote Desktop Connection on a Windows computer, 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 Thumbprint (Windows) or SHA1 Fingerprints (Mac OS X). This is the unique identifier for the remote computer's security certificate.

c. In the Amazon EC2 console, select the instance, choose Actions, Monitor and troubleshoot, Get system log.

d. In the system log output, look for RDP-CERTIFICATE-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 on a Windows computer, 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 X] 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, see Add spaces and switch between them.
Old console

**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 (.pem) 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
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 (p. 900), you can use an RDP client to connect to your instance using its IPv6 address (for example, 2001:db8:1234:1a00:9691:9503:25ad:1761) instead of using its public IPv4 address or public DNS hostname.

To connect to your Windows instance using its IPv6 address

1. Get the initial administrator password for your instance, as described in Connect to your Windows instance using RDP (p. 414). This password is required to connect to your instance.
2. [Windows] Open the RDP client on your Windows computer, choose Show Options, and do the following:
• For **Computer**, enter the IPv6 address of your Windows instance.
• For **User name**, enter **Administrator**.
• Choose **Connect**.
• When prompted, enter the password that you saved previously.

[Mac OS X] Open the RDP client on your computer and do the following:
• Choose **New**.
• For **PC Name**, enter the IPv6 address of your Windows instance.
• For **User name**, enter **Administrator**.
• Close the dialog box. Under **My Desktops**, select the connection, and choose **Start**.
• When prompted, enter the password that you saved previously.

3. Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. If you trust the certificate, you can choose **Yes** or **Continue**. Otherwise,
you can verify the identity of the remote computer, as described in Connect to your Windows instance using RDP (p. 414).

**Connect to a Windows instance using Session Manager**

Session Manager is a fully-managed AWS Systems Manager capability for managing your Amazon EC2 instances through an interactive, one-click, browser-based shell, or through the AWS CLI. You can use Session Manager to start a session with an instance in your account. After the session is started, you can run PowerShell commands as you would for any other connection type. For more information about Session Manager, see AWS Systems Manager Session Manager in the AWS Systems Manager User Guide.

Before attempting to connect to an instance using Session Manager, ensure that the necessary setup steps have been completed. For more information, see Getting Started with Session Manager.

**To connect to a Windows instance using Session Manager on the Amazon EC2 console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Connect**.
4. For **Connection method**, choose **Session Manager**.
5. Choose **Connect**.

![Connect to instance](image)

**Tip**

If you receive an error that you're not authorized to perform one or more Systems Manager actions (ssm:command-name), then you must update your policies to allow you to start sessions from the Amazon EC2 console. For more information and instructions, see Quickstart Default IAM Policies for Session Manager in the AWS Systems Manager User Guide.

**Configure your accounts**

After you connect, we recommend that you perform the following:
• Change the administrator password from the default value. You can change the password while you are logged on to the instance itself, just as you would on any computer running Windows Server.

• Create another user account with administrator privileges on the instance. This is a safeguard in case you forget the administrator password or have a problem with the administrator account. The new user account must have permission to access the instance remotely. Open System Properties by right-clicking on the This PC icon on your Windows desktop or File Explorer and selecting Properties. Choose Remote settings, and choose Select Users to add the user to the Remote Desktop Users group.

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.

To map the remote session drive to your local drive

1. Open the Remote Desktop Connection client.
2. Choose Show Options.
3. Choose the Local Resources tab.
4. Under **Local Devices and resources**, 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 start your instance if it has an Amazon EBS volume as its root device. The instance retains its instance ID, but can change as described in the Overview (p. 426) section.

When you stop an instance, we shut it down. We don't charge usage for a stopped instance, or data transfer fees, but we do charge for the storage for any Amazon EBS volumes. Each time you start a stopped instance we charge a minimum of one minute for usage. After one minute, we charge only for the seconds you use. For example, if you run an instance for 20 seconds and then stop it, we charge for a full one minute. If you run an instance for 3 minutes and 40 seconds, we charge for exactly 3 minutes and 40 seconds of usage.

While the instance is stopped, you can treat its root volume like any other volume, and modify it (for example, repair file system problems or update software). You just detach the volume from the stopped instance, attach it to a running instance, make your changes, detach it from the running instance, and then reattach it to the stopped instance. Make sure that you reattach it using the storage device name that's specified as the root device in the block device mapping for the instance.

If you decide that you no longer need an instance, you can terminate it. As soon as the state of an instance changes to shutting-down or terminated, we stop charging for that instance. For more information, see Terminate your instance (p. 441). If you'd rather hibernate the instance, see Hibernate your On-Demand or Reserved Windows instance (p. 428). For more information, see Differences between reboot, stop, hibernate, and terminate (p. 389).

Contents

- Overview (p. 426)
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 (though in some cases, it remains on the current host).
- The instance retains its private IPv4 addresses and any IPv6 addresses when stopped and started. We release the public IPv4 address and assign a new one when you start it.
- The instance retains its associated Elastic IP addresses. You're charged for any Elastic IP addresses associated with a stopped instance. With EC2-Classic, an Elastic IP address is dissociated from your instance when you stop it. For more information, see EC2-Classic (p. 1022).
- 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 Configure a Windows instance using the EC2Config service (p. 493).
- If your instance is in an Auto Scaling group, the Amazon EC2 Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. For more information, see Health Checks for Auto Scaling Instances in the Amazon EC2 Auto Scaling User Guide.
- When you stop a ClassicLink instance, it's unlinked from the VPC to which it was linked. You must link the instance to the VPC again after starting it. For more information about ClassicLink, see ClassicLink (p. 1030).

For more information, see Differences between reboot, stop, hibernate, and terminate (p. 389).

You can modify the following attributes of an instance only when it is stopped:

- Instance type
- User data
- Kernel
- RAM disk

If you try to modify these attributes while the instance is running, Amazon EC2 returns the IncorrectInstanceState error.

What happens when you stop an instance

When an EC2 instance is stopped using the stop-instances command, the following is registered at the OS level:

- The API request sends a button press event to the guest.
- Various system services are stopped as a result of the button press event. Graceful shutdown is triggered by the ACPI shutdown button press event from the hypervisor.
Stop and start

- ACPI shutdown is initiated.
- The instance shuts down when the graceful shutdown process exits. There is no configurable OS shutdown time.
- If the instance OS does not shut down cleanly within a few minutes, a hard shutdown is performed.

By default, when you initiate a shutdown from an Amazon EBS-backed instance, the instance stops. You can change this behavior so that it terminates instead. For more information, see Change the instance initiated shutdown behavior (p. 445).

Stop and start your instances

You can stop and start your Amazon EBS-backed instance using the console or the command line.

New console

**To stop and start an Amazon EBS-backed instance using the console**

1. When you stop an instance, the data on any instance store volumes is erased. Before you stop an instance, verify that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.
2. In the navigation pane, choose **Instances** and select the instance.
3. Choose **Instance state, Stop instance**. If this option is disabled, either the instance is already stopped or its root device is an instance store volume.
4. When prompted for confirmation, choose **Stop**. It can take a few minutes for the instance to stop.
5. (Optional) While your instance is stopped, you can modify certain instance attributes. For more information, see **Modify a stopped instance** (p. 428).
6. To start the stopped instance, select the instance, and choose **Instance state, Start instance**.
7. It can take a few minutes for the instance to enter the **running** state.

Old console

**To stop and start an Amazon EBS-backed instance using the console**

1. When you stop an instance, the data on any instance store volumes is erased. Before you stop an instance, verify that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.
2. In the navigation pane, choose **Instances** and select the instance.
3. Choose **Actions, Instance State, Stop**. If this option is disabled, either the instance is already stopped or its root device is an instance store volume.
4. When prompted for confirmation, choose **Yes, Stop**. It can take a few minutes for the instance to stop.
5. (Optional) While your instance is stopped, you can modify certain instance attributes. For more information, see **Modify a stopped instance** (p. 428).
6. To start the stopped instance, select the instance, and choose **Actions, Instance State, Start**.
7. In the confirmation dialog box, choose **Yes, Start**. It can take a few minutes for the instance to enter the **running** state.

**To stop and start an Amazon EBS-backed instance using the command line**

You can use one of the following commands. For more information about these command line interfaces, see **Access Amazon EC2** (p. 3).
• **stop-instances** and **start-instances** (AWS CLI)
• **Stop-EC2Instance** and **Start-EC2Instance** (AWS Tools for Windows PowerShell)

### Modify 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 [Change the instance type (p. 231)](##).
- To change the user data for your instance, see [Work with instance user data (p. 594)](##).
- To enable or disable EBS-optimization for your instance, see [Modifying EBS–Optimization (p. 1360)](##).
- To change the `DeleteOnTermination` attribute of the root volume for your instance, see [Update the block device mapping of a running instance (p. 1421)](##). You are not required to stop the instance to change this attribute.

**To modify an instance attribute using the command line**

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

- **modify-instance-attribute** (AWS CLI)
- **Edit-EC2InstanceAttribute** (AWS Tools for Windows PowerShell)

### Troubleshoot stopping your instance

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 [Troubleshoot stopping your instance (p. 1495)](##).

### Hibernate your On-Demand or Reserved Windows instance

When you hibernate an instance, Amazon EC2 signals the operating system to perform hibernation (suspend-to-disk). Hibernation saves the contents from the instance memory (RAM) to your Amazon Elastic Block Store (Amazon EBS) root volume. Amazon EC2 persists the instance's EBS root volume and any attached EBS data volumes. When you start your instance:

- The EBS root volume is restored to its previous state
- The RAM contents are reloaded
- The processes that were previously running on the instance are resumed
- Previously attached data volumes are reattached and the instance retains its instance ID

You can hibernate an instance only if it's **enabled for hibernation** (p. 431) and it meets the hibernation prerequisites (p. 430).

If an instance or application takes a long time to bootstrap and build a memory footprint to become fully productive, you can use hibernation to pre-warm the instance. To pre-warm the instance, you:

1. Launch it with hibernation enabled.
2. Bring it to a desired state.
3. Hibernate it, ready to be resumed to the same state as needed.

You're not charged for instance usage for a hibernated instance when it is in the stopped state. You are charged for instance usage while the instance is in the stopping state, when the contents of the RAM are transferred to the EBS root volume. (This is different from when you stop an instance (p. 425) without hibernating it.) You're not charged for data transfer. However, you are charged for storage of any EBS volumes, including storage for the RAM contents.

If you no longer need an instance, you can terminate it at any time, including when it is in a stopped (hibernated) state. For more information, see Terminate your instance (p. 441).

Note
For information about using hibernation on Linux instances, see Hibernate your Linux instance in the Amazon EC2 User Guide for Linux Instances.
For information about hibernating Spot Instances, see Hibernate interrupted Spot Instances (p. 319).

Contents
- Overview of hibernation (p. 429)
- Hibernation prerequisites (p. 430)
- Limitations (p. 431)
- Enable hibernation for an instance (p. 431)
- Hibernate an instance (p. 434)
- Start a hibernated instance (p. 436)
- Troubleshoot hibernation (p. 437)

Overview of hibernation

The following diagram shows a basic overview of the hibernation process.

When you hibernate a running instance, the following happens:

- When you initiate hibernation, the instance moves to the stopping state. Amazon EC2 signals the operating system to perform hibernation (suspend-to-disk). The hibernation freezes all of the processes, saves the contents of the RAM to the EBS root volume, and then performs a regular shutdown.
- After the shutdown is complete, the instance moves to the stopped state.
- Any EBS volumes remain attached to the instance, and their data persists, including the saved contents of the RAM.
- Any Amazon EC2 instance store volumes remain attached to the instance, but the data on the instance store volumes is lost.
- In most cases, the instance is migrated to a new underlying host computer when it's started. This is also what happens when you stop and start an instance.
When you start the instance, the instance boots up and the operating system reads in the contents of the RAM from the EBS root volume, before unfreezing processes to resume its state.

The instance retains its private IPv4 addresses and any IPv6 addresses. When you start the instance, the instance continues to retain its private IPv4 addresses and any IPv6 addresses.

Amazon EC2 releases the public IPv4 address. When you start the instance, Amazon EC2 assigns a new public IPv4 address to the instance.

The instance retains its associated Elastic IP addresses. You're charged for any Elastic IP addresses associated with a hibernated instance. With EC2-Classic, an Elastic IP address is disassociated from your instance when you hibernate it. For more information, see EC2-Classic (p. 1022).

When you hibernate a ClassicLink instance, it's unlinked from the VPC to which it was linked. You must link the instance to the VPC again after starting it. For more information, see ClassicLink (p. 1030).

For information about how hibernation differs from reboot, stop, and terminate, see Differences between reboot, stop, hibernate, and terminate (p. 389).

**Hibernation prerequisites**

To hibernate an On-Demand Instance or Reserved Instance, the following prerequisites must be in place:

- **Supported Windows AMIs** (must be an HVM AMI that supports hibernation)
  - Windows Server 2012 AMI released 2019.09.11 or later
  - Windows Server 2012 R2 AMI released 2019.09.11 or later
  - Windows Server 2016 AMI released 2019.09.11 or later
  - Windows Server 2019 AMI released 2019.09.11 or later

For information about the supported Linux AMIs, see Hibernation prerequisites in the Amazon EC2 User Guide for Linux Instances.

- **Supported instance families**
  - Xen: C3, C4, I3, M3, M4, R3, R4, T2
  - Nitro: C5, C5d, M5, M5a, M5ad, M5d, R5, R5a, R5ad, R5d, T3*, T3a*

  * For hibernation, we recommend that you use a T3 or T3a instance with at least 1 GB of RAM.

- **Instance size** - not supported for bare metal instances.
- **Instance RAM size** - must be up to 16 GB.
- **Root volume type** - must be an EBS volume, not an instance store volume.
- **Supported EBS volume types** - General Purpose SSD (gp2 and gp3) or Provisioned IOPS SSD (io1 and io2). If you choose a Provisioned IOPS SSD volume type, to achieve optimum performance for hibernation, you must provision the EBS volume with the appropriate IOPS. For more information, see Amazon EBS volume types (p. 1163).
- **EBS root volume size** - must be large enough to store the RAM contents and accommodate your expected usage, for example, OS or applications. If you enable hibernation, space is allocated on the root volume at launch to store the RAM.
- **EBS root volume encryption** - To use hibernation, the root volume must be encrypted to ensure the protection of sensitive content that is in memory at the time of hibernation. When RAM data is moved to the EBS root volume, it is always encrypted. Encryption of the root volume is enforced at instance launch. Use one of the following three options to ensure that the root volume is an encrypted EBS volume:
  - EBS "single-step" encryption: You can launch encrypted EBS-backed EC2 instances from an unencrypted AMI and also enable hibernation at the same time. For more information, see Use encryption with EBS-backed AMIs (p. 127).
• EBS encryption by default: You can enable EBS encryption by default to ensure all new EBS volumes created in your AWS account are encrypted. This way, you can enable hibernation for your instances without specifying encryption intent at instance launch. For more information, see Encryption by default (p. 1331).

• Encrypted AMI: You can enable EBS encryption by using an encrypted AMI to launch your instance. If your AMI does not have an encrypted root snapshot, you can copy it to a new AMI and request encryption. For more information, see Encrypt an unencrypted image during copy (p. 132) and Copy an AMI (p. 117).

• Enable hibernation at launch - You cannot enable hibernation on an existing instance (running or stopped). For more information, see Enable hibernation for an instance (p. 431).

• Purchasing options - This feature is available for On-Demand Instances and Reserved Instances. It is not available for Spot Instances. For more information, see Hibernate interrupted Spot Instances (p. 319).

Limitations

• When you hibernate an instance, the data on any instance store volumes is lost.

• You can't hibernate an instance that has more than 16 GB of RAM.

• If you create a snapshot or AMI from an instance that is hibernated or has hibernation enabled, you might not be able to connect to the instance.

• You can't change the instance type or size of an instance with hibernation enabled.

• You can't hibernate an instance that is in an Auto Scaling group or used by Amazon ECS. If your instance is in an Auto Scaling group and you try to hibernate it, the Amazon EC2 Auto Scaling service marks the stopped instance as unhealthy, and might 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.

• You can't hibernate an instance that is configured to boot in UEFI mode.

• If you hibernate an instance that was launched into a Capacity Reservation, the Capacity Reservation does not ensure that the hibernated instance can resume after you try to start it.

• We do not support keeping an instance hibernated for more than 60 days. To keep the instance for longer than 60 days, you must start the hibernated instance, stop the instance, and start it.

• We constantly update our platform with upgrades and security patches, which can conflict with existing hibernated instances. We notify you about critical updates that require a start for hibernated instances so that we can perform a shutdown or a reboot to apply the necessary upgrades and security patches.

Enable hibernation for an instance

To hibernate an instance, it must first be enabled for hibernation. To enable hibernation, you must do it while launching the instance.

Important
You can't enable or disable hibernation for an instance after you launch it.

Console

To enable hibernation using the console

1. Follow the Launch an instance using the Launch Instance Wizard (p. 392) procedure.

2. On the Choose an Amazon Machine Image (AMI) page, select an AMI that supports hibernation. For more information about supported AMIs, see Hibernation prerequisites (p. 430).
3. On the **Choose an Instance Type** page, select a supported instance type, and choose **Next: Configure Instance Details**. For information about supported instance types, see [Hibernate prerequisites](p. 430).

4. On the **Configure Instance Details** page, for **Stop - Hibernate Behavior**, select the **Enable hibernation as an additional stop behavior** check box.

5. On the **Add Storage** page, for the root volume, specify the following information:
   
   - For **Size (GiB)**, enter the EBS root volume size. The volume must be large enough to store the RAM contents and accommodate your expected usage.
   - For **Volume Type**, select a supported EBS volume type, General Purpose SSD (gp2 and gp3) or Provisioned IOPS SSD (io1 and io2).
   - For **Encryption**, select the encryption key for the volume. If you enabled encryption by default in this AWS Region, the default encryption key is selected.

   For more information about the prerequisites for the root volume, see [Hibernate prerequisites](p. 430).

6. Continue as prompted by the wizard. When you've finished reviewing your options on the **Review Instance Launch** page, choose **Launch**. For more information, see [Launch an instance using the Launch Instance Wizard](p. 392).

**AWS CLI**

**To enable hibernation using the AWS CLI**

Use the `run-instances` command to launch an instance. Specify the EBS root volume parameters using the `--block-device-mappings file://mapping.json` parameter, and enable hibernation using the `--hibernation-options Configured=true` parameter.

```
aws ec2 run-instances \
   --image-id ami-0abcdef1234567890 \
   --instance-type m5.large \
   --block-device-mappings file://mapping.json \
   --hibernation-options Configured=true \
   --count 1 \
   --key-name MyKeyPair
```

Specify the following in `mapping.json`:

```
[
   {
      "DeviceName": "/dev/xvda",
      "Ebs": {
         "VolumeSize": 30,
         "VolumeType": "gp2",
         "Encrypted": true
      }
   }
]
```

**Note**

The value for `DeviceName` must match the root device name associated with the AMI. To find the root device name, use the `describe-images` command, as follows:

```
aws ec2 describe-images --image-id ami-0abcdef1234567890
```
If you enabled encryption by default in this AWS Region, you can omit "Encrypted":
true.

PowerShell

To enable hibernation using the AWS Tools for Windows PowerShell

Use the New-EC2Instance command to launch an instance. Specify the EBS root volume by first defining the block device mapping, and then adding it to the command using the -BlockDeviceMappings parameter. Enable hibernation using the -HibernationOptions_Configured $true parameter.

```powershell
PS C:\> $ebs_encrypt = New-Object Amazon.EC2.Model.BlockDeviceMapping
PS C:\> $ebs_encrypt.DeviceName = "/dev/xvda"
PS C:\> $ebs_encrypt.Ebs = New-Object Amazon.EC2.Model.EbsBlockDevice
PS C:\> $ebs_encrypt.Ebs.VolumeSize = 30
PS C:\> $ebs_encrypt.Ebs.VolumeType = "gp2"
PS C:\> $ebs_encrypt.Ebs.Encrypted = $true
PS C:\> New-EC2Instance
   -ImageId ami-0abcdef1234567890
   -InstanceType m5.large
   -BlockDeviceMappings $ebs_encrypt
   -HibernationOptions_Configured $true
   -MinCount 1
   -MaxCount 1
   -KeyName MyKeyPair
```

**Note**
The value for DeviceName must match the root device name associated with the AMI. To find the root device name, use the Get-EC2Image command, as follows:

```powershell
Get-EC2Image -ImageId ami-0abcdef1234567890
```

If you enabled encryption by default in this AWS Region, you can omit Encrypted = $true from the block device mapping.

New console

To view if an instance is enabled for hibernation using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance and, on the Details tab, in the Instance details section, inspect Stop- hibernate behavior. Enabled indicates that the instance is enabled for hibernation.

Old console

To view if an instance is enabled for hibernation using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance and, in the details pane, inspect Stop - Hibernation behavior. Enabled indicates that the instance is enabled for hibernation.
AWS CLI

To view if an instance is enabled for hibernation using the AWS CLI

Use the describe-instances command and specify the --filters "Name=hibernation-options.configured,Values=true" parameter to filter instances that are enabled for hibernation.

```bash
aws ec2 describe-instances \
--filters "Name=hibernation-options.configured,Values=true"
```

The following field in the output indicates that the instance is enabled for hibernation.

```json
"HibernationOptions": {
    "Configured": true
}
```

PowerShell

To view if an instance is enabled for hibernation using the AWS Tools for Windows PowerShell

Use the Get-EC2Instance command and specify the -Filter @{ Name="hibernation-options.configured"; Value="true"} parameter to filter instances that are enabled for hibernation.

```powershell
Get-EC2Instance \n    -Filter @{ Name="hibernation-options.configured"; Value="true"}
```

The output lists the EC2 instances that are enabled for hibernation.

Hibernate an instance

You can hibernate an instance if the instance is enabled for hibernation (p. 431) and meets the hibernation prerequisites (p. 430). If an instance cannot hibernate successfully, a normal shutdown occurs.

New console

To hibernate an Amazon EBS-backed instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select an instance, and choose Instance state, Hibernate instance. If Hibernate instance is disabled, the instance is already hibernated or stopped, or it can't be hibernated. For more information, see Hibernation prerequisites (p. 430).
4. When prompted for confirmation, choose Hibernate. It can take a few minutes for the instance to hibernate. The instance state changes to Stopping while the instance is hibernating, and then Stopped when the instance has hibernated.

Old console

To hibernate an Amazon EBS-backed instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select an instance, and choose Actions, Instance State, Stop - Hibernate. If Stop - Hibernate is disabled, the instance is already hibernated or stopped, or it can't be hibernated. For more information, see Hibernation prerequisites (p. 430).

4. In the confirmation dialog box, choose Yes, Stop - Hibernate. It can take a few minutes for the instance to hibernate. The Instance State changes to Stopping while the instance is hibernating, and then Stopped when the instance has hibernated.

AWS CLI

To hibernate an Amazon EBS-backed instance using the AWS CLI

Use the stop-instances command and specify the --hibernate parameter.

```
aws ec2 stop-instances \
  --instance-ids i-1234567890abcdef0 \
  --hibernate
```

PowerShell

To hibernate an Amazon EBS-backed instance using the AWS Tools for Windows PowerShell

Use the Stop-EC2Instance command and specify the -Hibernate $true parameter.

```
Stop-EC2Instance \n  -InstanceId i-1234567890abcdef0 \n  -Hibernate $true
```

New console

To view if hibernation was initiated on an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance and, on the Details tab, in the Instance details section, inspect State transition message. The message Client.UserInitiatedHibernate: User initiated hibernate indicates that hibernation was initiated on the instance.

Old console

To view if hibernation was initiated on an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance and, in the details pane, inspect State transition reason message. The message Client.UserInitiatedHibernate: User initiated hibernate indicates that hibernation was initiated on the instance.

AWS CLI

To view if hibernation was initiated on an instance using the AWS CLI

Use the describe-instances command and specify the state-reason-code filter to see the instances on which hibernation was initiated.
To view if hibernation was initiated on an instance using the AWS Tools for Windows PowerShell

Use the `Get-EC2Instance` command and specify the `state-reason-code` filter to see the instances on which hibernation was initiated.

```powershell
Get-EC2Instance -Filter @{Name="state-reason-code";Value="Client.UserInitiatedHibernate"}
```

The output lists the EC2 instances on which hibernation was initiated.

### Start a hibernated instance

Start a hibernated instance by starting it in the same way that you would start a stopped instance.

**New console**

To start a hibernated instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select a hibernated instance, and choose **Instance state, Start instance**. It can take a few minutes for the instance to enter the running state. During this time, the instance status checks (p. 810) show the instance in a failed state until the instance has started.

**Old console**

To start a hibernated instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select a hibernated instance, and choose **Actions, Instance State, Start**. It can take a few minutes for the instance to enter the running state. During this time, the instance status checks (p. 810) show the instance in a failed state until the instance has started.

**AWS CLI**

To start a hibernated instance using the AWS CLI

Use the `start-instances` command.

```bash
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```
PowerShell

To start a hibernated instance using the AWS Tools for Windows PowerShell

Use the Start-EC2Instance command.

```
Start-EC2Instance -InstanceId i-1234567890abcdef0
```

Troubleshoot hibernation

Use this information to help diagnose and fix issues that you might encounter when hibernating an instance.

Can't hibernate immediately after launch

If you try to hibernate an instance too quickly after you've launched it, you get an error.

You must wait for about five minutes after launch before hibernating.

Takes too long to transition from stopping to stopped, and memory state not restored after start

If it takes a long time for your hibernating instance to transition from the stopping state to stopped, and if the memory state is not restored after you start, this could indicate that hibernation was not properly configured.

Windows Server 2016 and later

Check the EC2 launch log and look for messages that are related to hibernation. To access the EC2 launch log, connect (p. 413) to the instance and open the C:\ProgramData\Amazon\EC2-Windows\Launch\Log\Ec2Launch.log file in a text editor.

Note

By default, Windows hides files and folders under C:\ProgramData. To view EC2 Launch directories and files, enter the path in Windows Explorer or change the folder properties to show hidden files and folders.

Find the log lines for hibernation. If the log lines indicate a failure or the log lines are missing, there was most likely a failure configuring hibernation at launch.

For example, the following message indicates that hibernation failed to configure: Message: Failed to enable hibernation.

If the log line contains HibernationEnabled: true, hibernation was successfully configured.

Windows Server 2012 R2 and earlier

Check the EC2 config log and look for messages that are related to hibernation. To access the EC2 config log, connect (p. 413) to the instance and open the C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt file in a text editor. Find the log lines for SetHibernateOnSleep. If the log lines indicate a failure or the log lines are missing, there was most likely a failure configuring hibernation at launch.

For example, the following message indicates that the instance root volume is not large enough:

SetHibernateOnSleep: Failed to enable hibernation: Hibernation failed with the following error: There is not enough space on the disk.
If the log line is SetHibernateOnSleep: HibernationEnabled: true, hibernation was successfully configured.

If you do not see any logs from these processes, your AMI might not support hibernation. For information about supported AMIs, see Hibernation prerequisites (p. 430).

**Instance size**

If you’re using a T3 or T3a instance with less than 1 GB of RAM, try increasing the size of the instance to one that has at least 1 GB of RAM.

**Instance "stuck" in the stopping state**

If you hibernated your instance and it appears "stuck" in the stopping state, you can forcibly stop it. For more information, see Troubleshoot stopping your instance (p. 1495).

**Reboot your instance**

An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance. When you reboot an instance, it keeps its public DNS name (IPv4), private and public IPv4 address, IPv6 address (if applicable), and any data on its instance store volumes.

Rebooting an instance doesn't start a new instance billing period (with a minimum one-minute charge), unlike stopping and starting your instance.

We might schedule your instance for a reboot for necessary maintenance, such as to apply updates that require a reboot. No action is required on your part; we recommend that you wait for the reboot to occur within its scheduled window. For more information, see Scheduled events for your instances (p. 816).

We recommend that you use the Amazon EC2 console, a command line tool, or the Amazon EC2 API to reboot your instance instead of running the operating system reboot command from your instance. If you use the Amazon EC2 console, a command line tool, or the Amazon EC2 API to reboot your instance, we perform a hard reboot if the instance does not cleanly shut down within a few minutes. If you use AWS CloudTrail, then using Amazon EC2 to reboot your instance also creates an API record of when your instance was rebooted.

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.

**New console**

**To reboot an instance using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance and choose Actions, Instance state, Reboot instance.
4. Choose Reboot when prompted for confirmation. The instance remains in the running state.

**Old console**

**To reboot an instance using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, Instance State, Reboot**.
4. Choose **Yes, Reboot** when prompted for confirmation. The instance remains in the running state.

**To reboot an instance using the command line**

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

- `reboot-instances` (AWS CLI)
- `Restart-EC2Instance` (AWS Tools for Windows PowerShell)

**Instance retirement**

An instance is scheduled to be retired when AWS detects irreparable failure of the underlying hardware that hosts the instance. When an instance reaches its scheduled retirement date, it is stopped 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.

For more information about the types of instance events, see Scheduled events for your instances (p. 816).

**Contents**

- Identify instances scheduled for retirement (p. 439)
- Actions to take for instances scheduled for retirement (p. 440)

**Identify instances scheduled for retirement**

If your instance is scheduled for retirement, you receive an email prior to the event with the instance ID and retirement date. You can also check for instances that are scheduled for retirement using the Amazon EC2 console or the command line.

**Important**

If an instance is scheduled for retirement, we recommend that you take action as soon as possible because the instance might be unreachable. (The email notification you receive states the following: "Due to this degradation your instance could already be unreachable.") For more information about the recommended action you should take, see Check if your instance is reachable.

**Ways to identify instances scheduled for retirement**

- Email notification (p. 439)
- Console identification (p. 440)

**Email notification**

If your instance is scheduled for retirement, you receive an email prior to the event with the instance ID and retirement date.

The email is sent to the primary account holder and the operations contact. For more information, see Adding, changing, or removing alternate contacts in the AWS Billing and Cost Management User Guide.
Console identification

If you use an email account that you do not check regularly for instance retirement notifications, you can use the Amazon EC2 console or the command line to determine if any of your instances are scheduled for retirement.

To identify instances scheduled for retirement using the console

1. Open the Amazon EC2 console.
2. In the navigation pane, choose EC2 Dashboard. Under Scheduled events, you can see the events that are associated with your Amazon EC2 instances and volumes, organized by Region.

   ![Scheduled events](image)

3. If you have an instance with a scheduled event listed, select its link below the Region name to go to the Events page.
4. The Events page lists all resources that have events associated with them. To view instances that are scheduled for retirement, select Instance resources from the first filter list, and then Instance stop or retirement from the second filter list.
5. If the filter results show that an instance is scheduled for retirement, select it, and note the date and time in the Start time field in the details pane. This is your instance retirement date.

To identify instances scheduled for retirement using the command line

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

- describe-instance-status (AWS CLI)
- Get-EC2InstanceStatus (AWS Tools for Windows PowerShell)

Actions to take for instances scheduled for retirement

To preserve the data on your retiring instance, you can perform one of the following actions. It's important that you take this action before the instance retirement date to prevent unforeseen downtime and data loss.

Check if your instance is reachable

When you are notified that your instance is scheduled for retirement, we recommend that you take the following action as soon as possible:

- Check if your instance is reachable by either connecting (p. 413) to or pinging your instance.
- If your instance is reachable, you should plan to stop/start your instance at an appropriate time before the scheduled retirement date, when the impact is minimal. For more information about stopping and starting your instance, and what to expect when your instance is stopped, such as the effect on public, private, and Elastic IP addresses that are associated with your instance, see Stop and start your instance (p. 425). Note that data on instance store volumes is lost when you stop and start your instance.
- If your instance is unreachable, you should take immediate action and perform a stop/start (p. 425) to recover your instance.
• Alternatively, if you want to terminate (p. 441) your instance, plan to do so as soon as possible so that you stop incurring charges for the instance.

Create a backup of your instance

Create an EBS-backed AMI from your instance so that you have a backup. To ensure data integrity, stop the instance before you create the AMI. You can wait for the scheduled retirement date when the instance is stopped, or stop the instance yourself before the retirement date. You can start the instance again at any time. For more information, see Create a custom Windows AMI (p. 37).

Launch a replacement instance

After you create an AMI from your instance, you can use the AMI to launch a replacement instance. From the Amazon EC2 console, select your new AMI and then choose **Actions, Launch**. Follow the wizard to launch your instance. For more information about each step in the wizard, see Launch an instance using the Launch Instance Wizard (p. 392).

Terminate your instance

You can delete your instance when you no longer need it. This is referred to as terminating your instance. As soon as the state of an instance changes to shutting-down or terminated, you stop incurring charges for that instance.

You can't connect to or start an instance after you've terminated it. However, you can launch additional instances using the same AMI. If you'd rather stop and start your instance, or hibernate it, see Stop and start your instance (p. 425) or Hibernate your On-Demand or Reserved Windows instance (p. 428). For more information, see Differences between reboot, stop, hibernate, and terminate (p. 389).

Contents

- Instance termination (p. 441)
- Terminating multiple instances with termination protection across Availability Zones (p. 442)
- What happens when you terminate an instance (p. 442)
- Terminate an instance (p. 443)
- Enable termination protection (p. 443)
- Change the instance initiated shutdown behavior (p. 445)
- Preserve Amazon EBS volumes on instance termination (p. 445)

Instance termination

After you terminate an instance, it remains visible in the console for a short while, and then the entry is automatically deleted. You cannot delete the terminated instance entry yourself. After an instance is terminated, resources such as tags and volumes are gradually disassociated from the instance and may no longer be visible on the terminated instance after a short while.

When an instance terminates, the data on any instance store volumes associated with that instance is deleted.

By default, Amazon EBS root device volumes are automatically deleted when the instance terminates. However, by default, any additional EBS volumes that you attach at launch, or any EBS volumes that you attach to an existing instance persist even after the instance terminates. This behavior is controlled by the volume's `DeleteOnTermination` attribute, which you can modify. For more information, see Preserve Amazon EBS volumes on instance termination (p. 445).

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 Enable termination protection (p. 443).

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 Change the instance initiated shutdown behavior (p. 445).

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 multiple instances with termination protection across Availability Zones**

If you terminate multiple instances across multiple Availability Zones, and one or more of the specified instances are enabled for termination protection, the request fails with the following results:

- The specified instances that are in the same Availability Zone as the protected instance are not terminated.
- The specified instances that are in different Availability Zones, where no other specified instances are protected, are successfully terminated.

For example, say you have the following instances:

<table>
<thead>
<tr>
<th>Instance</th>
<th>Availability Zone</th>
<th>Terminate protection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance A</td>
<td>us-east-1a</td>
<td>Disabled</td>
</tr>
<tr>
<td>Instance B</td>
<td></td>
<td>Disabled</td>
</tr>
<tr>
<td>Instance C</td>
<td>us-east-1b</td>
<td>Enabled</td>
</tr>
<tr>
<td>Instance D</td>
<td></td>
<td>Disabled</td>
</tr>
</tbody>
</table>

If you attempt to terminate all of these instances in the same request, the request reports failure with the following results:

- **Instance A** and **Instance B** are successfully terminated because none of the specified instances in **us-east-1a** are enabled for termination protection.
- **Instance C** and **Instance D** fail to terminate because at least one of the specified instances in **us-east-1b** (Instance C) is enabled for termination protection.

**What happens when you terminate an instance**

When an EC2 instance is terminated using the `terminate-instances` command, the following is registered at the OS level:

- The API request will send a button press event to the guest.
- Various system services will be stopped as a result of the button press event. **systemd** handles a graceful shutdown of the system. Graceful shutdown is triggered by the ACPI shutdown button press event from the hypervisor.
• ACPI shutdown will be initiated.
• The instance will shut down when the graceful shutdown process exits. There is no configurable OS shutdown time.

**Terminate an instance**

You can terminate an instance using the AWS Management Console or the command line.

By default, when you initiate a shutdown from an Amazon EBS-backed instance (using the `shutdown` or `poweroff` commands), the instance stops. The `halt` command does not initiate a shutdown. If used, the instance does not terminate; instead, it places the CPU into HLT and the instance remains running.

**New console**

**To terminate an instance using the console**

1. Before you terminate an instance, verify that you won't lose any data by checking that your Amazon EBS volumes won't be deleted on termination and that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.
2. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
3. In the navigation pane, choose **Instances**.
4. Select the instance, and choose **Instance state, Terminate instance**.
5. Choose **Terminate** when prompted for confirmation.

**Old console**

**To terminate an instance using the console**

1. Before you terminate an instance, verify that you won't lose any data by checking that your Amazon EBS volumes won't be deleted on termination and that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.
2. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](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 *Access Amazon EC2 (p. 3).*

- `terminate-instances` *(AWS CLI)*
- `Stop-EC2Instance` *(AWS Tools for Windows PowerShell)*

**Enable termination protection**

By default, you can terminate your instance using the Amazon EC2 console, command line interface, or API. To prevent your instance from being accidentally terminated using Amazon EC2, you can enable
termination protection for the instance. The DisableApiTermination attribute controls whether the instance can be terminated using the console, CLI, or API. By default, termination protection is disabled for your instance. You can set the value of this attribute when you launch the instance, while the instance is running, or while the instance is stopped (for Amazon EBS-backed instances).

The DisableApiTermination attribute does not prevent you from terminating an instance by initiating shutdown from the instance (using an operating system command for system shutdown) when the InstanceInitiatedShutdownBehavior attribute is set. For more information, see Change the instance initiated shutdown behavior (p. 445).

Limitations

You can't enable termination protection for Spot Instances—a Spot Instance is terminated when the Spot price exceeds the amount you're willing to pay for Spot Instances. However, you can prepare your application to handle Spot Instance interruptions. For more information, see Spot Instance interruptions (p. 317).

The DisableApiTermination attribute does not prevent Amazon EC2 Auto Scaling from terminating an instance. For instances in an Auto Scaling group, use the following Amazon EC2 Auto Scaling features instead of Amazon EC2 termination protection:

- To prevent instances that are part of an Auto Scaling group from terminating on scale in, use instance protection. For more information, see Instance Protection in the Amazon EC2 Auto Scaling User Guide.
- To prevent Amazon EC2 Auto Scaling from terminating unhealthy instances, suspend the ReplaceUnhealthy process. For more information, see Suspending and Resuming Scaling Processes in the Amazon EC2 Auto Scaling User Guide.
- To specify which instances Amazon EC2 Auto Scaling should terminate first, choose a termination policy. For more information, see Customizing the Termination Policy in the Amazon EC2 Auto Scaling User Guide.

To enable termination protection for an instance at launch time

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the dashboard, choose Launch Instance and follow the directions in the wizard.
3. On the Configure Instance Details page, select the Enable termination protection check box.

To enable termination protection for a running or stopped instance

1. Select the instance, and choose Actions, Instance Settings, Change Termination Protection.
2. Choose Yes, Enable.

To disable termination protection for a running or stopped instance

1. Select the instance, and choose Actions, Instance Settings, Change Termination Protection.
2. Choose Yes, Disable.

To enable or disable termination protection using the command line

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

- modify-instance-attribute (AWS CLI)
- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)
Change the instance initiated shutdown behavior

By default, when you initiate a shutdown from an Amazon EBS-backed instance (using a command such as `shutdown` or `poweroff`), the instance stops (Note that `halt` does not issue a `poweroff` command and, if used, the instance will not terminate; instead, it will place the CPU into HLT and the instance will remain running). You can change this behavior using the `InstanceInitiatedShutdownBehavior` attribute for the instance so that it terminates instead. You can update this attribute while the instance is running or stopped.

You can update the `InstanceInitiatedShutdownBehavior` attribute using the Amazon EC2 console or the command line. The `InstanceInitiatedShutdownBehavior` attribute only applies when you perform a shutdown from the operating system of the instance itself; it does not apply when you stop an instance using the StopInstances API or the Amazon EC2 console.

To change the shutdown behavior of an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance.
4. Choose Actions, Instance settings, Change shutdown behavior. The current behavior is selected.
5. To change the behavior, select Stop or Terminate from Shutdown behavior and then choose Apply.

To change the shutdown behavior of an instance using the command line

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

- `modify-instance-attribute` (AWS CLI)
- `Edit-EC2InstanceAttribute` (AWS Tools for Windows PowerShell)

Preserve Amazon EBS volumes on instance termination

When an instance terminates, Amazon EC2 uses the value of the `DeleteOnTermination` attribute for each attached Amazon EBS volume to determine whether to preserve or delete the volume.

The default value for the `DeleteOnTermination` attribute differs depending on whether the volume is the root volume of the instance or a non-root volume attached to the instance.

Root volume

By default, the `DeleteOnTermination` attribute for the root volume of an instance is set to `true`. Therefore, the default is to delete the root volume of the instance when the instance terminates. The `DeleteOnTermination` attribute can be set by the creator of an AMI as well as by the person who launches an instance. When the attribute is changed by the creator of an AMI or by the person who launches an instance, the new setting overrides the original AMI default setting. We recommend that you verify the default setting for the `DeleteOnTermination` attribute after you launch an instance with an AMI.

Non-root volume

By default, when you attach a non-root EBS volume to an instance (p. 1186), its `DeleteOnTermination` attribute is set to `false`. Therefore, the default is to preserve these volumes. After the instance terminates, you can take a snapshot of the preserved volume or attach it to another instance. You must delete a volume to avoid incurring further charges. For more information, see Delete an Amazon EBS volume (p. 1206).
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 View the EBS volumes in an instance block device mapping (p. 1421).

You can change the value of the `DeleteOnTermination` attribute for a volume when you launch the instance or while the instance is running.

**Examples**

- Change the root volume to persist at launch using the console (p. 446)
- Change the root volume to persist at launch using the command line (p. 446)
- Change the root volume of a running instance to persist using the command line (p. 447)

**Change the root volume to persist at launch using the console**

Using the console, you can change the `DeleteOnTermination` attribute when you launch an instance. To change this attribute for a running instance, you must use the command line.

**To change the root volume of an instance to persist at launch using the console**

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

You can verify the setting by viewing details for the root device volume on the instance's details pane. Next to **Block devices**, choose the entry for the root device volume. By default, **Delete on termination** is True. If you change the default behavior, **Delete on termination** is False.

**Change 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 Access 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`:

```json
[
    {
      "DeviceName": "/dev/sda1",
      "Ebs": {
        "DeleteOnTermination": false,
        "SnapshotId": "snap-1234567890abcdef0",
        "VolumeType": "gp2"
      }
    }
]
```
Change 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 Access 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. If the impaired instance has a public IPv4 address, the instance retains the public IPv4 address after recovery. If the impaired instance is in a placement group, the recovered instance runs in the placement group.

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

Topics
• Requirements (p. 448)
• Create an Amazon CloudWatch alarm to recover an instance (p. 448)
• Troubleshoot instance recovery failures (p. 448)

Requirements
The recover action is supported only on instances with the following characteristics:

• Uses one of the following instance types: C3, C4, C5, C5a, C5n, M3, M4, M5, M5a, M5n, M5zn, M6i, P3, R3, R4, R5, R5a, R5b, R5n, T2, T3, T3a, high memory (virtualized only), X1, X1e
• Runs in a virtual private cloud (VPC)
• Uses default or dedicated instance tenancy
• Has only EBS volumes (do not configure instance store volumes)

Create an Amazon CloudWatch alarm to recover an instance
For information about creating an Amazon CloudWatch alarm to recover an instance, see Add recover actions to Amazon CloudWatch alarms (p. 868).

Troubleshoot instance recovery failures
The following issues can cause automatic recovery of your instance to fail:

• Temporary, insufficient capacity of replacement hardware.
• The instance has an attached instance store storage, which is an unsupported configuration for automatic instance recovery.
• There is an ongoing Service Health Dashboard event that prevented the recovery process from successfully executing. Refer to http://status.aws.amazon.com/ for the latest service availability information.
• The instance has reached the maximum daily allowance of three recovery attempts.

The automatic recovery process attempts to recover your instance for up to three separate failures per day. If the instance system status check failure persists, we recommend that you manually stop and start the instance. For more information, see Stop and start your instance (p. 425).

Your instance may subsequently be retired if automatic recovery fails and a hardware degradation is determined to be the root cause for the original system status check failure.

Configure 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
• Configure a Windows instance using EC2Launch v2 (p. 449)
• Configure a Windows instance using EC2Launch (p. 486)
• Configure a Windows instance using the EC2Config service (p. 493)
Configure a Windows instance using EC2Launch v2

All supported instances of Amazon EC2 running Windows Server include the EC2Launch v2 launch agent (EC2Launch.exe). EC2Launch v2 performs tasks during instance startup and runs if an instance is stopped and later started, or restarted. EC2Launch v2 can also perform tasks on demand. Some of these tasks are automatically enabled, while others must be enabled manually. The EC2Launch v2 service supports all EC2Config and EC2Launch features.

This service uses a configuration file to control its operation. You can update the configuration file using either a graphical tool or by directly editing it as a single .yml file (agent-config.yml). The service binaries are located in the %ProgramFiles%\Amazon\EC2Launch directory.

EC2Launch v2 publishes Windows event logs to help you troubleshoot errors and set triggers. For more information, see Windows event logs (p. 479).

Supported operating systems

- Windows Server 2019 (Long-Term Servicing Channel and Semi-Annual Channel)
- Windows Server 2016
- Windows Server 2012 and 2012 R2
- Windows Server 2008 SP2 and 2008 R2

EC2Launch v2 section contents

- EC2Launch v2 overview (p. 449)
- Install the latest version of EC2Launch v2 (p. 452)
- Migrate to EC2Launch v2 (p. 453)
- Stop, restart, delete, or uninstall EC2Launch v2 (p. 454)
- Verify the EC2Launch v2 version (p. 455)
- Subscribe to EC2Launch v2 service notifications (p. 456)
- EC2Launch v2 settings (p. 456)
- Troubleshoot EC2Launch v2 (p. 477)
- EC2Launch v2 version histories (p. 484)

EC2Launch v2 overview

EC2Launch v2 is a service that performs tasks during instance startup and runs if an instance is stopped and later started, or restarted.
Overview topics

- Compare Amazon EC2 launch services (p. 450)
- EC2Launch v2 concepts (p. 450)
- EC2Launch v2 tasks (p. 451)

Compare Amazon EC2 launch services

The following table shows the major functional differences between EC2Config, EC2Launch v1, and EC2Launch v2.

<table>
<thead>
<tr>
<th>Feature</th>
<th>EC2Config</th>
<th>EC2Launch v1</th>
<th>EC2Launch v2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executed as</td>
<td>Windows Service</td>
<td>PowerShell Scripts</td>
<td>Windows Service</td>
</tr>
<tr>
<td>Configuration file</td>
<td>XML</td>
<td>XML</td>
<td>YAML</td>
</tr>
<tr>
<td>Set Administrator username</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>User data size</td>
<td>16 KB</td>
<td>16 KB</td>
<td>60 KB (compressed)</td>
</tr>
<tr>
<td>Local user data baked on AMI</td>
<td>No</td>
<td>No</td>
<td>Yes, configurable</td>
</tr>
<tr>
<td>Task configuration in user data</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Configurable wallpaper</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Customize task execution order</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Configurable tasks</td>
<td>15</td>
<td>9</td>
<td>20 at launch</td>
</tr>
<tr>
<td>Supports Windows Event Viewer</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Number of Event Viewer event types</td>
<td>2</td>
<td>0</td>
<td>30</td>
</tr>
</tbody>
</table>

EC2Launch v2 concepts

The following concepts are useful to understand when considering EC2Launch v2.

Task
A task can be invoked to perform an action on an instance. For a complete list of available tasks for EC2Launch v2, see EC2Launch v2 tasks (p. 451). Each task includes a set of stages in which it can run, a defined frequency, and inputs. Tasks can be configured in the agent-config file or through user-data.

Stages

A stage is a logical grouping of tasks that are run by the service. Some tasks can run only in a specific stage. Others can run in multiple stages. When using local data, you must specify the stage in which a task will run. When using user data, the stage is implied.

The following list shows the stages in the order in which they run:

1. Boot
2. Network
3. PreReady
4. PostReady
5. UserData

Frequency

Task frequency is used to schedule when tasks should run, depending on the boot context.

The following frequencies can be specified:

- Once — The task runs once, when the AMI has booted for the first time (finished Sysprep).
- Always — The task runs every time that the launch agent runs. The launch agent runs when:
  - an instance starts or restarts
  - the EC2Launch service runs
  - EC2Launch.exe run is invoked

agent-config

agent-config is a file that is located in the configuration folder for EC2Launch v2. It includes configuration for the boot, network, preready, and postready stages. This file is used to specify the configuration for an instance for tasks that should run when the AMI is either booted for the first time or for subsequent times.

By default, the EC2Launch v2 installation installs an agent-config file that includes recommended configurations that are used in standard Amazon Windows AMIs. You can update the configuration file to alter the default boot experience for your AMI that EC2Launch v2 specifies.

User data

User data is data that is configurable when you launch an instance. You can update user data to dynamically change how custom AMIs or quickstart AMIs are configured. EC2Launch v2 supports 60 kB user data input length. User data includes only the UserData stage, and therefore runs after the agent-config file. You can enter user data when you launch an instance using the launch instance wizard, or you can modify user data from the EC2 console. For more information about working with user data, see Run commands on your Windows instance at launch (p. 572).

EC2Launch v2 tasks

EC2Launch v2 can perform the following tasks at each boot:

- Set up new and optionally customized wallpaper that renders information about the instance.
Set the attributes for the administrator account that is created on the local machine.
Add DNS suffixes to the list of search suffixes. Only suffixes that do not already exist are added to the list.
Set drive letters for any additional volumes and extend them to use available space.
Write files to the disk, either from the internet or from the configuration. If the content is in the configuration, it can be base64 decoded or encoded. If the content is from the internet, it can be unzipped.
Execute scripts either from the internet or from the configuration. If the script is from the configuration, it can be base64 decoded. If the script is from the internet, it can be unzipped.
Execute a program with given arguments.
Set the computer name.
Send instance information to the Amazon EC2 console.
Send the RDP certificate thumbprint to the EC2 console.
Dynamically extend the operating system partition to include any unpartitioned space.
Execute user data. For more information about specifying user data, see EC2Launch v2 task configuration (p. 468).
Set persistent static routes to reach the metadata service and AWS KMS servers.
Set non-boot partitions to MBR or GPT.
Start the Systems Manager (SSM) service following Sysprep.
Optimize ENA settings.
Enable OpenSSH for later Windows versions.
Enable Jumbo Frames.
Set Sysprep to run with EC2Launch v2.
Publish Windows event logs.

Install the latest version of EC2Launch v2

EC2Launch v2 is currently available by download, by installation from SSM Distributor, and on all supported Windows AMIs.

Download

To install the latest version of EC2Launch v2, download the service from the following locations:

**Note**
AmazonEC2Launch.msi does not uninstall previous versions of the EC2 launch services, such as EC2Launch (v1) or EC2Config. To upgrade to EC2Launch v2 from an earlier launch service version, see Migrate to EC2Launch v2 (p. 453).

- **64Bit** — https://s3.amazonaws.com/amazon-ec2launch-v2/windows/amd64/latest/AmazonEC2Launch.msi
- **32Bit** — https://s3.amazonaws.com/amazon-ec2launch-v2/windows/386/latest/AmazonEC2Launch.msi

Install from AWS SSM Distributor

You can install the AWSEC2Launch-Agent package from AWS SSM Distributor. For instructions on how to install a package from SSM Distributor, see Install or update packages in the AWS SSM User Guide.

Use AMI with EC2Launch v2 preinstalled (non-production workloads)
EC2Launch v2 is preinstalled on the following AMIs. Do not use these AMIs for production workloads as they are intended only for you to verify if the EC2Launch v2 service works well with your existing processes and workloads. You can find these AMIs from the Amazon EC2 console or you can find them using the EC2 CLI and searching with the prefix EC2LaunchV2_Preview-Windows_Server-

- EC2LaunchV2_Preview-Windows_Server-2004-English-Core-Base
- EC2LaunchV2_Preview-Windows_Server-2019-English-Full-Base
- EC2LaunchV2_Preview-Windows_Server-2019-English-Core-Base
- EC2LaunchV2_Preview-Windows_Server-2016-English-Full-Base
- EC2LaunchV2_Preview-Windows_Server-2016-English-Core-Base
- EC2LaunchV2_Preview-Windows_Server-2012_R2_RTM-English-Full-Base
- EC2LaunchV2_Preview-Windows_Server-2012_R2_RTM-English-Core
- EC2LaunchV2_Preview-Windows_Server-2012_R2_RTM-English-Full-Base
- EC2LaunchV2_Preview-Windows_Server-2016-English-Full-SQL_2017_Express

**Installation options**

When you install or upgrade EC2Launch v2, your existing configuration, located at %ProgramData%/Amazon/EC2Launch/config/agent-config.yml, is not replaced. Perform a clean installation to overwrite an existing configuration to use the latest version.

You can perform a clean installation using the EC2Launch v2 interface or the command line.

**Perform a clean installation using the EC2Launch v2 user interface**

When you install EC2Launch v2, choose the Clean Install option under **Clean Existing Configuration**.

![Clean Existing Configuration](image)

**Perform a clean installation using the command line**

To perform a clean installation of EC2Launch v2 using the command line, run the following Windows command:

```
msiexec /i "C:\Users\Administrator\Desktop\AmazonEC2Launch.msi" ADDLOCAL="Basic,Clean" /q
```

**Migrate to EC2Launch v2**

The EC2Launch migration tool upgrades the installed launch agent (EC2Config and EC2Launch v1) by uninstalling it and installing EC2Launch v2. Applicable configurations from previous launch services are automatically migrated to the new service. The migration tool does not detect any
scheduled tasks linked to EC2Launch v1 scripts; therefore, it does not automatically set up those tasks in EC2Launch v2. To configure these tasks, edit the `agent-config.yml` (p. 468) file, or use the EC2Launch v2 settings dialog box (p. 456). For example, if an instance has a scheduled task that runs `InitializeDisks.ps1`, then after you run the migration tool, you must specify the volumes you want to initialize in the EC2Launch v2 settings dialog box. See Step 6 of the procedure to Change settings using the EC2Launch v2 settings dialog box (p. 456).

You can download the migration tool or install with an SSM RunCommand document.

You can download the tool from the following locations:

- **64Bit** — https://s3.amazonaws.com/amazon-ec2launch-v2-utils/MigrationTool/windows/amd64/latest/EC2LaunchMigrationTool.zip
- **32Bit** — https://s3.amazonaws.com/amazon-ec2launch-v2-utils/MigrationTool/windows/386/latest/EC2LaunchMigrationTool.zip

**Note**
You must run the EC2Launch v2 migration tool as an Administrator. EC2Launch v2 is installed as a service after you run the migration tool. It does not run immediately. By default, it runs during instance startup and runs if an instance is stopped and later started, or restarted.

Use the `AWSEC2Launch-RunMigration` SSM document to migrate to the latest EC2Launch version with SSM Run Command. The document does not require any parameters. For more information about using SSM Run Command, see AWS Systems Manager Run Command.

The migration tool applies the following configurations from EC2Config to EC2Launch v2.

- If `Ec2DynamicBootVolumeSize` is set to `false`, removes EC2Launch v2 boot stage
- If `Ec2SetPassword` is set to `Enabled`, sets EC2Launch v2 password type to `random`
- If `Ec2SetPassword` is set to `Disabled`, sets EC2Launch v2 password type to `donothing`
- If `SetDnsSuffixList` is set to `false`, removes EC2Launch v2 `setDnsSuffix` task
- If `EC2SetComputerName` is set to `true`, adds EC2Launch v2 `setHostName` task to `yaml` configuration

The migration tool applies the following configurations from EC2Launch v1 to EC2Launch v2.

- If `ExtendBootVolumeSize` is set to `false`, removes EC2Launch v2 boot stage
- If `AdminPasswordType` is set to `Random`, sets EC2Launch v2 password type to `random`
- If `AdminPasswordType` is set to `Specify`, sets EC2Launch v2 password type to static and password data to the password specified in `AdminPassword`
- If `SetWallpaper` is set to `false`, removes EC2Launch v2 `setWallpaper` task
- If `AddDnsSuffixList` is set to `false`, removes EC2Launch v2 `setDnsSuffix` task
- If `SetComputerName` is set to `true`, adds EC2Launch v2 `setHostName` task

**Stop, restart, delete, or uninstall EC2Launch v2**

You can manage the EC2Launch v2 service just as you would any other Windows service.

EC2Launch v2 runs once on boot and runs all of the configured tasks. After executing tasks, the service enters a stopped state. When you restart the service, the service will run all of the configured tasks again and return to a stopped state.

To apply updated settings to your instance, you can stop and restart the service. If you are manually installing EC2Launch v2, you must first stop the service first.
To stop the EC2Launch v2 service

1. Launch and connect to your Windows instance.
2. On the Start menu, choose Administrative Tools, and then open Services.
3. In the list of services, right-click Amazon EC2Launch, and select Stop.

To restart the EC2Launch v2 service

1. Launch and connect to your Windows instance.
2. On the Start menu, choose Administrative Tools, and then open Services.
3. In the list of services, right-click Amazon EC2Launch, and select Restart.

If you don’t need to update the configuration settings, create your own AMI, or use AWS Systems Manager, you can delete and uninstall the service. Deleting a service removes its registry subkey. Uninstalling a service removes the files, the registry subkeys, and any shortcuts to the service.

To delete the EC2Launch v2 service

1. Start a command prompt window.
2. Run the following command:

   ```sc delete EC2Launch```

To uninstall EC2Launch v2

1. Launch and connect to your Windows instance.
2. On the Start menu, select Control Panel.
3. Open Programs and Features.
4. In the list of programs, select Amazon EC2Launch v2, and select Uninstall.

Verify the EC2Launch v2 version

Use the following procedure to verify the version of EC2Launch v2 that is installed on your instances.

To verify the installed version of EC2Launch v2

1. Launch an instance from your AMI and connect to it.
2. In the Control Panel, select Programs and Features.
3. In the list of installed programs, look for Amazon EC2Launch. Its version number appears in the Version column.

For information about the EC2Launch v2 versions included in the Windows AMIs, see AWS Windows AMIs (p. 27).

For the latest version of EC2Launch v2, see EC2Launch v2 version history (p. 484).

For the latest version of the EC2Launch v2 migration tool, see EC2Launch v2 migration tool version history (p. 485).

You can receive notifications when new versions of the EC2Launch v2 service are released. For more information, see Subscribe to EC2Launch v2 service notifications (p. 456).
Subscribe to EC2Launch v2 service notifications

Amazon SNS can notify you when new versions of the EC2Launch v2 service are released. Use the following procedure to subscribe to these notifications.

**Subscribe to EC2Launch v2 notifications**

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-1:309726204594:amazon-ec2launch-v2`.
   b. For **Protocol**, choose **Email**.
   c. For **Endpoint**, enter 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 EC2Launch v2 service is released, we send notifications to subscribers. If you no longer want to receive these notifications, use the following procedure to unsubscribe.

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**.

**EC2Launch v2 settings**

This section contains information about how to configure settings for EC2Launch v2.

**Topics include:**

- Change settings using the EC2Launch v2 settings dialog box (p. 456)
- EC2Launch v2 directory structure (p. 462)
- Configure EC2Launch v2 using the CLI (p. 463)
- EC2Launch v2 task configuration (p. 468)
- EC2Launch v2 exit codes and reboots (p. 476)
- EC2Launch v2 and Sysprep (p. 476)

**Change settings using the EC2Launch v2 settings dialog box**

The following procedure describes how to use the EC2Launch v2 settings dialog box to enable or disable settings.

1. Launch and connect to your Windows instance.
2. From the Start menu, choose **All Programs**, and then navigate to **EC2Launch settings**.
3. On the General tab of the EC2Launch settings dialog box, you can enable or disable the following settings.
a. **Set Computer Name**

If this setting is enabled (it is disabled by default), the current host name is compared to the desired host name at each boot. If the host names do not match, the host name is reset, and the system then optionally reboots to pick up the new host name. If a custom host name is not specified, it is generated using the hexadecimal-formatted private IPv4 address, for example, `ip-AC1F4E6`. To prevent your existing host name from being modified, do not enable this setting.

b. **Extend Boot 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.

c. **Set Administrator Account**

When enabled, you can set the username and password attributes for the administrator account that is created on your local machine. If this feature is not enabled, an administrator account is not created on the system following Sysprep. Provide a password in `adminPassword` only if `adminPasswordType` is Specify.

The password types are defined as follows:

i. **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.

ii. **Specify**

   EC2Launch uses the password that you specify in `adminPassword`. If the password does not meet the system requirements, EC2Launch generates a random password instead. The password is stored in `agent-config.yml` as clear text and is deleted after Sysprep sets the administrator password. EC2Launch encrypts the password using the user's key.

iii. **DoNothing**

   EC2Launch uses the password that you specify in the `unattend.xml` file. If you don't specify a password in `unattend.xml`, the administrator account is disabled.

d. **Start SSM Service**

When selected, the Systems Manager service is enabled to start following Sysprep. EC2Launch v2 performs all of the tasks described earlier (p. 451), and the SSM Agent processes requests for Systems Manager capabilities, such as Run Command and State Manager.

You can use Run Command to upgrade your existing instances to use the latest version of the EC2Launch v2 service and SSM Agent. For more information, see [Update SSM Agent by using Run Command](https://docs.aws.amazon.com/systems-manager/latest/userguide/using-run-command.html) in the [AWS Systems Manager User Guide](https://docs.aws.amazon.com/systems-manager/latest/userguide/).  

e. **Optimize ENA**

When selected, ENA settings are configured to ensure that ENA Receive Side Scaling and Receive Queue Depth settings are optimized for AWS. For more information, see [Configure RSS CPU affinity](https://docs.aws.amazon.com/systems-manager/latest/userguide/configuring-rss-affinity.html).

f. **Enable SSH**

This setting enables OpenSSH for later Windows versions to allow for remote system administration.

g. **Enable Jumbo Frames**
Select to enable Jumbo Frames. Jumbo Frames can have unintended effects on your network communications, so ensure you understand how Jumbo Frames will impact your system before enabling. For more information about Jumbo Frames, see Jumbo frames (9001 MTU) (p. 987).

h. **Prepare for Imaging**

Select whether you want your EC2 instance to shut down with or without Sysprep. When you want to run Sysprep with EC2Launch v2, choose **Shutdown with Sysprep**.

4. On the **DNS Suffix** tab, you can select whether you want to add a DNS suffix list for DNS resolution of servers running in EC2, without providing the fully qualified domain name. DNS suffixes can contain the variables $REGION and $AZ. Only suffixes that do not already exist will be added to the list.

5. On the **Wallpaper** tab, you can enable the display of selected instance details on the wallpaper. You also have the option of choosing a custom image. The details are generated each time that you log in. Clear the check box to remove instance details from the wallpaper.
6. On the **Volumes** tab, select whether you want to initialize the volumes that are attached to the instance. Enabling sets drive letters for any additional volumes and extends them to use available space. If you select **All**, all of the storage volumes are initialized. If you select **Devices**, only devices that are specified in the list are initialized. You must enter the device for each device to be initialized. Use the devices listed on the EC2 console, for example, `xvdb` or `/dev/nvme0n1`. The dropdown list displays the storage volumes that are attached to the instance. To enter a device that is not attached to the instance, enter it in the text field.

**Name**, **Letter**, and **Partition** are optional fields. If no value is specified for **Partition**, storage volumes larger than 2 TB are initialized with the GPT partition type, and those smaller than 2 TB are initialized with the MBR partition type. If devices are configured, and a non-NTFS device either contains a partition table, or the first 4 KB of the disk contain data, then the disk is skipped and the action logged.
Amazon EC2Launch settings

General | DNS suffix | Wallpaper | Volumes

Initialize volumes

- Initialize
- All
- Devices

Devices
If you choose Devices, only the devices listed below are initialized. You must enter the Device for each device to be initialized. Use the devices listed on the EC2 console, for example, xvdb or /dev/nvme0n1. Name, Letter, and Partition are optional.

<table>
<thead>
<tr>
<th>Device</th>
<th>Name</th>
<th>Letter</th>
<th>Partition</th>
</tr>
</thead>
</table>

Save   Exit
The following is an example configuration YAML file created from the settings entered in the EC2Launch dialog.

```yaml
version: 1.0
config:
  - stage: boot
tasks:
    - task: extendRootPartition
  - stage: preReady
tasks:
    - task: activateWindows
      inputs:
        activation:
          type: amazon
    - task: setDnsSuffix
      inputs:
        suffixes:
          - $REGION.ec2-utilities.amazonaws.com
    - task: setAdminAccount
      inputs:
        password:
          type: random
    - task: setWallpaper
      inputs:
        path: C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg
        attributes:
          - hostName
          - instanceId
          - privateIpAddress
          - publicIpAddress
          - instanceSize
          - availabilityZone
          - architecture
          - memory
          - network
  - stage: postReady
tasks:
    - task: startSsm
```

**EC2Launch v2 directory structure**

EC2Launch v2 should be installed in the following directories:

- Service binaries: `%ProgramFiles%\Amazon\EC2Launch`
- Service data (settings, log files, and state files): `%ProgramData%\Amazon\EC2Launch`

**Note**

By default, Windows hides files and folders under `C:\ProgramData`. To view EC2Launch v2 directories and files, you must either enter the path in Windows Explorer or change the folder properties to show hidden files and folders.

The `%ProgramFiles%\Amazon\EC2Launch` directory contains binaries and supporting libraries. It includes the following subdirectories:

- settings
  - `EC2LaunchSettingsUI.exe` — user interface for modifying the `agent-config.yml` file
  - `YamlDotNet.dll` — DLL for supporting some operations in the user interface
- tools
  - `ebsnvme-id.exe` — tool for examining the metadata of the EBS volumes on the instance
• AWSAcpiSpcrReader.exe — tool for determining the correct COM port to use
• EC2LaunchEventMessage.dll — DLL for supporting the Windows event logging for EC2Launch.

• service
  • EC2LaunchService.exe — Windows service executable that is launched when the launch agent runs as a service.
  • EC2Launch.exe — main EC2Launch executable
  • EC2LaunchAgentAttribution.txt — attribution for code used within EC2 Launch

The %ProgramData%\Amazon\EC2Launch directory contains the following subdirectories. All of the data produced by the service, including logs, configuration, and state, is stored in this directory.

• config — Configuration

  The service configuration file is stored in this directory as agent-config.yml. This file can be updated to modify, add, or remove tasks run by the service by default.

• log — Instance logs

  Logs for the service (agent.log), console (console.log), performance (bench.log), and errors (error.log) are stored in this directory. Log files are appended to on subsequent executions of the service.

• state — Service state data

  The state that the service uses to determine which tasks should run is stored here. There is a .run-once file that indicates whether the service has already run after Sysprep (so tasks with a frequency of once will be skipped on the next run). This subdirectory includes a state.json and previous-state.json to track the status of each task.

• sysprep — Sysprep

  This directory contains files that are used to determine which operations to perform by Sysprep when it creates a customized Windows AMI that can be reused.

Configure EC2Launch v2 using the CLI

You can use the Command Line Interface (CLI) to configure your EC2Launch settings and manage the service. The following section contains descriptions and usage information for the CLI commands that you can use to manage EC2Launch v2.

Commands

• collect-logs (p. 464)
• get-agent-config (p. 464)
• list-volumes (p. 465)
• reset (p. 465)
• run (p. 465)
• status (p. 466)
• sysprep (p. 466)
• validate (p. 467)
• version (p. 467)
• wallpaper (p. 467)
**collect-logs**

Collects log files for EC2Launch, zips the files, and places them in a specified directory.

**Example**

```bash
e2launch collect-logs -o C:\Mylogs.zip
```

**Usage**

e2launch collect-logs [flags]

**Flags**

- `-h, --help`
  
  help for collect-logs

- `-o, --output string`
  
  path to zipped output log files

**get-agent-config**

Prints `agent-config.yml` in the format specified (JSON or YAML). If no format is specified, `agent-config.yml` is printed in the format previously specified.

**Example**

```bash
e2launch get-agent-config -f json
```

**Example 2**

The following PowerShell commands show how to edit and save the `agent-config` file in JSON format.

```powershell
#config = e2launch get-agent-config --format json | ConvertFrom-Json
#jumboFrame =
{
    "task": "enableJumboFrames"
}
@#config.config | %{$_.stage -eq 'postReady'}{$_.tasks += (ConvertFrom-Json -InputObject $jumboFrame)}
#config | ConvertTo-Json -Depth 6 | Out-File -encoding UTF8 $env:ProgramData/Amazon/EC2Launch/config/agent-config.yml
```

**Usage**

e2launch get-agent-config [flags]

**Flags**

- `-h, --help`
  
  help for get-agent-config

- `-f, --format string`
  
  format string
output format of agent-config file: json, yaml

list-volumes
Lists all of the storage volumes attached to the instance, including ephemeral and EBS volumes.

Example

ec2launch list-volumes

Usage
ec2launch list-volumes

Flags
-h, --help
help for list-volumes
reset
Deletes the .runonce file so that tasks specified to run once will run on the next execution; optionally deletes the service and sysprep logs.

Example

ec2launch reset -c

Usage
ec2launch reset [flags]

Flags
-c, --clean
cleans instance logs before reset
-h, --help
help for reset
run
Runs EC2Launch v2.

Example

ec2launch run

Usage
ec2launch run [flags]

Flags
status

Gets the status of the EC2Launch service. Optionally blocks the process until the service is finished. The process exit code determines the service state:

- 0 — the service ran and was successful.
- 1 — the service ran and failed.
- 2 — the service is still running.

Example:

```
ec2launch status -b
```

Usage

```
ec2launch status [flags]
```

Flags

- `-b, --block`
blocks the process until the services finishes running

- `-h, --help`
help for status

sysprep

Resets the service state, updates `unattend.xml`, disables RDP, and runs Sysprep.

Example:

```
ec2launch sysprep
```

Usage

```
ec2launch sysprep [flags]
```

Flags

- `-c, --clean`
cleans instance logs before sysprep

- `-h, --help`
help for Sysprep

- `-s, --shutdown`
shuts down the instance after sysprep
validate

Validates the agent-config file C:\ProgramData\Amazon\EC2LaunchAgent\config\agent-config.yml.

Example

ec2launch validate

Usage

ec2launch validate [flags]

Flags

-h,--help
help for validate

version

Gets the executable version.

Example

ec2launch version

Usage

ec2launch version [flags]

Flags

-h,--help
help for version

wallpaper

Sets new wallpaper to the wallpaper path that is provided (.jpg file), and displays the selected instance details.

Example

ec2launch wallpaper ^
--path="C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg" ^
--attributes=hostName,instanceId,privateIpAddress,publicIpAddress,instanceSize,availabilityZone,architecture,memory,network

Usage

ec2launch wallpaper [flags]

Flags

--attributes strings
wallpaper attributes
EC2Launch v2 task configuration

This section includes the configuration tasks, details, and examples for the `agent-config.yml` and `user-data.yml` files.

**Tasks and examples**

- `activateWindows` (p. 468)
- `enableJumboFrames` (p. 469)
- `enableOpenSsh` (p. 469)
- `executeProgram` (p. 469)
- `executeScript` (p. 470)
- `extendRootPartition` (p. 471)
- `initializeVolume` (p. 471)
- `optimizeEna` (p. 472)
- `setAdminAccount` (p. 472)
- `setDnsSuffix` (p. 473)
- `setHostName` (p. 473)
- `setWallpaper` (p. 474)
- `startSsm` (p. 474)
- `sysprep` (p. 474)
- `writeFile` (p. 475)
- Example: `agent-config.yml` (p. 475)
- Example: `user data` (p. 476)

**activateWindows**

Activates Windows against a set of AWS KMS servers.

*Frequency* — once

*AllowedStages* — [PreReady]

*Inputs* —

```
activation: (map)
type: (string) activation type to use, set to amazon
```

*Example*

```yaml
task: activateWindows
inputs:
  activation:
    type: amazon
```
**enableJumboFrames**

Enables Jumbo Frames, which increase the maximum transmission unit (MTU) of the network adapter. For more information, see Jumbo frames (9001 MTU) (p. 987).

*Frequency* — always

*AllowedStages* — [PostReady, UserData]

*Inputs* — none

*Example*

```plaintext
task: enableJumboFrames
```

**enableOpenSsh**

Enables Windows OpenSSH and adds the public key for the instance to the authorized keys folder.

*Frequency* — once

*AllowedStages* — [PreReady, UserData]

*Inputs* — none

*Example*

The following example shows how to enable OpenSSH on an instance, and to add the public key for the instance to the authorized keys folder. This configuration works only on instances running Windows Server 2019.

```plaintext
task: enableOpenSsh
```

**executeProgram**

Executes a program with optional arguments and a specified frequency.

*Frequency* — see *Inputs*

*AllowedStages* — [PostReady, UserData]

*Inputs* —

  - *frequency*: (string) one of once or always
  - *path*: (string) path to the executable
  - *arguments*: (list of strings) list of string arguments to pass to the executable
  - *runAs*: (string) must be set to localSystem

*Example*

The following example shows how to run an executable file that is already on an instance.

```plaintext
task: executeProgram
inputs:
- frequency: always
  path: C:\Users\Administrator\Desktop\setup.exe
```
Example 2

The following example shows how to run an executable file that is already on an instance. This configuration installs a VLC .exe file that is present on the C: drive of the instance. /L=1033 and /S are VLC arguments passed as a string list with the VLC .exe file.

```
task: executeProgram
inputs:
  - frequency: always
    path: C:\vlc-3.0.11-win64.exe
    arguments: ['/L=1033','/S']
  runAs: localSystem
```

eexecuteScript

Executes a script with optional arguments and a specified frequency.

Frequency — see Inputs

AllowedStages — [PostReady, UserData]

Inputs —

frequency: (string) one of once or always

type: (string) one of batch or powershell

arguments: (list of strings) list of string arguments to pass to the shell. This parameter is not supported when type is set to batch.

content: (string) contents of the script

runAs: (string) one of admin or localSystem

detach: (boolean) defaults to false. Set to true if the script should be run in detached mode, where EC2Launch runs it and continues with other tasks. Script exit codes have no effect in this mode.

Example

```
task: executeScript
inputs:
  - frequency: always
    type: powershell
    content: |
      Get-Process | Out-File -FilePath .\Process.txt
  runAs: localSystem
```

Example 2

The following example shows how to run a PowerShell script on an EC2 instance. This configuration creates a text file in the C: drive.

```
task: executeScript
inputs:
  - frequency: always
    type: powershell
    runAs: admin
```
Example 3

The following example shows an idempotent script that reboots an instance multiple times.

```
task: executeScript
inputs:
- frequency: always
  type: powershell
  runAs: localSystem
content: |
    $name = $env:ComputerName
    if ($name -ne $desiredName) {
        Rename-Computer -NewName $desiredName
        exit 3010
    }
    $domain = Get-ADDomain
    if ($domain -ne $desiredDomain) {
        Add-Computer -DomainName $desiredDomain
        exit 3010
    }
    $telnet = Get-WindowsFeature -Name Telnet-Client
    if (-not $telnet.Installed) {
        Install-WindowsFeature -Name "Telnet-Client"
        exit 3010
    }
```

extendRootPartition

Extends the root volume to use all of the available space on the disk.

*Frequency* — once

*AllowedStages* — [Boot]

*Inputs* — none

*Example*

```
task: extendRootPartition
```

initializeVolume

Initializes volumes attached to the instance so that they are activated and partitioned. Any volumes that are detected as not empty are not initialized. A volume is considered empty if the first 4 KiB of a volume are empty, or if a volume does not have a Windows-recognizable drive layout. The volume letter field is always applied when this task runs, regardless of whether the drive is already initialized.

*Frequency* — always

*AllowedStages* — [PostReady, UserData]

*Inputs* —

- *initialize* *(string)* type of initialization strategy to use; one of all or devices
devices: (list of maps)

device: device identifier used when creating the instance; some examples are xvdb, xvdf, or /dev/nvme0n1

name: (string) drive name to assign

letter: (string) drive letter to assign

partition: (string) partitioning type to use; one of mbr or gpt

Example 1

The following example shows inputs for the InitializeVolume task to set selected volumes to be initialized.

```
task: initializeVolume
inputs:
  initialize: devices
  devices:
  - device: xvdb
    name: MyVolumeOne
    letter: D
    partition: mbr
  - device: /dev/nvme0n1
    name: MyVolumeTwo
    letter: E
    partition: gpt
```

Example 2

The following example shows how to initialize EBS volumes that are attached to an instance. This configuration will initialize all empty EBS volumes that are attached to the instance. If a volume is not empty, then it will not be initialized.

```
task: initializeVolume
inputs:
  initialize: all
```

optimizeEna

Optimizes ENA settings based on the current instance type; might reboot the instance.

Frequency — always

AllowedStages — [PostReady, UserData]

Inputs — none

Example

```
task: optimizeEna
```

setAdminAccount

Sets attributes for the default administrator account that is created on the local machine.

Frequency — once
### setAdminAccount

**AllowedStages** — [PreReady]

**Inputs** —

- name: (string) name of the administrator account
- password: (map)
  - type: (string) strategy to set the password, either as static, random, or doNothing
  - data: (string) stores data if the type field is static

**Example**

```yaml
task: setAdminAccount
inputs:
  name: Administrator
  password:
    type: random
```

### setDnsSuffix

Adds DNS suffixes to the list of search suffixes. Only suffixes that do not already exist are added to the list.

**Frequency** — always

**AllowedStages** — [PreReady]

**Inputs** —

- suffixes: (list of strings) list of one or more valid DNS suffixes; valid substitution variables are $REGION and $AZ

**Example**

```yaml
task: setDnsSuffix
inputs:
  suffixes:
    - $REGION.ec2-utilities.amazonaws.com
```

### setHostName

Sets the hostname of the computer to a custom string or, if hostName is not specified, the private IPv4 address.

**Frequency** — always

**AllowedStages** — [PostReady, UserData]

**Inputs** —

- hostName: (string) optional host name, which must be formatted as follows.
  - Must be 15 characters or less
  - Must contain only alphanumeric (a-z, A-Z, 0-9) and hyphen (-) characters.
  - Must not consist entirely of numerical characters.

- reboot: (boolean) denotes whether a reboot is permitted when the hostname is changed
Example

```
task: setHostName
inputs:
  reboot: true
```

**setWallpaper**

Sets up the instance with custom wallpaper that displays instance attributes.

*Frequency* — always

*AllowedStages* — [PreReady, UserData]

*Inputs* —

- `path` (string) path to a local .jpg file to use as the wallpaper image
- `attributes` (list of strings) list of attributes to add to the wallpaper; one of `hostName`, `instanceId`, `privateIpAddress`, `publicIpAddress`, `instanceSize`, `availabilityZone`, `architecture`, `memory`, or `network`

Example

```
task: setWallpaper
inputs:
  path: C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg
  attributes:
    - hostName
    - instanceId
    - privateIpAddress
    - publicIpAddress
```

**startSsm**

Starts the Systems Manager (SSM) service following Sysprep.

*Frequency* — always

*AllowedStages* — [PostReady, UserData]

*Inputs* — none

Example

```
task: startSsm
```

**sysprep**

Resets the service state, updates `unattend.xml`, disables RDP, and runs Sysprep. This task runs only after all other tasks are completed.

*Frequency* — once

*AllowedStages* — [UserData]

*Inputs* —

- `clean` (boolean) cleans instance logs before running Sysprep
shutdown: (boolean) shuts down the instance after running Sysprep

Example

task: sysprep
inputs:
  clean: true
  shutdown: true

writeFile

Writes a file to a destination.

Frequency — see Inputs

AllowedStages — [PostReady, UserData]

Inputs —

frequency: (string) one of once or always
destination: (string) path to which to write the content
content: (string) text to write to the destination

Example

task: writeFile
inputs:
  - frequency: once
  destination: C:\Users\Administrator\Desktop\booted.txt
  content: Windows Has Booted

Example: agent-config.yml

The following example shows settings for the agent-config.yml configuration file.

version: 1.0
config:
  - stage: boot
tasks:
  - task: extendRootPartition
  - stage: preReady
tasks:
  - task: activateWindows
    inputs:
      activation:
        type: amazon
  - task: setDnsSuffix
    inputs:
      suffixes:
        - $REGION.ec2-utilities.amazonaws.com
  - task: setAdminAccount
    inputs:
      password:
        type: random
  - task: setWallpaper
    inputs:
      path: C:\ProgramData\Amazon\EC2Launch\wallpaper\Ec2Wallpaper.jpg
      attributes:
        - hostName
Example: user data

For more information about user data, see Run commands on your Windows instance at launch (p. 572).

The following example shows settings for user data.

```yaml
version: 1.0
tasks:
  - task: executeScript
    inputs:
      - frequency: always
        type: powershell
        runAs: localSystem
        content: |
          New-Item -Path 'C:\PowerShellTest.txt' -ItemType File
```

The following format is compatible with the previous version of this service.

```powershell
$file = $env:SystemRoot + "\Temp" + (Get-Date).ToString("MM-dd-yy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

EC2Launch v2 exit codes and reboots

You can use EC2Launch v2 to define how exit codes are handled by your scripts. By default, the exit code of the last command that is run in a script is reported as the exit code for the entire script. For example, if a script includes three commands and the first command fails but the following ones succeed, the run status is reported as success because the final command succeeded.

If you want a script to reboot an instance, then you must specify exit 3010 in your script, even when the reboot is the last step in your script. exit 3010 instructs EC2Launch v2 to reboot the instance and call the script again until it returns an exit code that is not 3010, or until the maximum reboot count has been reached. EC2Launch v2 permits a maximum of 5 reboots per task. If you attempt to reboot an instance from a script by using a different mechanism, such as Restart-Computer, then the script run status will be inconsistent. For example, it may get stuck in a restart loop or not perform the restart.

If you are using a legacy user data format that is compatible with older agents, the user data may run more times than you intend it to. For more information, see Service runs user data more than once (p. 478) in the Troubleshooting section.

EC2Launch v2 and Sysprep

The EC2Launch v2 service runs Sysprep, a Microsoft tool that enables you to create a customized Windows AMI that can be reused. When EC2Launch v2 calls Sysprep, it uses the files in %ProgramData
%\Amazon\EC2Launch to determine which operations to perform. You can edit these files indirectly using the **EC2Launch settings** dialog box, or directly using a YAML editor or a text editor. However, there are some advanced settings that aren’t available in the **EC2Launch settings** 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 [Create a custom Windows AMI](#) (p. 37).

**Troubleshoot EC2Launch v2**

This section shows common troubleshooting scenarios for EC2Launch v2, information about viewing Windows event logs, and console log output and messages.

**Troubleshooting topics**

- **Common troubleshooting scenarios** (p. 477)
- **Windows event logs** (p. 479)
- **EC2Launch v2 console log output** (p. 482)

**Common troubleshooting scenarios**

This section shows common troubleshooting scenarios and steps for resolution.

**Scenarios**

- **Service fails to set the wallpaper** (p. 477)
- **Service fails to run user data** (p. 477)
- **Service runs a task only one time** (p. 478)
- **Service fails to run a task** (p. 478)
- **Service runs user data more than once** (p. 478)
- **Scheduled tasks from EC2Launch v1 fail to run after migration to EC2Launch v2** (p. 478)
- **Service fails to run a task** (p. 478)
- **Service initializes an EBS volume that is not empty** (p. 479)

**Service fails to set the wallpaper**

**Resolution**

1. Check that %AppData%\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\setwallpaper.lnk exists.
2. Check %ProgramData%\Amazon\EC2Launch\log\agent.log to see if any errors occurred.

**Service fails to run user data**

**Possible cause:** Service may have failed before running user data.

**Resolution**

1. Check %ProgramData%\Amazon\EC2Launch\state\previous-state.json.
2. See if boot, network, preReady, and postReadyLocalData have all been marked as success.
3. If one of the stages failed, check %ProgramData%\Amazon\EC2Launch\log\agent.log for specific errors.
Service runs a task only one time

Resolution
1. Check the frequency of the task.
2. If the service already ran after Sysprep, and the task frequency is set to once, the task will not run again.
3. Set the frequency of the task to always if you want it to run the task every time EC2Launch v2 runs.

Service fails to run a task

Resolution
1. Check the latest entries in %ProgramData%\Amazon\EC2Launch\log\agent.log.
2. If no errors occurred, try running the service manually from "%ProgramFiles%\Amazon \EC2Launch\EC2Launch.exe" run to see if the tasks succeed.

Service runs user data more than once

Resolution
User data is handled differently between EC2Launch v1 and EC2Launch v2. EC2Launch v1 runs user data as a scheduled task on the instance when persist is set to true. If persist is set to false, the task is not scheduled even when it exits with a reboot or is interrupted while running.

EC2Launch v2 runs user data as an agent task and tracks its run state. If user data issues a computer restart or if user data was interrupted while running, the run state persists as pending and the user data will run again at the next instance boot. If you want to prevent the user data script from running more than once, make the script idempotent.

The following example idempotent script sets the computer name and joins a domain.

```powershell
$name = $env:computername
if ($name -ne $desiredName) {
    Rename-Computer -NewName $desiredName
}
$domain = Get-ADDomain
if ($domain -ne $desiredDomain) {
    Add-Computer -DomainName $desiredDomain
}
$telnet = Get-WindowsFeature -Name Telnet-Client
if (-not $telnet.Installed) {
    Install-WindowsFeature -Name "Telnet-Client"
}
</powershell>
<persist>false</persist>

Scheduled tasks from EC2Launch v1 fail to run after migration to EC2Launch v2

Resolution
The migration tool does not detect any scheduled tasks linked to EC2Launch v1 scripts; therefore, it does not automatically set up those tasks in EC2Launch v2. To configure these tasks, edit the agent-config.yml (p. 468) file, or use the EC2Launch v2 settings dialog box (p. 456). For example, if an instance has a scheduled task that runs InitializeDisks.ps1, then after you run the migration tool,
you must specify the volumes you want to initialize in the EC2Launch v2 settings dialog box. See Step 6 of the procedure to Change settings using the EC2Launch v2 settings dialog box (p. 456).

Service fails to run a task

Resolution

1. Check the latest entries in %ProgramData%\Amazon\EC2Launch\log\agent.log.
2. If no errors occurred, try running the service manually from "%ProgramFiles%\Amazon\EC2Launch\EC2Launch.exe" run to see if the tasks succeed.

Service initializes an EBS volume that is not empty

Resolution

Before it initializes a volume, EC2Launch v2 attempts to detect whether it is empty. If a volume is not empty, it skips the initialization. Any volumes that are detected as not empty are not initialized. A volume is considered empty if the first 4 KiB of a volume are empty, or if a volume does not have a Windows-recognizable drive layout. A volume that was initialized and formatted on a Linux system does not have a Windows-recognizable drive layout, for example MBR or GPT. Therefore, it will be considered as empty and initialized. If you want to preserve this data, do not rely on EC2Launch v2 empty drive detection. Instead, specify volumes that you would like to initialize in the EC2Launch v2 settings dialog box (p. 456) (see step 6) or in the agent-config.yml (p. 471).

Windows event logs

EC2Launch v2 publishes Windows event logs for important events, such as service starting, Windows is ready, and task success and failure. Event identifiers uniquely identify a particular event. Each event contains stage, task, and level information, and a description. You can set triggers for specific events using the event identifier.

Event IDs provide information about an event and uniquely identify some events. The least significant digit of an event ID indicates the severity of an event.

<table>
<thead>
<tr>
<th>Event</th>
<th>Least significant digit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success</td>
<td>. . .0</td>
</tr>
<tr>
<td>Informational</td>
<td>. . .1</td>
</tr>
<tr>
<td>Warning</td>
<td>. . .2</td>
</tr>
<tr>
<td>Error</td>
<td>. . .3</td>
</tr>
</tbody>
</table>

Service-related events that are generated when the service starts or stops include a single digit event identifier.

<table>
<thead>
<tr>
<th>Event</th>
<th>Single digit identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success</td>
<td>0</td>
</tr>
<tr>
<td>Informational</td>
<td>1</td>
</tr>
<tr>
<td>Warning</td>
<td>2</td>
</tr>
<tr>
<td>Error</td>
<td>3</td>
</tr>
</tbody>
</table>
The event messages for EC2LaunchService.exe events begin with Service:. The event messages for EC2Launch.exe events do not begin with Service:.

Four digit event IDs include information about the stage, task, and severity of an event.

Topics
- Event ID format (p. 480)
- Event ID examples (p. 480)
- Windows event log schema (p. 481)

Event ID format

The following table shows the format of an EC2Launch v2 event identifier.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>21</td>
<td>0</td>
</tr>
<tr>
<td>S</td>
<td>T</td>
<td>L</td>
</tr>
</tbody>
</table>

The letters and numbers in the table represent the following event type and definitions.

<table>
<thead>
<tr>
<th>Event type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>S (Stage)</td>
<td>0 - Service-level message 1 - Boot 2 - Network 3 - PreReady 5 - Windows is Ready 6 - PostReady 7 - User Data</td>
</tr>
<tr>
<td>T (Task)</td>
<td>The tasks represented by the corresponding two values are different for each stage. To view the complete list of events, see Windows Event log schema (p. 481).</td>
</tr>
<tr>
<td>L (Level of the event)</td>
<td>0 - Success 1 - Informational 2 - Warning 3 - Error</td>
</tr>
</tbody>
</table>

Event ID examples

The following are example event IDs.

- 5000 - Windows is ready to use
- 3010 - Activate windows task in PreReady stage was successful
- 6013 - Set wallpaper task in PostReady Local Data stage encountered an error
## Windows event log schema

<table>
<thead>
<tr>
<th>MessageId/Event Id</th>
<th>Event message</th>
</tr>
</thead>
<tbody>
<tr>
<td>. . . 0</td>
<td>Success</td>
</tr>
<tr>
<td>. . . 1</td>
<td>Informational</td>
</tr>
<tr>
<td>. . . 2</td>
<td>Warning</td>
</tr>
<tr>
<td>. . . 3</td>
<td>Error</td>
</tr>
<tr>
<td>x</td>
<td>EC2Launch service-level logs</td>
</tr>
<tr>
<td>0</td>
<td>EC2Launch service exited successfully</td>
</tr>
<tr>
<td>1</td>
<td>EC2Launch service informational logs</td>
</tr>
<tr>
<td>2</td>
<td>EC2Launch service warning logs</td>
</tr>
<tr>
<td>3</td>
<td>EC2Launch service error logs</td>
</tr>
<tr>
<td>10</td>
<td>Replace state.json with previous-state.json</td>
</tr>
<tr>
<td>100</td>
<td>Serial Port</td>
</tr>
<tr>
<td>200</td>
<td>Sysprep</td>
</tr>
<tr>
<td>300</td>
<td>PrimaryNic</td>
</tr>
<tr>
<td>400</td>
<td>Metadata</td>
</tr>
<tr>
<td>x000</td>
<td>Stage (1 digit), Task (2 digits), Status (1 digit)</td>
</tr>
<tr>
<td>1000</td>
<td>Boot</td>
</tr>
<tr>
<td>1010</td>
<td>Boot - extend_root_partition</td>
</tr>
<tr>
<td>2000</td>
<td>Network</td>
</tr>
<tr>
<td>2010</td>
<td>Network - add_routes</td>
</tr>
<tr>
<td>3000</td>
<td>PreReady</td>
</tr>
<tr>
<td>3010</td>
<td>PreReady - activate_windows</td>
</tr>
<tr>
<td>3020</td>
<td>PreReady - install_egpu_manager</td>
</tr>
<tr>
<td>3030</td>
<td>PreReady - set_monitor_on</td>
</tr>
<tr>
<td>3040</td>
<td>PreReady - set_hibernation</td>
</tr>
<tr>
<td>3050</td>
<td>PreReady - set_admin_account</td>
</tr>
<tr>
<td>3060</td>
<td>PreReady - set_dns_suffix</td>
</tr>
<tr>
<td>3070</td>
<td>PreReady - set_wallpaper</td>
</tr>
<tr>
<td>3080</td>
<td>PreReady - set_update_schedule</td>
</tr>
<tr>
<td>3090</td>
<td>PreReady - output_log</td>
</tr>
<tr>
<td>MessageId/Event Id</td>
<td>Event message</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>3100</td>
<td>PreReady - enable_open_ssh</td>
</tr>
<tr>
<td>5000</td>
<td>Windows is Ready to use</td>
</tr>
<tr>
<td>6000</td>
<td>PostReadyLocalData</td>
</tr>
<tr>
<td>7000</td>
<td>PostReadyUserData</td>
</tr>
<tr>
<td>6010/7010</td>
<td>PostReadyLocal/UserData - set_wallpaper</td>
</tr>
<tr>
<td>6020/7020</td>
<td>PostReadyLocal/UserData - set_update_schedule</td>
</tr>
<tr>
<td>6030/7030</td>
<td>PostReadyLocal/UserData - set_hostname</td>
</tr>
<tr>
<td>6040/7040</td>
<td>PostReadyLocal/UserData - execute_program</td>
</tr>
<tr>
<td>6050/7050</td>
<td>PostReadyLocal/UserData - execute_script</td>
</tr>
<tr>
<td>6060/7060</td>
<td>PostReadyLocal/UserData - manage_package</td>
</tr>
<tr>
<td>6070/7070</td>
<td>PostReadyLocal/UserData - initialize_volume</td>
</tr>
<tr>
<td>6080/7080</td>
<td>PostReadyLocal/UserData - write_file</td>
</tr>
<tr>
<td>6090/7090</td>
<td>PostReadyLocal/UserData - start_ssm</td>
</tr>
<tr>
<td>7100</td>
<td>PostReadyUserData - enable_open_ssh</td>
</tr>
<tr>
<td>6110/7110</td>
<td>PostReadyLocal/UserData - enable_jumbo_frames</td>
</tr>
</tbody>
</table>

**EC2Launch v2 console log output**

This section contains sample console log output for EC2Launch v2 and lists all of the EC2Launch v2 console log error messages to help you to troubleshoot issues.

**Outputs**

- EC2Launch v2 console log output (p. 482)
- EC2Launch v2 console log messages (p. 483)

**EC2Launch v2 console log output**

The following is sample console log output for EC2Launch v2.

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Log Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020/08/13 17:25:12Z</td>
<td>Windows is being configured. SysprepState=IMAGE_STATE_UNDEPLOYABLE</td>
</tr>
<tr>
<td>2020/08/13 17:27:44Z</td>
<td>Windows is being configured. SysprepState=IMAGE_STATE_UNDEPLOYABLE</td>
</tr>
<tr>
<td>2020/08/13 17:28:02Z</td>
<td>Windows sysprep configuration complete.</td>
</tr>
<tr>
<td>2020/08/13 17:28:03Z</td>
<td>Message: Waiting for meta-data accessibility...</td>
</tr>
<tr>
<td>2020/08/13 17:28:03Z</td>
<td>Message: Meta-data is now available.</td>
</tr>
</tbody>
</table>
EC2Launch v2 console log messages

The following is a list of all of the EC2Launch v2 console log messages.

```
Message: Error EC2Launch service is stopping. {error message}
Error setting up EC2Launch agent folders
See instance logs for detail
Error stopping service
Error initializing service
Message: Windows sysprep configuration complete
Message: Invalid administrator username: {invalid username}
Message: Invalid administrator password
Username: {username}
Password: <Password>{encrypted password}</Password>
AMI Origin Version: {amiVersion}
AMI Origin Name: {amiName}
Microsoft Windows NT {currentVersion}.{currentBuildNumber}
OsVersion: {currentVersion}
OsProductName: {productName}
OsBuildLabEx: {buildLabEx}
OsCurrentBuild: {currentBuild}
OsReleaseId: {releaseId}
Language: {language}
TimeZone: {timeZone}
Offset: UTC {offset}
Launch agent: EC2Launch {BuildVersion}
AMI-ID: {amiId}
Instance-ID: {instanceId}
Instance Type: {instanceType}
RDPCERTIFICATE-SUBJECTNAME: {certificate subject name}
RDPCERTIFICATE-THUMBPRINT: {thumbprint hash}
SqlServerBilling: {sql billing}
SqlServerInstall: {sql patch leve, edition type}
Driver: AWS NVMe Driver {version}
Driver: Inbox NVMe Driver {version}
Driver: AWS PV Driver Package {version}
Microsoft-Hyper-V is {status}
Unable to get service status for vmms
Microsoft-Hyper-V is {status}
SSM: Amazon SSM Agent {version}
```
AWS VSS Version: {version}
Message: Windows sysprep configuration complete
Message: Windows is being configured. SysprepState is {state}
Windows is still being configured. SysprepState is {state}
Message: Windows is Ready to use
Message: Waiting for meta-data accessibility...
Message: Meta-data is now available.
Message: Still waiting for meta-data accessibility...
Message: Failed to find primary network interface...retrying...

EC2Launch v2 version histories

Version histories
- EC2Launch v2 version history (p. 484)
- EC2Launch v2 migration tool version history (p. 485)

EC2Launch v2 version history

The following table describes the released versions of EC2Launch v2.

<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
</table>
| 2.0.548 | • Adds leading zeros for hex IP hostname.  
• Fixes file permissions for enableOpenSsh task.  
• Fixes sysprep command crash. | August 4, 2021 |
| 2.0.470 | • Fixes bug in network stage to wait for DHCP to assign an IP to the instance.  
• Fixes bug with setDnsSuffix when SearchList registry key does not exist.  
• Fixes bug in DNS devolution logic in setDnsSuffix.  
• Adds network routes after intermediate reboots.  
• Allows initializeVolume to re-letter existing volumes.  
• Removes extra information from version subcommand. | July 20, 2021 |
| 2.0.285 | • Adds option to run user scripts in a detached process.  
• Legacy userdata now runs in a detached process, which is similar behavior to the prior launch agent.  
• Adds CLI flag to the sysprep and reset commands, which allows them to block until the service stops.  
• Restricts the config folder permissions. | March 8, 2021 |
| 2.0.207 | • Adds optional hostName field to setHostName task.  
• Fixes reboot bug. Reboot tasks executeScript and executeProgram will be marked as running.  
• Adds more return codes to the status command.  
• Adds bootstrap service to fix startup issue when running on t2.nano instance type.  
• Fixes clean installation mode to remove files not tracked by installer. | February 2, 2021 |
| 2.0.160 | • Fixes validate command to detect invalid stage name.  
• Adds w32tm resync command in addroutes task. | December 4, 2020 |
<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0.153</td>
<td>Adds Sysprep functionality in UserData.</td>
<td>November 3, 2020</td>
</tr>
<tr>
<td>2.0.146</td>
<td>• Fixes issue with RootExtend on non-English AMIs.</td>
<td>October 6, 2020</td>
</tr>
<tr>
<td></td>
<td>• Grants users group write permission to log files.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Creates MS Reserved partition for GPT volumes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Adds list-volumes command and volume dropdown in Amazon EC2Launch</td>
<td></td>
</tr>
<tr>
<td></td>
<td>settings.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Adds get-agent-config command for printing agent-config.yml file in</td>
<td></td>
</tr>
<tr>
<td></td>
<td>yaml or json format.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Erases static password if no public key detected.</td>
<td></td>
</tr>
<tr>
<td>2.0.124</td>
<td>• Adds option to display OS version on wallpaper.</td>
<td>September 10, 2020</td>
</tr>
<tr>
<td></td>
<td>• Initializes encrypted EBS volumes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Adds routes for VPCs with no local DNS name.</td>
<td></td>
</tr>
<tr>
<td>2.0.104</td>
<td>• Creates DNS suffix search list if it does not exist.</td>
<td>August 12, 2020</td>
</tr>
<tr>
<td></td>
<td>• Skips Hibernation if not requested.</td>
<td></td>
</tr>
<tr>
<td>2.0.0</td>
<td>Initial release.</td>
<td>June 30, 2020</td>
</tr>
</tbody>
</table>

**EC2Launch v2 migration tool version history**

The following table describes the released versions of the EC2Launch v2 migration tool.

<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.130</td>
<td>Increments the version number of the EC2Launch agent to 2.0.548.</td>
<td>August 5, 2021</td>
</tr>
<tr>
<td>1.0.113</td>
<td>Uses IMDSv2 in place of IMDSv1.</td>
<td>June 4, 2021</td>
</tr>
<tr>
<td>1.0.101</td>
<td>Increments the version number of the EC2Launch agent to 2.0.285.</td>
<td>March 12, 2021</td>
</tr>
<tr>
<td>1.0.86</td>
<td>Increments the version number of the EC2Launch agent to 2.0.207.</td>
<td>February 3, 2021</td>
</tr>
<tr>
<td>1.0.76</td>
<td>Increments the version number of the EC2Launch agent to 2.0.160.</td>
<td>December 4, 2020</td>
</tr>
<tr>
<td>1.0.69</td>
<td>Increments the version number of the EC2Launch agent to 2.0.153.</td>
<td>November 5, 2020</td>
</tr>
<tr>
<td>1.0.65</td>
<td>Increments the version number of the EC2Launch agent to 2.0.146.</td>
<td>October 9, 2020</td>
</tr>
<tr>
<td>1.0.60</td>
<td>Increments the version number of the EC2Launch agent to 2.0.124.</td>
<td>September 10, 2020</td>
</tr>
<tr>
<td>1.0.54</td>
<td>• Installs EC2Launch v2 if no agents are installed.</td>
<td>August 12, 2020</td>
</tr>
</tbody>
</table>
Configure a Windows instance using EC2Launch

EC2Launch is a set of Windows PowerShell scripts that replaced the EC2Config service on Windows Server 2016 and later AMIs. The latest launch service for all supported Windows Server versions is EC2Launch v2 (p. 449), which replaces both EC2Config and EC2Launch.

Contents
- EC2Launch tasks (p. 486)
- Install the latest version of EC2Launch (p. 487)
- Verify the EC2Launch version (p. 487)
- EC2Launch directory structure (p. 487)
- Configure EC2Launch (p. 488)
- EC2Launch version history (p. 491)

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 Work with instance user data (p. 594).
- Sets persistent static routes to reach the metadata service and AWS 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 Update metadata/KMS routes for Server 2016 and later when launching a custom AMI (p. 47).

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.

<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.50</td>
<td>Increments the version number of the EC2Launch agent to 2.0.104. Decouples the SSM agent.</td>
<td>August 10, 2020</td>
</tr>
<tr>
<td>1.0.0</td>
<td>Removes NuGet dependency.</td>
<td>June 30, 2020</td>
</tr>
</tbody>
</table>
• Send the *Windows is ready to use* message to the EC2 console.

For more information about Windows Server 2019, see Compare Features in Windows Server Versions on Microsoft.com.

**Install 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.

```powershell
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 and later 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.

All EC2Launch directories inherit their permissions from `C:\ProgramData`, with the exception of the following:

- `C:\ProgramData\Amazon\EC2-Windows\Launch\Module\Scripts` — This folder inherits all initial permissions from `C:\ProgramData` when it is created, but removes access for normal users to `CreateFiles` in the directory.
Configure 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. 488)
- Schedule EC2Launch to run on every boot (p. 489)
- Initialize drives and map drive letters (p. 489)
- Send Windows event logs to the EC2 console (p. 490)
- Send Windows is ready message after a successful boot (p. 490)

Configure initialization tasks

Specify settings in the LaunchConfig.json file to enable or disable the following initialization tasks:

- Set the computer name.
- Set the monitor to always stay on.
- 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.

```json
{
  "setComputerName": false,
  "setMonitorAlwaysOn": true,
  "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 happens the next time EC2Launch runs:

- If AdminPasswordType is still set to Random, EC2Launch will generate a new password at the next boot. After that boot, AdminPasswordType is automatically set to DoNothing to prevent EC2Launch from generating new passwords on subsequent boots. To prevent EC2Launch from generating a new password on the first boot, manually set AdminPasswordType to DoNothing before you reboot.
- 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.

**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 initializes drives that are not already 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:

```json
{
  "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:

```json
{
  "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.
**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 AWS Windows AMIs (p. 27).

To download and install the latest version of EC2Launch, see Install the latest version of EC2Launch (p. 487).

The following table describes the released versions of EC2Launch. Note that the version format changed after version 1.3.610.

<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.2003411</td>
<td>• Changed password generation logic to exclude passwords with low complexity.</td>
<td>04 August 2021</td>
</tr>
<tr>
<td>1.3.2003364</td>
<td>• Updated Install-EgpuManager with IMDSv2 support.</td>
<td>07 June 2021</td>
</tr>
<tr>
<td>1.3.2003312</td>
<td>• Added log lines before and after setMonitorAlwaysOn setting.</td>
<td>04 May 2021</td>
</tr>
<tr>
<td></td>
<td>• Added AWS Nitro Enclaves package version to console log.</td>
<td></td>
</tr>
<tr>
<td>1.3.2003284</td>
<td>Improved permission model by updating location for storing user data to LocalAppData.</td>
<td>23 March 2021</td>
</tr>
<tr>
<td>1.3.2003236</td>
<td>• Updated method for setting user password in Set-AdminAccount and Randomize-LocalAdminPassword.</td>
<td>11 February 2021</td>
</tr>
<tr>
<td></td>
<td>• Fixed InitializeDisks to check whether disk is set to read only before setting it to writable.</td>
<td></td>
</tr>
<tr>
<td>1.3.2003210</td>
<td>Localization fix for install.ps1.</td>
<td>07 January 2021</td>
</tr>
<tr>
<td>1.3.2003205</td>
<td>Security fix for install.ps1 to update permissions on %ProgramData%AmazonEC2-WindowsLaunchModuleScripts directory.</td>
<td>28 December 2020</td>
</tr>
<tr>
<td>1.3.2003189</td>
<td>Added w32tm resync after adding routes.</td>
<td>04 December 2020</td>
</tr>
<tr>
<td>1.3.2003155</td>
<td>Updated instance type information.</td>
<td>25 August 2020</td>
</tr>
<tr>
<td>1.3.2003150</td>
<td>Added OsCurrentBuild and OsReleaseId to console output.</td>
<td>22 April 2020</td>
</tr>
<tr>
<td>1.3.2003040</td>
<td>Fixed IMDS version 1 fallback logic.</td>
<td>07 April 2020</td>
</tr>
<tr>
<td>1.3.2002730</td>
<td>Added support for IMDS V2.</td>
<td>03 March 2020</td>
</tr>
<tr>
<td>1.3.2002240</td>
<td>Fixed minor issues.</td>
<td>31 October 2019</td>
</tr>
<tr>
<td>1.3.2001660</td>
<td>Fixed automatic login issue for users without password after first time executing Sysprep.</td>
<td>02 July 2019</td>
</tr>
<tr>
<td>1.3.2001360</td>
<td>Fixed minor issues.</td>
<td>27 March 2019</td>
</tr>
<tr>
<td>Version</td>
<td>Details</td>
<td>Release date</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>1.3.2001220</td>
<td>All PowerShell scripts signed.</td>
<td>28 February 2019</td>
</tr>
<tr>
<td>1.3.2001200</td>
<td>Fixed issue with InitializeDisks.ps1 where running the script on a node in a Microsoft Windows Server Failover Cluster would format drives on remote nodes whose drive letter matched the local drive letter.</td>
<td>27 February 2019</td>
</tr>
<tr>
<td>1.3.2001160</td>
<td>Fixed missing wallpaper in Windows 2019.</td>
<td>22 February 2019</td>
</tr>
</tbody>
</table>
| 1.3.2001040 | • Added plugin for setting the monitor to never turn off to fix ACPI issues.  
• 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. | 21 January 2019 |
| 1.3.2000930 | Fix for adding routes to metadata on ipv6-enabled ENIs.                                                                                                                                                 | 2 January 2019  |
| 1.3.2000760 | • Added default configuration for RSS and Receive Queue settings for ENA devices.  
• Disabled hibernation during Sysprep.                                                                                                                                                             | 5 December 2018 |
| 1.3.2000630 | • 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 Systems Manager.                                                                         | 9 November 2018 |
| 1.3.200039.0 | • Fixed improper drive lettering for EBS NVME volumes.  
• Added additional logging for NVME driver versions.                                                                                                                                                    | 15 August 2018  |
| 1.3.2000080 | Fixed minor issues.                                                                                                                                                                                   |                |
| 1.3.610     | Fixed issue with redirecting output and errors to files from user data.                                                                                                                                  |                |
| 1.3.590     | • Added missing instances types in the wallpaper.  
• Fixed an issue with drive letter mapping and disk installation.                                                                                                                                       |                |
| 1.3.580     | • 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 –NoShutdown option to enable Sysprep with no shutdown.                                                                                                                                       |                |
| 1.3.540     | Fixed minor issues.                                                                                                                                                                                   |                |
| 1.3.530     | Fixed minor issues.                                                                                                                                                                                   |                |
Configure a Windows instance using the EC2Config service

The latest launch service for all supported Windows Server versions is EC2Launch v2 (p. 449), which replaces both EC2Config and EC2Launch.

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 replaced EC2Config on Windows AMIs for Windows Server 2016 and later. For more information, see Configure a Windows instance using EC2Launch (p. 486). The latest launch service for all supported Windows Server versions is EC2Launch v2 (p. 449), which replaces both EC2Config and EC2Launch.

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. 494)
- Install the latest version of EC2Config (p. 494)
- Stop, restart, delete, or uninstall EC2Config (p. 496)
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 Work with instance user data (p. 594).

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 a single NIC or 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 standardized Amazon Machine Image (AMI) using Sysprep (p. 40).

Install 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.
EC2Launch replaces EC2Config on Windows Server 2016 and later AMIs. For more information, see Configure a Windows instance using EC2Launch (p. 486). The latest launch service for all supported Windows Server versions is EC2Launch v2 (p. 449), which replaces both EC2Config and EC2Launch.

For information about how to receive notifications for EC2Config updates, see Subscribe to EC2Config service notifications (p. 517). For information about the changes in each version, see the EC2Config version history (p. 506).

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 SSM Agent on the instance from the Microsoft Services snap-in.

   Note
   The updated EC2Config version information will not appear in the instance System Log or Trusted Advisor check until you reboot or stop and start your instance.
Stop, restart, delete, or uninstall 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 AWS Systems Manager, 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 Update SSM Agent by using Run Command 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 Create a custom Windows AMI (p. 37).

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 run it.

For more information, see User data execution (p. 574).

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.

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.

<table>
<thead>
<tr>
<th>Hostname</th>
<th>WIN-U0RFOJCTPUU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance ID</td>
<td>1-d583f76a</td>
</tr>
<tr>
<td>Public IP Address</td>
<td>54.208.43.227</td>
</tr>
<tr>
<td>Private IP Address</td>
<td>172.31.42.195</td>
</tr>
<tr>
<td>Availability Zone</td>
<td>us-east-1b</td>
</tr>
<tr>
<td>Instance Size</td>
<td>t2.micro</td>
</tr>
<tr>
<td>Architecture</td>
<td>AMD64</td>
</tr>
</tbody>
</table>

The information displayed on the desktop background is controlled by the settings file EC2ConfigService\Settings\WallpaperSettings.xml.

Enable Hibernation

Use this setting to allow EC2 to signal the operating system to perform hibernation.

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. AWS PV drivers and 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 standardized Amazon Machine Image (AMI) using Sysprep (p. 40).

**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 (AWS KMS).
- **AWS.EC2.Windows.CloudWatch.json**—Controls which performance counters to send to CloudWatch and which logs to send to CloudWatch Logs.
- **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 AWS KMS server.

- **SetAutodiscover**—Indicates whether to detect a AWS KMS automatically.
- **TargetKMSServer**—Stores the private IP address of a AWS KMS. The AWS KMS must be in the same Region as your instance.
- **DiscoverFromZone**—Discovers the AWS KMS server from the specified DNS zone.
- **ReadFromUserData**—Gets the AWS KMS server from UserData.
- **LegacySearchZones**—Discovers the AWS 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.

• Ec2OutputRDP Cert—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 AWS KMS client settings, and then activates Windows.

To modify the AWS 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 runs 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 (corresponds to the User Data check box on the Ec2 Service Properties dialog box).
• **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
<?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.

```xml
<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.

```xml
<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.

```xml
<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. 504)
• Configure proxy settings using the system.net element (p. 505)
• Configure proxy settings using Microsoft Group Policy and Microsoft Internet Explorer (p. 505)

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"
```
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.

```xml
<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 AWS Windows AMIs (p. 27).

You can receive notifications when new versions of the EC2Config service are released. For more information, see Subscribe to EC2Config service notifications (p. 517).

The following table describes the released versions of EC2Config. For information about the updates for SSM Agent, see Systems Manager SSM Agent Release Notes.

<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.9.4419</td>
<td>• Fixed IMDS version 1 fallback logic</td>
<td>2 June 2021</td>
</tr>
<tr>
<td></td>
<td>• Updated all usage of Windows temp directory to EC2Config temp directory</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• New version of SSM Agent 3.0.1124.0</td>
<td></td>
</tr>
<tr>
<td>4.9.4381</td>
<td>• Added support for SSM document schema version 2.2 in EC2ConfigUpdater</td>
<td>4 May 2021</td>
</tr>
<tr>
<td></td>
<td>• Added AWS Nitro Enclaves package version to console log</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• New version of SSM Agent 3.0.529.0</td>
<td></td>
</tr>
<tr>
<td>4.9.4326</td>
<td>• Removed all links in the settings UI</td>
<td>3 March 2021</td>
</tr>
<tr>
<td></td>
<td>• This is the last EC2Config version that supports Windows Server 2008.</td>
<td></td>
</tr>
<tr>
<td>4.9.4279</td>
<td>• Fixed security issue related to Ec2ConfigMonitor scheduled task</td>
<td>11 December 2020</td>
</tr>
<tr>
<td></td>
<td>• Fixed drive letter mapping issue and incorrect ephemeral disk count</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Added OsCurrentBuild and OsReleaseId to console output</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• New version of SSM Agent 2.3.871.0</td>
<td></td>
</tr>
<tr>
<td>4.9.4222</td>
<td>• Fixed IMDS version 1 fallback logic</td>
<td>7 April 2020</td>
</tr>
<tr>
<td></td>
<td>• New version of SSM Agent 2.3.842.0</td>
<td></td>
</tr>
<tr>
<td>4.9.4122</td>
<td>• Added support for IMDS v2</td>
<td>4 March 2020</td>
</tr>
<tr>
<td></td>
<td>• New version of SSM Agent 2.3.814.0</td>
<td></td>
</tr>
<tr>
<td>4.9.3865</td>
<td>• Fixed issue detecting COM port for Windows Server 2008 R2 on metal instances</td>
<td>31 October 2019</td>
</tr>
<tr>
<td></td>
<td>• New version of SSM Agent 2.3.722.0</td>
<td></td>
</tr>
<tr>
<td>4.9.3519</td>
<td>• New version of SSM Agent 2.3.634.0</td>
<td>18 June 2019</td>
</tr>
<tr>
<td>4.9.3429</td>
<td>• New version of SSM Agent 2.3.542.0</td>
<td>25 April 2019</td>
</tr>
<tr>
<td>4.9.3289</td>
<td>• New version of SSM Agent 2.3.444.0</td>
<td>11 February 2019</td>
</tr>
<tr>
<td>4.9.3270</td>
<td>• Added plugin for setting the monitor to never turn off to fix ACPI issues</td>
<td>22 January 2019</td>
</tr>
<tr>
<td></td>
<td>• SQL Server edition and version written to console</td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>Details</td>
<td>Release date</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 4.9.3230 | • Drive Letter Mapping description updated to better align to functionality  
• New version of SSM Agent 2.3.372.0 | 10 January 2019 |
| 4.9.3160 | • Increased wait time for primary NIC  
• Added default configuration for RSS and Receive Queue settings for ENA devices  
• Disabled hibernation during Sysprep  
• New version of SSM Agent 2.3.344.0  
• Upgraded AWS SDK to 3.3.29.13 | 15 December 2018 |
| 4.9.3067 | • Improvements made to instance hibernation  
• New version of SSM Agent 2.3.235.0 | 8 November 2018 |
| 4.9.3034 | • Added route 169.254.169.253/32 for DNS server  
• New version of SSM Agent 2.3.193.0 | 24 October 2018 |
| 4.9.2986 | • Added signing for all EC2Config related binaries  
• New version of SSM Agent 2.3.136.0 | 11 October 2018 |
| 4.9.2953 | New version of SSM Agent (2.3.117.0) | 2 October 2018 |
| 4.9.2926 | New version of SSM Agent (2.3.68.0) | 18 September 2018 |
| 4.9.2905 | • 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 | 17 September 2018 |
| 4.9.2854 | New version of SSM Agent (2.3.13.0) | 17 August 2018 |
| 4.9.2831 | New version of SSM Agent (2.2.916.0) | 7 August 2018 |
| 4.9.2818 | New version of SSM Agent (2.2.902.0) | 31 July 2018 |
| 4.9.2756 | New version of SSM Agent (2.2.800.0) | 27 June 2018 |
| 4.9.2688 | New version of SSM Agent (2.2.607.0) | 25 May 2018 |
| 4.9.2660 | New version of SSM Agent (2.2.546.0) | 11 May 2018 |
| 4.9.2644 | New version of SSM Agent (2.2.493.0) | 26 April 2018 |
| 4.9.2586 | New version of SSM Agent (2.2.392.0) | 28 March 2018 |
| 4.9.2565 | • 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 | 13 March 2018 |
<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.9.2549</td>
<td>New version of SSM Agent (2.2.325.0)</td>
<td>8 March 2018</td>
</tr>
<tr>
<td>4.9.2461</td>
<td>New version of SSM Agent (2.2.257.0)</td>
<td>15 February 2018</td>
</tr>
<tr>
<td>4.9.2439</td>
<td>New version of SSM Agent (2.2.191.0)</td>
<td>6 February 2018</td>
</tr>
<tr>
<td>4.9.2400</td>
<td>New version of SSM Agent (2.2.160.0)</td>
<td>16 January 2018</td>
</tr>
</tbody>
</table>
| 4.9.2327 | • 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 | 2 January 2018 |
| 4.9.2294 | New version of SSM Agent (2.2.103.0) | 4 December 2017 |
| 4.9.2262 | New version of SSM Agent (2.2.93.0) | 15 November 2017 |
| 4.9.2246 | New version of SSM Agent (2.2.82.0) | 11 November 2017 |
| 4.9.2218 | New version of SSM Agent (2.2.64.0) | 29 October 2017 |
| 4.9.2212 | New version of SSM Agent (2.2.58.0) | 23 October 2017 |
| 4.9.2203 | New version of SSM Agent (2.2.45.0) | 19 October 2017 |
| 4.9.2188 | New version of SSM Agent (2.2.30.0) | 10 October 2017 |
| 4.9.2180 | • New version of SSM Agent (2.2.24.0)  
• Added the Elastic GPU plugin for GPU instances | 5 October 2017 |
| 4.9.2143 | New version of SSM Agent (2.2.16.0) | 1 October 2017 |
| 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 | • New version of SSM Agent (2.0.879.0)  
• Fixed the CloudWatch Logs directory path for Windows Server 2003 |  |
<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
</table>
| 4.9.1981 | • New version of SSM Agent (2.0.847.0)  
• Fixed the issue with `important.txt` being generated in EBS volumes. |  |
| 4.9.1964 | New version of SSM Agent (2.0.842.0) |  |
| 4.9.1951 | • 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 | • 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 | • New version of SSM Agent (2.0.796.0)  
• Fixed an issue with output/error redirection for admin userdata execution. |  |
| 4.9.1863 | • 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 | • New version of SSM Agent (2.0.672.0)  
• Fixed agent update issue with v4.3, v4.4, and v4.5 |  |
<p>| 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) |  |</p>
<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
</tr>
</thead>
</table>
| 4.0.1343    | • Run Command, State Manager, 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. 496).  
• 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 and Configuring SSM Agent on Windows Instances in the AWS Systems Manager User Guide.  
• 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. |
| 3.19.1153   | • Re-enabled activation plugin for instances with old AWS 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   | • Fix to reliably add routes to the primary network adapter.  
• Updates to improve support for AWS services. |
| 3.17.1032   | • 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 Systems Manager Run Command output to S3 bucket names with '.' character.                                                                                                         |
| 3.14.786    | 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:  
<InitializeDrivesSettings><SettingsGroup>FormatWithoutTRIM</SettingsGroup></InitializeDrivesSettings |
| 3.13.727    | Systems Manager Run Command - Fixes to process commands reliably after windows reboot.                                                                                                                |
| 3.12.649    | • 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 Systems Manager Run Command. |
<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
</table>
| 3.11.521  | • Fixes to enable RDP thumbprint generation for Windows Server 2003.  
• Fixes to include timezone and UTC offset in the EC2Config log lines.  
• Systems Manager support to run Run Command commands in parallel.  
• Roll back previous change to bring partitioned disks online. |              |
| 3.10.442  | • Fix Systems Manager configuration failures when installing MSI applications.  
• Fix to reliably bring storage disks online.  
• Updates to improve support for AWS services.                                                                                                     |              |
| 3.9.359   | • 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   | • 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   | • Improvements to the ec2config-cli utility for config testing and troubleshooting within instance.  
• Avoid adding static routes for AWS KMS and meta-data 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   | • Windows activation reliability fix to first use link local address 169.254.0.250/251 for activating windows via AWS KMS  
• Improved proxy handling for Systems Manager, 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   | • 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                                                      |              |
<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
</table>
| 3.4.212 | - Fixes to EC2Config when used in combination with VM-Import.  
          - Fixed service naming issue in the WiX installer. | |
| 3.3.174 | - Improved exception handling for Systems Manager and domain join failures.  
          - Change to support Systems Manager 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  | - Performance improvements by delay loading Systems Manager SSM assemblies.  
          - Improved exception handling for malformed sysprep2008.xml.  
          - Command line support for Systems Manager “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  | - Enable support for Systems Manager.  
          - Automatically domain join EC2 Windows instances to an AWS directory via Systems Manager.  
          - Configure and upload CloudWatch logs/metrics via Systems Manager.  
          - Install PowerShell modules via Systems Manager.  
          - Install MSI applications via Systems Manager. | |
| 2.4.233 | - Added scheduled task to recover EC2Config from service startup failures.  
          - Improvements to the Console log error messages.  
          - Updates to improve support for AWS services. | |
| 2.3.313 | - 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  | - Handled NullPointerException while querying a registry key for determining Windows Sysprep state which returned null occasionally.  
          - Freed up unmanaged resources in finally block. | |
<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.11</td>
<td>Fixed a issue in CloudWatch plugin for handling empty log lines.</td>
<td></td>
</tr>
</tbody>
</table>
| 2.2.10  | • 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   | • 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   | • 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   | • 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 |               |
| 2.2.5   | • 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   | • Fixed a typo in PostSysprep.cmd  
          • Fixed the bug which EC2Config does not pin itself onto start menu for OS2012+ |               |
| 2.2.3   | • 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 RealTimeIsUniversal key, output incorrect settings of the RealTimeIsUniveral registry key to the Console  
          • Removed EC2Config dependency on Windows temp folder  
          • Removed UserData execution dependency on .Net 3.5 |               |
| 2.2.2   | • Added check to service stop behavior to check that resources are being released  
          • Fixed issue with long execution times when joined to domain |               |
<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
</table>
| 2.2.1   | • 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   | • Added support for .Net4 and .Net4.5 only environments  
         • Updated Installer |  |
| 2.1.19  | • Added ephemeral disk labeling support when using Intel network driver (eg. C3 instance Type). For more information, see Enhanced networking on Windows (p. 960).  
         • 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  | • 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  | • 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  | • 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):  
           • *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 |  |
### EC2Config service

<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
</table>
| 2.1.15  | • 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 Upgrade PV drivers on Windows instances (p. 524).  
• 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 runs immediately following 'Windows is Ready' | |
| 2.1.14  | Desktop wallpaper fixes | |
| 2.1.13  | • 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  | • 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  | • Desktop wallpaper overlay no longer persists with Sysprep or Shutdown without Sysprep  
• Userdata option to run 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 | |
| 2.1.7   | • 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 | |

515
<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
</tr>
</thead>
</table>
| 2.1.6   | • 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 sysprep.ini 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) |
| 2.1.2   | • 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 Sysprep.xml 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 InstallUpdates.ps1 to /Scripts folder for automating Patches and cleanup prior to Sysprep |
| 2.1.0   | • Desktop wallpaper displays instance information by default upon first logon (not disconnect/reconnect)  
         • PowerShell can be run from the userdata by surrounding the code with <powershell></powershell> |
Subscribe 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

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**.

Troubleshoot 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. Select the instance and choose **Instance state**, and then choose **Stop instance**.

**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 instances** 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 the volume** (p. 1204) and then **attach the volume** (p. 1186) 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. 1187).

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.


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**. Type **DefaultUserName** as the name and enter a password in 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. 425) 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.

<table>
<thead>
<tr>
<th></th>
<th>RedHat PV</th>
<th>Citrix PV</th>
<th>AWS PV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance type</td>
<td>Not supported for all instance types. If you specify an unsupported instance type, the instance is impaired.</td>
<td>Supported for Xen instance types.</td>
<td>Supported for Xen instance types.</td>
</tr>
<tr>
<td>Attached volumes</td>
<td>Supports up to 16 attached volumes.</td>
<td>Supports more than 16 attached volumes.</td>
<td>Supports more than 16 attached volumes.</td>
</tr>
<tr>
<td>Network</td>
<td>The driver has known issues where the network connection resets under high loads; for example, fast FTP file transfers.</td>
<td></td>
<td>The driver automatically configures jumbo frames on the network adapter when on a compatible instance type. When the instance is in a cluster placement group (p. 975), this offers better network performance between instances in the cluster placement group.</td>
</tr>
</tbody>
</table>

The following table shows which PV drivers you should run on each version of Windows Server on Amazon EC2.

<table>
<thead>
<tr>
<th>Windows Server version</th>
<th>PV driver version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2019</td>
<td>AWS PV latest version</td>
</tr>
<tr>
<td>Windows Server 2016</td>
<td>AWS PV latest version</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td>AWS PV latest version</td>
</tr>
<tr>
<td>Windows Server 2012</td>
<td>AWS PV latest version</td>
</tr>
</tbody>
</table>
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 drivers also have a Windows service named LiteAgent, which runs in user-mode. It handles tasks such as shutdown and restart events from AWS APIs on Xen generation instances. You can access and manage services by running `Services.msc` from the command line. When running on Nitro generation instances, the AWS PV drivers are not used and the LiteAgent service will self-stop starting with driver version 8.2.4. Updating to the latest AWS PV driver also updates the LiteAgent and improves reliability on all instance generations.

### Install 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 [Walkthrough: Automatically Update PV Drivers on EC2 Windows Instances (Console)] in the AWS Systems Manager User Guide.
- You can download the driver package and run the install program manually. Be sure to check the `readme.txt` file for system requirements. For information about downloading and installing the

---

<table>
<thead>
<tr>
<th>Windows Server version</th>
<th>PV driver version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2008 R2</td>
<td>AWS PV version 8.3.4 and earlier</td>
</tr>
<tr>
<td>Windows Server 2008</td>
<td>Citrix PV 5.9</td>
</tr>
<tr>
<td>Windows Server 2003</td>
<td>Citrix PV 5.9</td>
</tr>
</tbody>
</table>
AWS PV drivers, or upgrading a domain controller, see Upgrade Windows Server instances (AWS PV upgrade) (p. 525).

**AWS PV driver package history**

The following table shows the changes to AWS PV drivers for each driver release.

<table>
<thead>
<tr>
<th>Package version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.4.0</td>
<td>• Stability fixes to address rare cases of stuck disk IO.</td>
<td>2 March 2021</td>
</tr>
<tr>
<td></td>
<td>• Stability fixes to address rare cases of crashes during EBS volume</td>
<td></td>
</tr>
<tr>
<td></td>
<td>detachment.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Added feature to distribute load across multiple cores for workloads</td>
<td></td>
</tr>
<tr>
<td></td>
<td>that leverage more than 20,000 IOPS and experience degradation due to</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bottlenecks. To enable this feature, see Workloads that leverage more</td>
<td></td>
</tr>
<tr>
<td></td>
<td>than 20,000 disk IOPS experience degradation due to CPU bottlenecks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(p. 535).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• AWS PV 8.4 installation on Windows Server 2008 R2 will fail.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AWS PV version 8.3.4 and earlier are supported on Windows Server 2008</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R2.</td>
<td></td>
</tr>
<tr>
<td>8.3.4</td>
<td>Improved reliability of network device attachment.</td>
<td>4 August 2020</td>
</tr>
<tr>
<td>8.3.3</td>
<td>• Update to XenStore-facing component to prevent bug check during</td>
<td>4 February 2020</td>
</tr>
<tr>
<td></td>
<td>error-handling paths.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Update to storage component to avoid crashes when an invalid SRB is</td>
<td></td>
</tr>
<tr>
<td></td>
<td>submitted.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>To update this driver on Windows Server 2008 R2 instances, you must</td>
<td></td>
</tr>
<tr>
<td></td>
<td>first verify that the appropriate patches are installed to address the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>following Microsoft Security Advisory: Microsoft Security Advisory</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3033929.</td>
<td></td>
</tr>
<tr>
<td>8.3.2</td>
<td>Enhanced reliability of networking components.</td>
<td>30 July 2019</td>
</tr>
<tr>
<td>8.3.1</td>
<td>Improved performance and robustness of storage component.</td>
<td>12 June 2019</td>
</tr>
<tr>
<td>8.2.7</td>
<td>Improved efficiency to support migrating to latest generation instance</td>
<td>20 May 2019</td>
</tr>
<tr>
<td></td>
<td>types.</td>
<td></td>
</tr>
<tr>
<td>8.2.6</td>
<td>Improved efficiency of crash dump path.</td>
<td>15 January 2019</td>
</tr>
<tr>
<td>8.2.5</td>
<td>Additional security enhancements.</td>
<td>12 December 2018</td>
</tr>
<tr>
<td></td>
<td>PowerShell installer now available in package.</td>
<td></td>
</tr>
<tr>
<td>8.2.4</td>
<td>Reliability improvements.</td>
<td>2 October 2018</td>
</tr>
<tr>
<td>8.2.3</td>
<td>Bug fixes and performance improvements.</td>
<td>29 May 2018</td>
</tr>
<tr>
<td></td>
<td>Report EBS volume ID as disk serial number for EBS volumes. This</td>
<td></td>
</tr>
<tr>
<td></td>
<td>enables cluster scenarios such as S2D.</td>
<td></td>
</tr>
<tr>
<td>Package version</td>
<td>Details</td>
<td>Release date</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>8.2.1</td>
<td>Network and storage performance improvements plus multiple robustness fixes. To verify that this version has been installed, refer to the following Windows registry value: HKLM\Software\Amazon\PVDriver\Version 8.2.1.</td>
<td>8 March 2018</td>
</tr>
<tr>
<td>7.4.6</td>
<td>Stability fixes to make AWS PV drivers more resilient.</td>
<td>26 April 2017</td>
</tr>
<tr>
<td>7.4.3</td>
<td>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.</td>
<td>18 Nov 2016</td>
</tr>
<tr>
<td>7.4.2</td>
<td>Stability fixes for support of X1 instance type.</td>
<td>2 Aug 2016</td>
</tr>
<tr>
<td>7.4.1</td>
<td>• 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 bug check code 0x0000DEAD. • Stability fixes in AWS PV Network driver. • Added support for Windows Server 2008R2.</td>
<td>12 July 2016</td>
</tr>
<tr>
<td>7.3.2</td>
<td>• 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.</td>
<td>24 June 2015</td>
</tr>
<tr>
<td>7.3.1</td>
<td>TRIM update: Fix related to TRIM requests. This fix stabilizes instances and improves instance performance when managing large numbers of TRIM requests.</td>
<td></td>
</tr>
<tr>
<td>7.3.0</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>7.2.5</td>
<td>• 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.</td>
<td></td>
</tr>
<tr>
<td>7.2.4</td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>
### 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. 534).

### 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.

### Subscribe 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 from the console**

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**

2. In the navigation pane, choose **Subscriptions**.
3. Select the check box for the subscription and then choose **Actions**, **Delete subscriptions**. When prompted for confirmation, choose **Delete**.

**To subscribe to EC2 notifications using the AWS CLI**

To subscribe to EC2 notifications with the AWS CLI, use the following command.

```bash
aws sns subscribe --topic-arn arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers --protocol email --notification-endpoint YourUserName@YourDomainName.ext
```

**To subscribe to EC2 notifications using the AWS Tools for PowerShell**

To subscribe to EC2 notifications with Tools for Windows PowerShell, use the following command.

```powershell
Connect-SNSNotification -TopicArn 'arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers' -Protocol email -Region us-east-1 -Endpoint 'YourUserName@YourDomainName.ext'
```

**Upgrade PV drivers on Windows instances**

We recommend that you install the latest PV drivers to improve the stability and performance of your EC2 Windows instances. The directions on this page help you download the driver package and run the install program.

**To verify which driver your Windows instance uses**

Open **Network Connections** in Control Panel and view **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.

**System requirements**

Be sure to check the `readme.txt` file in the download for system requirements.

**Contents**

• Upgrade Windows Server instances (AWS PV upgrade) (p. 525)
• Upgrade a domain controller (AWS PV upgrade) (p. 526)
• Upgrade Windows Server 2008 and 2008 R2 instances (Redhat to Citrix PV upgrade) (p. 527)
• Upgrade your Citrix Xen guest agent service (p. 529)

**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, Windows Server 2016, or Windows Server 2019. 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. 526). The upgrade process for domain controller instances is different than standard editions of Windows.

**To upgrade AWS PV drivers**

1. We recommend that you create an AMI as a backup as follows, in case you need to roll back your changes.
   a. When you stop an instance, the data on any instance store volumes is erased. Before you stop an instance, verify that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.
   b. In the navigation pane, choose **Instances**.
   c. Select the instance that requires the driver upgrade, and choose **Instance state**, then **Stop instance**.
   d. After the instance is stopped, select the instance, choose **Actions**, then **Image and templates**, and then choose **Create image**.
   e. Choose **Instance state**, then **Start instance**.
2. Connect to the instance using Remote Desktop.
3. 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. 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.
4. **Download** the latest driver package to the instance.

Or, run the following PowerShell command:

```
PS C:\>invoke-webrequest https://s3.amazonaws.com/ec2-windows-drivers-downloads/AWSPV/Latest/AWSPVDriver.zip -outfile $env:USERPROFILE\pv_driver.zip
expand-archive $env:userprofile\pv_driver.zip -DestinationPath $env:userprofile\pv_drivers
```
5. Extract the contents of the folder and then run AWSPVDriverSetup.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, you can verify that the new driver was installed by connecting to the instance using Remote Desktop and then running the following PowerShell command:

```
Get-ItemProperty HKLM:\SOFTWARE\Amazon\PVDriver
```

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 package history (p. 521) 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. 534) 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. We recommend that you create a backup of your domain controller in case you need to roll back your changes. Using an AMI as a backup is not supported. For more information, see Backup and Restore Considerations for Virtualized Domain Controllers in the Microsoft documentation.

2. Run the following command to add DisableDCCheck to the registry:

```
526
```

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 are not detected. Domain controllers that use an NTDS folder on secondary drives will not boot because the secondary disk is not 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.

3. Run the following command to add DisableDCCCheck to the registry:
Download the latest driver package to the instance.

Extract the contents of the folder and then run AWSPVDriverSetup.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. Open Disk Management to review any offline secondary volumes and bring them online corresponding to the drive letters and folder mappings noted earlier.

You must connect to the instance by specifying the user name in the following format hostname\administrator. For example, Win2k12TestBox\administrator.

Run the following command to remove the DSRM boot configuration:

```
bcedit /deletevalue safeboot
```

Reboot the instance.

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 package history (p. 521).

Run the following command to delete DisabledDCCheck from the registry:

```
reg delete HKLM\SOFTWARE\Wow6432Node\Amazon\AWSPVDriverSetup /v DisabledDCCheck
```

Note

If you previously disabled TCP offloading (p. 534) 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 Install the latest version of EC2Config (p. 494).
- Verify that you have Windows PowerShell 3.0 installed. To verify the version that you have installed, run the following command in a PowerShell window:

```
PS C:\> $PSVersionTable.PSVersion
```

Windows PowerShell 3.0 is bundled in the Windows Management Framework (WMF) version 3.0 install package. If you need to install Windows PowerShell 3.0, see Windows Management Framework 3.0 in the Microsoft Download Center.

- Back up your important information on the instance, or create an AMI from the instance. For more information about creating an AMI, see Create a custom Windows AMI (p. 37). 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 Connect to your Windows instance (p. 413).
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, select Actions, then Monitor and troubleshoot, and then choose 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.

8. Connect to your instance and log in as the local administrator.
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.

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
For Windows Server 2008 R2 and later, we recommend you upgrade to AWS PV drivers that include the Guest Agent update.

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 Create a custom Windows AMI (p. 37). 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 Connect to your Windows instance (p. 413).
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.

Troubleshoot PV drivers

The following are solutions to issues that you might encounter with older Amazon EC2 images and PV drivers.

Contents
- Windows Server 2012 R2 loses network and storage connectivity after an instance reboot (p. 530)
- TCP offloading (p. 534)
- Time synchronization (p. 535)
- Workloads that leverage more than 20,000 disk IOPS experience degradation due to CPU bottlenecks (p. 535)

Windows Server 2012 R2 loses network and storage connectivity after an instance reboot

Important
This issue occurs only with AMIs made available before September 2014.

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 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 network device was removed, the script repairs the device, disables the Plug and Play Cleanup scan, and enables your instance to reboot with network connectivity enabled.

Contents
- Choose how to fix problems (p. 530)
- Method 1 - Enhanced networking (p. 531)
- Method 2 - Registry configuration (p. 532)
- Run the remediation script (p. 534)

Choose how to fix problems

There are two methods for restoring network and storage connectivity to an instance affected by this issue. Choose one of the following methods:
## Method 1 - Enhanced networking

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Instances**.
3. Locate the affected instance. Select the instance and choose **Instance state**, and then choose **Stop instance**.
   
   **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. Select the instance and choose **Actions**, then **Image and templates**, 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 **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.
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 package history](p. 521).
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. Select the instance, choose **Instance state**, and then choose **Stop instance**.

   **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.


   **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** (p. 1204) and then **attach the volume** (p. 1186) 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. 1187).
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:

    ```plaintext
    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 package history (p. 521).

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. Select the instance for which you want to run the remediation script. Choose Instance state, and then choose Stop instance.
   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. Select the instance, choose Actions, then Image and templates, and then choose Create image.
5. Choose Instance state, and then choose Start instance.
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

Important
This issue does not apply to instances running AWS PV or Intel network drivers.

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.

Warning
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.
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. 529).

### Workloads that leverage more than 20,000 disk IOPS experience degradation due to CPU bottlenecks

You can be affected by this issue if you are using Windows instances running AWS PV drivers that leverage more than 20,000 IOPS, and you experience bug check code `0x9E: USER_MODE_HEALTH_MONITOR`.

Disk reads and writes (IOs) in the AWS PV drivers occur in two phases: **IO preparation** and **IO completion**. By default, the preparation phase runs on a single arbitrary core. The completion phase runs on core 0. The amount of computation required to process an IO varies based on its size and other properties. Some IOs use more computation in the preparation phase, and others in the completion phase. When an instance drives more than 20,000 IOPS, the preparation or completion phase may result in a bottleneck, where the CPU upon which it runs is at 100% capacity. Whether or not the preparation or completion phase becomes a bottleneck depends on the properties of the IOs used by the application.
Starting with AWS PV drivers 8.4.0, the load of the preparation phase and the completion phase can be distributed across multiple cores, eliminating bottlenecks. Each application uses different IO properties. Therefore, applying one of the following configurations may raise, lower, or not impact the performance of your application. After you apply any of these configurations, monitor the application to verify that it is meeting your desired performance.

1. **Prerequisites**

   Before you begin this troubleshooting procedure, verify the following prerequisites:
   - Your instance uses AWS PV drivers version 8.4.0 or later. To upgrade, see Upgrade PV drivers on Windows instances (p. 524).
   - You have RDP access to the instance. For steps to connect to your Windows instance using RDP, see Connect to your Windows instance using RDP (p. 414).
   - You have administrator access on the instance.

2. **Observe CPU load on your instance**

   You can use Windows Task Manager to view the load on each CPU to determine potential bottlenecks to disk IO.
   1. Verify that your application is running and handling traffic similar to your production workload.
   2. Connect to your instance using RDP.
   3. Choose the Start menu on your instance.
   4. Enter Task Manager in the Start menu to open Task Manager.
   5. If Task Manager displays the Summary View, choose More details to expand the detailed view.
   6. Choose the Performance tab.
   7. Select CPU in the left pane.
   8. Right-click on the graph in the main pane and select Change graph to>Logical processors to display each individual core.
   9. Depending on how many cores are on your instance, you may see lines displaying CPU load over time, or you may just see a number.
      - If you see graphs displaying load over time, look for CPUs where the box is almost entirely shaded.
      - If you see a number on each core, look for cores that consistently show 95% or greater.
   10. Note whether core 0 or a different core is experiencing a heavy load.

3. **Choose which configuration to apply**

<table>
<thead>
<tr>
<th>Configuration name</th>
<th>When to apply this configuration</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default configuration</td>
<td>Workload is driving less than 20,000 IOPS, or other configurations did not improve performance or stability.</td>
<td>For this configuration, IO occurs on a few cores, which may benefit smaller workloads by increasing cache locality and reducing context switching.</td>
</tr>
<tr>
<td>Allow driver to choose whether to distribute completion</td>
<td>Workload is driving more than 20,000 IOPS and moderate or high load is observed on core 0.</td>
<td>This configuration is recommended for all Xen instances using PV 8.4.0 or later and leveraging more than 20,000 IOPS, whether or not problems are encountered.</td>
</tr>
</tbody>
</table>
### Configuration name | When to apply this configuration | Notes
--- | --- | ---
Distribute both preparation and completion | Workload is driving more than 20,000 IOPS, and either allowing the driver to choose the distribution did not improve performance, or a core other than 0 is experiencing a high load. | This configuration enables distribution of both IO preparation and IO completion.

**Note**

We recommend that you do not distribute IO preparation without also distributing IO completion (setting `DpcRedirection` without setting `NotifierDistributed`) because the completion phase is sensitive to overload by the preparation phase when the preparation phase is running in parallel.

**Registry key values**

- **NotifierDistributed**

  Value 0 or not present — The completion phase will run on core 0.

  Value 1 — The driver chooses to run the completion phase on core 0 or one additional core per attached disk.

  Value 2 — The driver runs the completion phase on one additional core per attached disk.

- **DpcRedirection**

  Value 0 or not present — The preparation phase will run on a single, arbitrary core.

  Value 1 — The preparation phase is distributed across multiple cores.

**Default configuration**

Apply the default configuration with AWS PV driver versions prior to 8.4.0, or if performance or stability degradation is observed after applying one of the other configurations in this section.

1. Connect to your instance using RDP.
2. Open a new PowerShell command prompt as an administrator.
3. Run the following commands to remove the `NotifierDistributed` and `DpcRedirection` registry keys.

   ```powershell
   Remove-ItemProperty -Path HKLM:\System\CurrentControlSet\Services\xenvbd\Parameters -Name NotifierDistributed
   Remove-ItemProperty -Path HKLM:\System\CurrentControlSet\Services\xenvbd\Parameters -Name DpcRedirection
   ```

4. Reboot your instance.
Allow driver to choose whether to distribute completion

Set NotifierDistributed registry key to allow the PV storage driver to choose whether or not to distribute IO completion.

1. Connect to your instance using RDP.
2. Open a new PowerShell command prompt as an administrator.
3. Run the following command to set the NotifierDistributed registry key.

```powershell
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd \Parameters -Value 0x00000001 -Name NotifierDistributed
```

4. Reboot your instance.

Distribute both preparation and completion

Set NotifierDistributed and DpcRedirection registry keys to always distribute both the preparation and completion phases.

1. Connect to your instance using RDP.
2. Open a new PowerShell command prompt as an administrator.
3. Run the following commands to set the NotifierDistributed and DpcRedirection registry keys.

```powershell
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd \Parameters -Value 0x00000002 -Name NotifierDistributed
```

```powershell
Set-ItemProperty -Type DWORD -Path HKLM:\System\CurrentControlSet\Services\xenvbd \Parameters -Value 0x00000001 -Name DpcRedirection
```

4. Reboot your instance.

AWS NVMe drivers for Windows instances

EBS volumes and instance store volumes are exposed as NVMe block devices on Nitro-based instances (p. 146). You must have the AWS NVMe driver installed in order to use an NVMe block device. The latest AWS Windows AMIs for Windows Server 2008 R2 and later contain the required AWS NVMe driver.

For more information about EBS and NVMe, see Amazon EBS and NVMe on Windows instances (p. 1343). For more information about SSD instance store and NVMe, see SSD instance store volumes (p. 1403).

Install or upgrade AWS NVMe drivers

If you are not using the latest AWS Windows AMIs provided by Amazon, use the following procedure to install the current AWS NVMe driver. You should perform this update at a time when it is convenient to reboot your instance. Either the install script will reboot your instance or you must reboot it as the final step.

**Prerequisites**

PowerShell 3.0 or later
To download and install the latest AWS NVMe driver

1. We recommend that you create an AMI as a backup as follows, in case you need to roll back your changes.
   a. When you stop an instance, the data on any instance store volumes is erased. Before you stop an instance, verify that you've copied any data that you need from your instance store volumes to persistent storage, such as Amazon EBS or Amazon S3.
   b. In the navigation pane, choose **Instances**.
   c. Select the instance that requires the driver upgrade, and choose **Instance state, Stop instance**.
   d. After the instance is stopped, select the instance, choose **Actions**, then **Image and templates**, and then choose **Create image**.
   e. Choose **Instance state, Start instance**.

2. Connect to your instance and log in as the local administrator.

3. Download and extract the drivers to your instance using one of the following options:
   - Using a browser:
     a. Download the latest driver package to the instance.
     b. Extract the zip archive.
   - Using PowerShell:

     ```powershell
     invoke-webrequest https://s3.amazonaws.com/ec2-windows-drivers-downloads/NVMe/Latest/AMSNVMe.zip -outfile $env:USERPROFILE\nvme_driver.zip
     expand-archive $env:userprofile\nvme_driver.zip -DestinationPath $env:userprofile\nvme_driver
     ```

4. Install the driver to your instance by running the `install.ps1` PowerShell script from the `nvme_driver` directory (`.\install.ps1`). If you get an error, make sure you are using PowerShell 3.0 or later.

5. If the installer does not reboot your instance, reboot the instance.

**AWS NVMe driver version history**

The following table describes the released versions of the AWS NVMe driver.

<table>
<thead>
<tr>
<th>Driver version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.2</td>
<td>Fixed issue with modifying EBS volumes actively processing IO, which may result in data corruption. Customers who do not modify online EBS volumes (for example, resizing or changing type) are not impacted.</td>
<td>10 September 2019</td>
</tr>
<tr>
<td>1.3.1</td>
<td>Reliability Improvements</td>
<td>21 May 2019</td>
</tr>
<tr>
<td>1.3.0</td>
<td>Device optimization improvements</td>
<td>31 August 2018</td>
</tr>
<tr>
<td>1.2.0</td>
<td>Performance and reliability improvements for AWS NVMe devices on all supported instances, including bare metal instances</td>
<td>13 June 2018</td>
</tr>
<tr>
<td>1.0.0</td>
<td>AWS NVMe driver for supported instance types running Windows Server</td>
<td>12 February 2018</td>
</tr>
</tbody>
</table>
Subscribe 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 from the console

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

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.

To subscribe to EC2 notifications using the AWS CLI

To subscribe to EC2 notifications with the AWS CLI, use the following command.

```
aws sns subscribe --topic-arn arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers --protocol email --notification-endpoint YourUserName@YourDomainName.ext
```

To subscribe to EC2 notifications using AWS Tools for Windows PowerShell

To subscribe to EC2 notifications with AWS Tools for Windows PowerShell, use the following command.

```
Connect-SNSNotification -TopicArn 'arn:aws:sns:us-east-1:801119661308:ec2-windows-drivers' -Protocol email -Region us-east-1 -Endpoint 'YourUserName@YourDomainName.ext'
```

Optimize CPU options

Amazon EC2 instances support multithreading, which enables multiple threads to run concurrently on a single CPU core. Each thread is represented as a virtual CPU (vCPU) on the instance. An instance has a default number of CPU cores, which varies according to instance type. For example, an m5.xlarge instance type has two CPU cores and two threads per core by default—four vCPUs in total.
Note
Each vCPU is a thread of a CPU core, except for T2 instances and instances powered by AWS Graviton2 processors.

In most cases, there is an Amazon EC2 instance type that has a combination of memory and number of vCPUs to suit your workloads. However, you can specify the following CPU options to optimize your instance for specific workloads or business needs:

- **Number of CPU cores**: You can customize the number of CPU cores for the instance. You might do this to potentially optimize the licensing costs of your software with an instance that has sufficient amounts of RAM for memory-intensive workloads but fewer CPU cores.

- **Threads per core**: You can disable multithreading by specifying a single thread per CPU core. You might do this for certain workloads, such as high performance computing (HPC) workloads.

You can specify these CPU options during instance launch. There is no additional or reduced charge for specifying CPU options. You’re charged the same as instances that are launched with default CPU options.

**Contents**
- Rules for specifying CPU options (p. 541)
- CPU cores and threads per CPU core per instance type (p. 541)
- Specify CPU options for your instance (p. 556)
- View the CPU options for your instance (p. 558)

**Rules for specifying CPU options**

To specify the CPU options for your instance, be aware of the following rules:

- CPU options can only be specified during instance launch and cannot be modified after launch.

- When you launch an instance, you must specify both the number of CPU cores and threads per core in the request. For example requests, see Specify CPU options for your instance (p. 556).

- 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. 541).

- To disable multithreading, specify one thread per core.

- When you change the instance type (p. 231) 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.

**Contents**
- Accelerated computing instances (p. 542)
- Compute optimized instances (p. 543)
- General purpose instances (p. 545)
- Memory optimized instances (p. 550)
- Storage optimized instances (p. 555)
## Accelerated computing instances

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Default CPU cores</th>
<th>Default threads per core</th>
<th>Valid CPU cores</th>
<th>Valid threads per core</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1 to 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>f1.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>1 to 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>f1.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>g3.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>1 to 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>g3.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>1 to 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>g3.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>g3s.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td>g4ad.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>g4ad.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>g4ad.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>g4ad.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 8, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>g4ad.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>2, 4, 8, 16, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>g4dn.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td>g4dn.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1 to 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>g4dn.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>1 to 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>g4dn.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>1 to 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>g4dn.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>g4dn.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>p2.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td>p2.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>1 to 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>p2.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
</tbody>
</table>
### Instance type | Default vCPUs | Default CPU cores | Default threads per core | Valid CPU cores | Valid threads per core
--- | --- | --- | --- | --- | ---
p3.2xlarge | 8 | 4 | 2 | 1 to 4 | 1, 2
p3.8xlarge | 32 | 16 | 2 | 1 to 16 | 1, 2
p3.16xlarge | 64 | 32 | 2 | 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32 | 1, 2
p3dn.24xlarge | 96 | 48 | 2 | 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48 | 1, 2

### Compute optimized instances

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Default CPU cores</th>
<th>Default threads per core</th>
<th>Valid CPU cores</th>
<th>Valid threads per core</th>
</tr>
</thead>
</table>
c4.large | 2 | 1 | 2 | 1 | 1, 2 |
c4.xlarge | 4 | 2 | 2 | 1, 2 | 1, 2 |
c4.2xlarge | 8 | 4 | 2 | 1 to 4 | 1, 2 |
c4.4xlarge | 16 | 8 | 2 | 1 to 8 | 1, 2 |
c4.8xlarge | 36 | 18 | 2 | 2, 4, 6, 8, 10, 12, 14, 16, 18 | 1, 2 |
c5.large | 2 | 1 | 2 | 1 | 1, 2 |
c5.xlarge | 4 | 2 | 2 | 2 | 1, 2 |
c5.2xlarge | 8 | 4 | 2 | 2, 4 | 1, 2 |
c5.4xlarge | 16 | 8 | 2 | 2, 4, 6, 8 | 1, 2 |
c5.9xlarge | 36 | 18 | 2 | 4, 6, 8, 10, 12, 14, 16, 18 | 1, 2 |
c5.12xlarge | 48 | 24 | 2 | 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24 | 1, 2 |
c5.18xlarge | 72 | 36 | 2 | 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36 | 1, 2 |
c5.24xlarge | 96 | 48 | 2 | 4, 6, 8, 10, 12, 14, 16, 18, 20 | 1, 2 |
<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Default CPU cores</th>
<th>Default threads per core</th>
<th>Valid CPU cores</th>
<th>Valid threads per core</th>
</tr>
</thead>
<tbody>
<tr>
<td>c5a.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5a.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5a.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1 to 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5a.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>1, 2, 3, 4, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5a.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>1, 2, 3, 4, 8, 12, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5a.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>1, 2, 3, 4, 8, 12, 16, 20, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5a.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5a.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5ad.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5ad.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5ad.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1 to 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5ad.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>1, 2, 3, 4, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5ad.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>1, 2, 3, 4, 8, 12, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5ad.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>1, 2, 3, 4, 8, 12, 16, 20, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5ad.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5ad.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>1, 2, 3, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5d.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5d.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5d.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
</tbody>
</table>
### Amazon Elastic Compute Cloud
User Guide for Windows Instances
Optimize CPU options

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Default CPU cores</th>
<th>Default threads per core</th>
<th>Valid CPU cores</th>
<th>Valid threads per core</th>
</tr>
</thead>
<tbody>
<tr>
<td>c5d.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5d.9xlarge</td>
<td>36</td>
<td>18</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5d.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5d.18xlarge</td>
<td>72</td>
<td>36</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5d.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5n.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5n.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5n.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5n.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5n.9xlarge</td>
<td>36</td>
<td>18</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18</td>
<td>1, 2</td>
</tr>
<tr>
<td>c5n.18xlarge</td>
<td>72</td>
<td>36</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36</td>
<td>1, 2</td>
</tr>
</tbody>
</table>

### General purpose instances

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Default CPU cores</th>
<th>Default threads per core</th>
<th>Valid CPU cores</th>
<th>Valid threads per core</th>
</tr>
</thead>
<tbody>
<tr>
<td>m4.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>m4.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td>m4.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1 to 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>m4.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>1 to 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>m4.10xlarge</td>
<td>40</td>
<td>20</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20</td>
<td>1, 2</td>
</tr>
<tr>
<td>Instance type</td>
<td>Default vCPUs</td>
<td>Default CPU cores</td>
<td>Default threads per core</td>
<td>Valid CPU cores</td>
<td>Valid threads per core</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>-------------------</td>
<td>-------------------------</td>
<td>-----------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>m4.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5a.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5a.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5a.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5a.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5a.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5a.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>6, 12, 18, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5a.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5a.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>12, 18, 24, 36, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5ad.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5ad.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>Instance type</td>
<td>Default vCPUs</td>
<td>Default CPU cores</td>
<td>Default threads per core</td>
<td>Valid CPU cores</td>
<td>Valid threads per core</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>-------------------</td>
<td>--------------------------</td>
<td>-----------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>m5ad.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5ad.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5ad.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5ad.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>6, 12, 18, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5ad.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5ad.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>12, 18, 24, 36, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5d.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5d.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5d.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5d.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5d.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5d.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5d.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5d.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5dn.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5dn.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5dn.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5dn.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5dn.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>Instance type</td>
<td>Default vCPUs</td>
<td>Default CPU cores</td>
<td>Default threads per core</td>
<td>Valid CPU cores</td>
<td>Valid threads per core</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>-------------------</td>
<td>--------------------------</td>
<td>-------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>m5dn.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5dn.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5dn.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5n.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5n.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5n.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5n.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5n.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5n.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5n.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5n.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5zn.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5zn.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5zn.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5zn.3xlarge</td>
<td>12</td>
<td>6</td>
<td>2</td>
<td>2, 4, 6</td>
<td>1, 2</td>
</tr>
<tr>
<td>m5zn.6xlarge</td>
<td>24</td>
<td>12</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12</td>
<td>1, 2</td>
</tr>
<tr>
<td>Instance type</td>
<td>Default vCPUs</td>
<td>Default CPU cores</td>
<td>Default threads per core</td>
<td>Valid CPU cores</td>
<td>Valid threads per core</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>-------------------</td>
<td>--------------------------</td>
<td>-----------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>m5zn.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>m6i.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>m6i.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td>m6i.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>m6i.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>m6i.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>m6i.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>m6i.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>m6i.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>m6i.32xlarge</td>
<td>128</td>
<td>64</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64</td>
<td>1, 2</td>
</tr>
<tr>
<td>t2.nano</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>t2.micro</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>t2.small</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>t2.medium</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
<td>1</td>
</tr>
<tr>
<td>t2.large</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
<td>1</td>
</tr>
<tr>
<td>t2.xlarge</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1 to 4</td>
<td>1</td>
</tr>
<tr>
<td>t2.2xlarge</td>
<td>8</td>
<td>8</td>
<td>1</td>
<td>1 to 8</td>
<td>1</td>
</tr>
<tr>
<td>t3.nano</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
</tbody>
</table>
### Optimize CPU options

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Default CPU cores</th>
<th>Default threads per core</th>
<th>Valid CPU cores</th>
<th>Valid threads per core</th>
</tr>
</thead>
<tbody>
<tr>
<td>t3.micro</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>t3.small</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>t3.medium</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>t3.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>t3.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>t3.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>t3a.nano</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>t3a.micro</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>t3a.small</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>t3a.medium</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>t3a.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>t3a.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>t3a.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
</tbody>
</table>

#### Memory optimized instances

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Default CPU cores</th>
<th>Default threads per core</th>
<th>Valid CPU cores</th>
<th>Valid threads per core</th>
</tr>
</thead>
<tbody>
<tr>
<td>r4.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>r4.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td>r4.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1 to 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>r4.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>1 to 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>r4.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>1 to 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>r4.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>Instance type</td>
<td>Default vCPUs</td>
<td>Default CPU cores</td>
<td>Default threads per core</td>
<td>Valid CPU cores</td>
<td>Valid threads per core</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------------</td>
<td>--------------------------</td>
<td>----------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>r5.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5a.1large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5a.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5a.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5a.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5a.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5a.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>6, 12, 18, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5a.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5a.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>12, 18, 24, 36, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5ad.1large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5ad.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5ad.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5ad.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5ad.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5ad.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>6, 12, 18, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>Instance type</td>
<td>Default vCPUs</td>
<td>Default CPU cores</td>
<td>Default threads per core</td>
<td>Valid CPU cores</td>
<td>Valid threads per core</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
<td>-------------------</td>
<td>--------------------------</td>
<td>----------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>r3ad.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>r3ad.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>12, 18, 24, 36, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5b.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5b.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5b.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5b.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5b.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5b.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5b.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5b.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5d.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5d.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5d.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5d.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5d.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5d.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5d.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
</tbody>
</table>
## Optimize CPU options

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Default CPU cores</th>
<th>Default threads per core</th>
<th>Valid CPU cores</th>
<th>Valid threads per core</th>
</tr>
</thead>
<tbody>
<tr>
<td>r5d.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5dn.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5dn.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5dn.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5dn.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5dn.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5dn.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5dn.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5dn.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5n.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5n.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5n.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5n.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5n.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5n.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>r5n.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>Instance type</td>
<td>Default vCPUs</td>
<td>Default CPU cores</td>
<td>Default threads per core</td>
<td>Valid CPU cores</td>
<td>Valid threads per core</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------</td>
<td>-------------------</td>
<td>--------------------------</td>
<td>-----------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>r5n.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48</td>
<td>1, 2</td>
</tr>
<tr>
<td>u-6tb1.56xlarge</td>
<td>224</td>
<td>224</td>
<td>1</td>
<td>1 to 224</td>
<td>1</td>
</tr>
<tr>
<td>u-6tb1.112xlarge</td>
<td>448</td>
<td>224</td>
<td>2</td>
<td>1 to 224</td>
<td>1, 2</td>
</tr>
<tr>
<td>u-9tb1.112xlarge</td>
<td>448</td>
<td>224</td>
<td>2</td>
<td>1 to 224</td>
<td>1, 2</td>
</tr>
<tr>
<td>u-12tb1.112xlarge</td>
<td>448</td>
<td>224</td>
<td>2</td>
<td>1 to 224</td>
<td>1, 2</td>
</tr>
<tr>
<td>x1.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>x1.32xlarge</td>
<td>128</td>
<td>64</td>
<td>2</td>
<td>4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64</td>
<td>1, 2</td>
</tr>
<tr>
<td>x1e.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td>x1e.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1 to 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>x1e.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>1 to 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>x1e.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>1 to 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>x1e.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>x1e.32xlarge</td>
<td>128</td>
<td>64</td>
<td>2</td>
<td>4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64</td>
<td>1, 2</td>
</tr>
<tr>
<td>z1d.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>z1d.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>z1d.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>z1d.3xlarge</td>
<td>12</td>
<td>6</td>
<td>2</td>
<td>2, 4, 6</td>
<td>1, 2</td>
</tr>
<tr>
<td>z1d.6xlarge</td>
<td>24</td>
<td>12</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12</td>
<td>1, 2</td>
</tr>
</tbody>
</table>
## Optimize CPU options

### Storage optimized instances

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Default CPU cores</th>
<th>Default threads per core</th>
<th>Valid CPU cores</th>
<th>Valid threads per core</th>
</tr>
</thead>
<tbody>
<tr>
<td>d2.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td>d2.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1 to 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>d2.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>1 to 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>d2.8xlarge</td>
<td>36</td>
<td>18</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18</td>
<td>1, 2</td>
</tr>
<tr>
<td>d3.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td>d3.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>d3.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>d3.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18</td>
<td>1, 2</td>
</tr>
<tr>
<td>d3en.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>d3en.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td>d3en.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>d3en.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>2, 4, 6, 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>d3en.6xlarge</td>
<td>24</td>
<td>12</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12</td>
<td>1, 2</td>
</tr>
<tr>
<td>d3en.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>d3en.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>h1.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1 to 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>h1.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>1 to 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>h1.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>1 to 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>h1.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
</tbody>
</table>
Specify 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. 550):

- 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

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Default vCPUs</th>
<th>Default CPU cores</th>
<th>Default threads per core</th>
<th>Valid CPU cores</th>
<th>Valid threads per core</th>
</tr>
</thead>
<tbody>
<tr>
<td>i3.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>i3.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
<td>1, 2</td>
</tr>
<tr>
<td>i3.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1 to 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>i3.4xlarge</td>
<td>16</td>
<td>8</td>
<td>2</td>
<td>1 to 8</td>
<td>1, 2</td>
</tr>
<tr>
<td>i3.8xlarge</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>1 to 16</td>
<td>1, 2</td>
</tr>
<tr>
<td>i3.16xlarge</td>
<td>64</td>
<td>32</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32</td>
<td>1, 2</td>
</tr>
<tr>
<td>i3en.large</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1, 2</td>
</tr>
<tr>
<td>i3en.xlarge</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1, 2</td>
</tr>
<tr>
<td>i3en.2xlarge</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2, 4</td>
<td>1, 2</td>
</tr>
<tr>
<td>i3en.3xlarge</td>
<td>12</td>
<td>6</td>
<td>2</td>
<td>2, 4, 6</td>
<td>1, 2</td>
</tr>
<tr>
<td>i3en.6xlarge</td>
<td>24</td>
<td>12</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12</td>
<td>1, 2</td>
</tr>
<tr>
<td>i3en.12xlarge</td>
<td>48</td>
<td>24</td>
<td>2</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24</td>
<td>1, 2</td>
</tr>
<tr>
<td>i3en.24xlarge</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48</td>
<td>1, 2</td>
</tr>
</tbody>
</table>
Disable multithreading

To disable multithreading, specify one thread per core.

To disable multithreading during instance launch (console)

1. Follow the Launch an instance using the Launch Instance Wizard (p. 392) procedure.
2. On the Configure Instance Details page, for CPU options, choose Specify CPU options.
3. For Core count, choose the number of required CPU cores. In this example, to specify the default CPU core count for an r4.4xlarge instance, choose 8.
4. To disable multithreading, for Threads per core, choose 1.
5. Continue as prompted by the wizard. When you've finished reviewing your options on the Review Instance Launch page, choose Launch. For more information, see Launch an instance using the Launch Instance Wizard (p. 392).

To disable multithreading during instance launch (AWS CLI)

Use the run-instances AWS CLI command and specify a value of 1 for ThreadsPerCore for the --cpu-options parameter. For CoreCount, specify the number of CPU cores. In this example, to specify the default CPU core count for an r4.4xlarge instance, specify a value of 8.

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options "CoreCount=8,ThreadsPerCore=1" --key-name MyKeyPair
```

Specify 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 Launch an instance using the Launch Instance Wizard (p. 392) 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 Launch an instance using the Launch Instance Wizard (p. 392).

To specify a custom number of vCPUs during instance launch (AWS CLI)

The following example launches an r4.4xlarge instance with six vCPUs.

Use the run-instances AWS CLI command and specify the number of CPU cores and number of threads in the --cpu-options parameter. You can specify three CPU cores and two threads per core to get six vCPUs.

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options "CoreCount=3,ThreadsPerCore=2" --key-name MyKeyPair
```
Alternatively, specify six CPU cores and one thread per core (disable multithreading) to get six vCPUs:

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options "CoreCount=6,ThreadsPerCore=1" --key-name MyKeyPair
```

### View the CPU options for your instance

You can view the CPU options for an existing instance in the Amazon EC2 console or by describing the instance using the AWS CLI.

**New console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Instances** and select the instance.
3. On the **Details** tab, under **Host and placement group**, find **Number of vCPUs**.
4. To view core count and threads per core, choose the value for **Number of vCPUs**.

**Old console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Instances** and select the instance.
3. Choose **Description** and find **Number of vCPUs**.
4. To view core count and threads per core, choose the value for **Number of vCPUs**.

**To view the CPU options for an instance (AWS CLI)**

Use the `describe-instances` command.

```
aws ec2 describe-instances --instance-ids i-123456789abcdef123
```

```
...
"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
```
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.

## Set 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 the date and time of your instance are not set correctly, the date in the signature may not match the date of the request, and AWS rejects the request.

Amazon provides the Amazon Time Sync Service, which is accessible from all EC2 instances, and is also used by other AWS services. This service uses a fleet of satellite-connected and atomic reference clocks in each Region to deliver accurate current time readings of the Coordinated Universal Time (UTC) global standard through Network Time Protocol (NTP). The Amazon Time Sync Service automatically smooths any leap seconds that are added to UTC.

The Amazon Time Sync Service is available through NTP at the 169.254.169.123 IPv4 address or the fd00:ec2::123 IPv6 address for any instance running in a VPC. Your instance does not require access to the internet, and you do not have to configure your security group rules or your network ACL rules to allow access. The latest versions of AWS Windows AMIs synchronize with the Amazon Time Sync Service by default.

**Note**

The examples in this section use the IPv4 address of the Amazon Time Sync Service: 169.254.169.253. If you are retrieving time for EC2 instances over the IPv6 address, ensure that you use the IPv6 address instead: fd00:ec2::253. The IPv6 address is only accessible on Instances built on the Nitro System (p. 146).

**Should I use UTC for my instances?**

We recommend that you use Coordinated Universal Time (UTC) for your instances to avoid human error and to facilitate synchronization across your CloudWatch Logs, Metrics, local logs, and other services. You can, however, choose to use a different time zone to better suit your requirements.

When you use local timezones rather than UTC, make sure that you account for aspects such as daylight savings time (when applicable) for automation, code, scheduled jobs, troubleshooting activities (correlating logs), and more.

Use the following procedures to configure the Amazon Time Sync Service on your instance from the command prompt. Alternatively, you can use external NTP sources. For more information about NTP and public time sources, see [http://www.ntp.org/](http://www.ntp.org/). An instance must have access to the internet for the external NTP time sources to work.
Change 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.

Configure network time protocol (NTP)

Amazon provides the Amazon Time Sync Service, which is accessible from all EC2 instances, and is also used by other AWS services. 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 IPv4 address or the fd00:ec2::123 IPv6 address for any instance running in a VPC, and your instance does not require internet access to use it. Starting with the August 2018 release, Windows AMIs use the Amazon Time Sync Service by default.

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:
**Set the time**

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

2. 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 required. 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. If your instance is within a domain, you should change the settings to use the domain controllers as the time source to avoid time skew. The security group of your instance must be configured to allow outbound UDP traffic on port 123 (NTP).

**To change the NTP servers**

1. From the Command Prompt window, run the following command:

   w32tm /config /manualpeerlist:"NTP servers" /syncfromflags:manual /update

   Where NTP servers is a space-delimited list of NTP servers for the instance to use.

2. 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 virtual 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 Amazon Time Sync 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 acts as a standalone NTP client instead of 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.

<table>
<thead>
<tr>
<th>Registry Path</th>
<th>Key Name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKLM:\System \CurrentControlSet\services \w32time\Config</td>
<td>UpdateInterval</td>
<td>120</td>
</tr>
<tr>
<td>HKLM:\System \CurrentControlSet\services \w32time\Parameters</td>
<td>NtpServer</td>
<td>169.254.169.123,0x9</td>
</tr>
<tr>
<td>HKLM:\System \CurrentControlSet\services \w32time\Parameters</td>
<td>Type</td>
<td>NTP</td>
</tr>
<tr>
<td>HKLM:\System \CurrentControlSet\services \w32time\TimeProviders \NtpClient</td>
<td>Enabled</td>
<td>1</td>
</tr>
<tr>
<td>HKLM:\System \CurrentControlSet\services \w32time\TimeProviders \NtpClient</td>
<td>InputProvider</td>
<td>1</td>
</tr>
<tr>
<td>HKLM:\System \CurrentControlSet\services \w32time\TimeProviders \NtpClient</td>
<td>SpecialPollInterval</td>
<td>900</td>
</tr>
</tbody>
</table>

**Configure 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 `RealTimeIsUniversal` registry key.

This key is set by default on all current generation instances. To verify whether the `RealTimeIsUniversal` registry key is set, see Step 4 in the following procedure. If the key is not set, follow the these steps from the beginning.

**To set the RealTimeIsUniversal registry key**

1. From the instance, open a Command Prompt window.
2. 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
   ```

3. If you are using a Windows Server 2008 AMI (not Windows Server 2008 R2) that was created before February 22, 2013, 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.
4. (Optional) Verify that the instance saved the key successfully using the following command:
Set the password

This command returns the subkeys for the **TimeZoneInformation** registry key. You should see the **RealTimeIsUniversal** 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    00000300020002000000000000000000
   StandardBias           REG_DWORD     0x0
   StandardName           REG_SZ        \@tzres.dll,-212
   StandardStart          REG_BINARY    00000B00010002000000000000000000
   TimeZoneKeyName        REG_SZ        Pacific Standard Time
   DynamicDaylightTimeDisabled REG_DWORD     0x0
   ActiveTimeBias         REG_DWORD     0x1a4
   RealTimeIsUniversal    REG_DWORD     0x1
```

**Related resources**

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](https://docs.microsoft.com/en-us/windows/win32/utime/how-the-windows-time-service-works) (Microsoft)
- [W32tm](https://docs.microsoft.com/en-us/windows/win32/utime/w32tm) (Microsoft)
- [How the Windows Time service treats a leap second](https://docs.microsoft.com/en-us/windows/win32/utime/how-the-windows-time-service-treats-a-leap-second) (Microsoft)

**Set the password for a Windows instance**

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** generates the default password. With AWS Windows AMIs for Windows Server 2016 and later, **EC2Launch** generates the default password.

**Note**

With Windows Server 2016 and later, **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.

**Change 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, enclose the password in double quotes.

   ```
   net user Administrator "new_password"
   ```
3. Store the new password in a safe place.

Change a lost or expired password

If you lose your password or it expires, you can generate a new password. For password reset procedures, see Reset a lost or expired Windows administrator password (p. 1484).

Add 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.

Add 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 2019 English Installation Media if your instance is running Windows Server 2019.
8. Choose Actions, Create Volume.
9. For Availability Zone, select the Availability Zone that matches your Windows instance. Choose Add Tag and specify Name for the tag key and a descriptive name for the tag value. Choose Create Volume.
10. In the Volume Successfully Created message, choose the volume that you just created.
11. Choose Actions, Attach Volume.
12. 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 Make an Amazon EBS volume available for use on Windows (p. 1187).

**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.

15. (Optional) When you are finished with the installation media, you can detach the volume. After you detach the volume, you can delete it. For more information, see Detach an Amazon EBS volume from a Windows instance (p. 1204) and Delete an Amazon EBS volume (p. 1206).

### Add 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.

   ```powershell
   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:

   ```json
   ...  
   Description         : Windows 2019 English Installation Media
   Encrypted           : False
   KmsKeyId            : 
   OwnerAlias          : amazon
   OwnerId             : 123456789012
   Progress            : 100%
   SnapshotId          : snap-22da283e
   StartTime           : 10/25/2019 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.

   ```powershell
   PS C:\> New-EC2Volume -AvailabilityZone us-east-1a -VolumeType gp2 -SnapshotId snap-22da283e
   ```

4. In the output, note the volume ID.
Add Windows components

5. Use the **Add-EC2Volume** cmdlet to attach the volume to your instance.

```
PS C:\> Add-EC2Volume -InstanceId i-08771ddaf98f9489 -VolumeId vol-06aa9e1fbf8b82ed1 -Device xvdh
```

6. Connect to your instance and make the volume available. For more information, see Make an Amazon EBS volume available for use on Windows (p. 1187).

   **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.

8. (Optional) When you are finished with the installation media, use the **Dismount-EC2Volume** cmdlet to detach the volume from your instance. After you detach the volume, you can use the **Remove-EC2Volume** cmdlet to delete the volume.

---

**Add 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": [
   ...]
   }
   ```

3. Use the **create-volume** command to create a volume from the snapshot. Specify the same Availability Zone as your instance.
Configure a secondary private IPv4 Address

4. In the output, note the volume ID.

```json
{
    "AvailabilityZone": "us-east-1a",
    "Encrypted": false,
    "VolumeType": "gp2",
    "VolumeId": "vol-0c98b37f30bcbc290",
    "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.

```bash
aws ec2 attach-volume --volume-id vol-0c98b37f30bcbc290 --instance-id i-01474ef662b89480 --device xvdg
```

6. Connect to your instance and make the volume available. For more information, see Make an Amazon EBS volume available for use on Windows (p. 1187).

   **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.

8. (Optional) When you are finished with the installation media, use the `detach-volume` command to detach the volume from your instance. After you detach the volume, you can use the `delete-volume` command to delete the volume.

## Configure 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:

**Topics**
- Prerequisite steps (p. 568)
- Step 1: Configure static IP addressing on your instance (p. 568)
- Step 2: Configure a secondary private IP address for your instance (p. 570)
- Step 3: Configure applications to Use the secondary private IP address (p. 571)

**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.

**Before you begin**
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.

Prerequisite steps

1. Assign the secondary private IPv4 address to the network interface for the instance. You can assign the secondary private IPv4 address when you launch the instance, or after the instance is running. For more information, see Assign a secondary private IPv4 address (p. 902).

2. Allocate an Elastic IP address and associate it with the secondary private IPv4 address. For more information, see Allocate an Elastic IP address (p. 927) and Associate an Elastic IP address with the secondary private IPv4 address (p. 904).

Step 1: Configure static IP addressing on your 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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP address</td>
<td>The IPv4 address obtained in step 2 above.</td>
</tr>
<tr>
<td>Subnet mask</td>
<td>The subnet mask obtained in step 2 above.</td>
</tr>
<tr>
<td>Default gateway</td>
<td>The default gateway address obtained in step 2 above.</td>
</tr>
<tr>
<td>Preferred DNS server</td>
<td>The DNS server obtained in step 2 above.</td>
</tr>
<tr>
<td>Alternate DNS server</td>
<td>The alternate DNS server obtained in step 2 above. If an alternate DNS server was not listed, leave this field blank.</td>
</tr>
</tbody>
</table>

**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 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**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Instances** and select your instance.
3. On the **Networking**, 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 instance (p. 568)**, and then choose **Add**.

<table>
<thead>
<tr>
<th>TCP/IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP address: 10.0.0.14</td>
</tr>
<tr>
<td>Subnet mask: 255.255.255.0</td>
</tr>
</tbody>
</table>

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.)
Run commands on your Windows instance at launch

When you launch a Windows instance in Amazon EC2, you can pass user data to the instance that can be used to perform automated configuration tasks or to run scripts after the instance starts. Instance user data is treated as opaque data; it is up to the instance to interpret it. User data is processed by EC2Launch v2 (supported preview AMIs and by download (p. 452)), EC2Launch (p. 486) on Windows Server 2016 and later, and EC2Config (p. 493) on Windows Server 2012 R2 and earlier.

For examples of the assembly of a UserData property in a AWS CloudFormation template, see Base64 Encoded UserData Property and Base64 Encoded UserData Property with AccessKey and SecretKey.

For information about running commands on your Linux instance at launch, see Running commands on your Linux instance at launch in the Amazon EC2 User Guide for Linux Instances.

Contents
- User data scripts (p. 572)
- User data execution (p. 574)
- User data and the console (p. 576)
- User data and the Tools for Windows PowerShell (p. 577)

User data scripts

For EC2Config or EC2Launch to run scripts, you must enclose the script within a special tag when you add it to user data. The tag that you use depends on whether the commands run in a Command Prompt window (batch commands) or use 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.

If you use an AWS API, including the AWS CLI, in a user data script, you must use an instance profile when launching the instance. An instance profile provides the appropriate AWS credentials required by the user data script to make the API call. For more information, see instance profiles (p. 1115). The permissions you assign to the IAM role depend on which services you are calling with the API. For more information, see IAM roles for Amazon EC2.

Script type
- Syntax for batch scripts (p. 573)
- Syntax for Windows PowerShell scripts (p. 573)
- Syntax for YAML configuration scripts (p. 573)
• Base64 encoding (p. 574)

Syntax for batch scripts

Specify a batch script using the `script` tag. Separate the commands using line breaks. For example:

```xml
<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 run one time when you launch the instance. To run the user data scripts every time you reboot or start the instance, add `<persist>true</persist>` to the user data.

```xml
<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-yyyy-hh-mm")
New-Item $file -ItemType file
</powershell>
```

By default, the user data scripts are run one time when you launch the instance. To run 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-yyyy-hh-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

Syntax for YAML configuration scripts

If you are using EC2Launch v2 to run scripts, you can use the YAML format. To view configuration tasks, details, and examples for EC2Launch v2, see EC2Launch v2 task configuration (p. 468).

Specify a YAML script with the `executeScript` task.

*Example YAML syntax to run a PowerShell script*

```yaml
version: 1.0
tasks:
  - task: executeScript
    inputs:
      - frequency: always
    type: powershell
```
Run commands at launch

Example YAML syntax to run a batch script

```yaml
version: 1.0
tasks:
  - task: executeScript
    inputs:
      - frequency: always
        type: batch
        runAs: localSystem
        content: |
          echo Current date and time >> %SystemRoot%\Temp\test.log
          echo %DATE% %TIME% >> %SystemRoot%\Temp\test.log
```

**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 run. The following is an example that encodes using Windows PowerShell.

```powershell
$UserData = [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Script))
```

The following is an example that decodes using PowerShell.

```powershell
```

For more information about base64 encoding, see [https://www.ietf.org/rfc/rfc4648.txt](https://www.ietf.org/rfc/rfc4648.txt).

**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 run the next time the instance reboots or restarts. Alternatively, you can specify that user data scripts are run every time the instance reboots or restarts.

User data scripts are run from the local administrator account when a random password is generated. Otherwise, user data scripts are run from the System account.

**Instance launch**

Scripts in the instance user data are run 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 v2, EC2Launch, and EC2Config contain the output from the standard output and standard error streams.

**EC2Launch v2**

The log file for EC2Launch v2 is `C:\ProgramData\Amazon\EC2Launch\log\agent.log`.

**Note**

The `C:\ProgramData` folder might be hidden. To view the folder, you must show hidden files and folders.

The following information is logged when the user data is run:

- **Info**: Converting user-data to yaml format – If the user data was provided in XML format
Run commands at launch

- Info: Initializing user-data state – The start of user data execution
- Info: Frequency is: always – If the user data task is running on every boot
- Info: Frequency is: once – If the user data task is running just once
- Stage: postReadyUserData execution completed – The end of user data execution

EC2Launch

The log file for EC2Launch is C:\ProgramData\Amazon\EC2-Windows\Launch\Log \UserdataExecution.log.

The C:\ProgramData folder might be hidden. To view the folder, you must show hidden files and folders.

The following information is logged when the user data is run:

- 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 run, their output is logged

EC2Config

The log file for EC2Config is C:\Program Files\Amazon\Ec2ConfigService\Logs \Ec2Config.log. The following information is logged when the user data is run:

- 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 run, their output is logged

Subsequent reboots or starts

When you update instance user data, user data scripts are not run automatically when you reboot or start the instance. However, you can enable user data execution so that user data scripts are run 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 run the next time the instance starts or reboots, even if you did not enable user data execution for subsequent reboots or starts. The user data scripts will not be executed on subsequent reboots or starts.

To enable user data execution with EC2Launch v2 (Preview AMIs)

- To run a task in user data on first boot, set frequency to once.
- To run a task in user data on every boot, set frequency to always.

To enable user data execution with EC2Launch (Windows Server 2016 or later)

1. Connect to your Windows instance.
2. Open a PowerShell command window and run the following command:

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -Schedule
```

3. Disconnect from your Windows instance. To run updated scripts the 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. 576).

**To enable user data execution with EC2Config (Windows Server 2012 R2 and earlier)**

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 run updated scripts the 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. 576).

**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 run every time you reboot or start the instance. When you select **As text**, the Amazon EC2 console performs the base64 encoding for you.

```
<powershell>
$file = $env:SystemRoot\Temp\DateTime\WindowsUser\.
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/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, Instance state, Stop instance**.

   **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 **Stop**. It can take a few minutes for the instance to stop.
5. With the instance still selected, choose **Actions, Instance settings, Edit user data**. You can't change the user data if the instance is running, but you can view it.
6. In the **Edit user data** dialog box, update the user data, and then choose **Save**. To run 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-yyyy-HH\"\")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

7. Start the instance. If you enabled user data execution for subsequent reboots or starts, the updated user data scripts are run 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. 595). For information about user data and the AWS CLI, see [User data and the AWS CLI](p. 595) 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 run user data scripts every time you reboot or start the instance, add `<persist>true</persist>`, as shown in the following example.
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`.

```
<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.

```
PHBvd2Vyc2hlbGw
+cDqgsZSW5hbWUcGZtchV0X2IgLU5ld05hbWUgXNlci1kJYXlhLXRlc3QNCjw8QzJzaGVzbD4=
```

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>

Example: Rename the instance to match the tag value

To read the tag value, rename the instance on first boot to match the tag value, and reboot, use the `Get-EC2Tag` command. To run this command successfully, you must have a role with `ec2:DescribeTags` permissions because tag information is unavailable in the metadata and must be retrieved by API call. For more information on how to attach a role to an instance, see Attaching an IAM Role to an Instance.

**Note**

This script fails on Windows Server versions prior to 2008.

```powershell
$nameValue = (get-ec2tag -filter @{Name="resource-id";Value=$instanceId},@{Name="key";Value="Name"}).Value
$pattern = "^\([!\-0-9]{1,15}$\[a-zA-Z0-9-]{1,15}$"#
#Verify Name Value satisfies best practices for Windows hostnames
If ($nameValue -match $pattern)
{Try
{Rename-Computer -NewName $nameValue -Restart -ErrorAction Stop}
Catch
{$_ErrorMessage = $_.Exception.Message
Write-Output "Rename failed: $_ErrorMessage"}
Else
{Throw "Provided name not a valid hostname. Please ensure Name value is between 1 and 15 characters in length and contains only alphanumeric or hyphen characters"}
</powershell>

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 (p. 596), for example, host name, events, and security groups.

You can also use instance metadata to access *user data* that you specified when launching your instance. For example, you can specify parameters for configuring your instance, or include a simple script. You can build generic AMIs and use user data to modify the configuration files supplied at launch time. For example, if you run web servers for various small businesses, they can all use the same generic AMI and retrieve their content from the Amazon S3 bucket that you specify in the user data at launch. To add a new customer at any time, create a bucket for the customer, add their content, and launch your AMI with the unique bucket name provided to your code in the user data. If you launch more than one instance at the same time, the user data is available to all instances in that reservation. Each instance that is part of the same reservation has a unique `ami-launch-index` number, allowing you to write code that controls what to do. For example, the first host might elect itself as the original node in a cluster.

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. 603).

**Important**

Although you can only access instance metadata and user data from within the instance itself, the data is not protected by authentication or cryptographic methods. Anyone who has direct access to the instance, and potentially any software running on the instance, can view its...
metadata. Therefore, you should not store sensitive data, such as passwords or long-lived encryption keys, as user data.

**Note**
The examples in this section use the IPv4 address of the instance metadata service: 169.254.169.254. If you are retrieving instance metadata for EC2 instances over the IPv6 address, ensure that you enable and use the IPv6 address instead: fd00:ec2::254. The IPv6 address of the instance metadata service is compatible with IMDSv2 commands. The IPv6 address is only accessible on Instances built on the Nitro System (p. 146).

**Contents**
- Use IMDSv2 (p. 580)
- Configure the instance metadata options (p. 583)
- Retrieve instance metadata (p. 586)
- Work with instance user data (p. 594)
- Retrieve dynamic data (p. 596)
- Instance metadata categories (p. 596)
- Instance identity documents (p. 604)

**Use IMDSv2**

You can access instance metadata from a running instance using one of the following methods:

- Instance Metadata Service Version 1 (IMDSv1) – a request/response method
- Instance Metadata Service Version 2 (IMDSv2) – a session-oriented method

By default, you can use either IMDSv1 or IMDSv2, or both. The instance metadata service distinguishes between IMDSv1 and IMDSv2 requests based on whether, for any given request, either the PUT or GET headers, which are unique to IMDSv2, are present in that request.

You can configure the instance metadata service on each instance such that local code or users must use IMDSv2. When you specify that IMDSv2 must be used, IMDSv1 no longer works. For more information, see Configure the instance metadata options (p. 583).

To retrieve instance metadata, see Retrieve instance metadata (p. 586).

**Note**
The examples in this section use the IPv4 address of the instance metadata service: 169.254.169.254. If you are retrieving instance metadata for EC2 instances over the IPv6 address, ensure that you enable and use the IPv6 address instead: fd00:ec2::254. The IPv6 address of the instance metadata service is compatible with IMDSv2 commands. The IPv6 address is only accessible on Instances built on the Nitro System (p. 146).

**How Instance Metadata Service Version 2 works**

IMDSv2 uses session-oriented requests. With session-oriented requests, you create a session token that defines the session duration, which can be a minimum of one second and a maximum of six hours. During the specified duration, you can use the same session token for subsequent requests. After the specified duration expires, you must create a new session token to use for future requests.

The following example uses a PowerShell shell script and IMDSv2 to retrieve the top-level instance metadata items. The example:

- Creates a session token lasting six hours (21,600 seconds) using the PUT request
- Stores the session token header in a variable named token
- Requests the top-level metadata items using the token
After you've created a token, you can reuse it until it expires. In the following example command, which
gets the ID of the AMI used to launch the instance, the token that is stored in $token in the previous
example is reused.

When you use IMDSv2 to request instance metadata, the request must include the following:

1. Use a PUT request to initiate a session to the instance metadata service. The PUT request returns a
   token that must be included in subsequent GET requests to the instance metadata service. The token
   is required to access metadata using IMDSv2.

2. Include the token in all GET requests to the instance metadata service. When token usage is set to
   required, requests without a valid token or with an expired token receive a 401 - Unauthorized
   HTTP error code. For information about changing the token usage requirement, see modify-instance-
   metadata-options in the AWS CLI Command Reference.

   • The token is an instance-specific key. The token is not valid on other EC2 instances and will be
     rejected if you attempt to use it outside of the instance on which it was generated.
   
   • The PUT request must include a header that specifies the time to live (TTL) for the token, in seconds,
     up to a maximum of six hours (21,600 seconds). The token represents a logical session. The TTL
     specifies the length of time that the token is valid and, therefore, the duration of the session.

   • After a token expires, to continue accessing instance metadata, you must create a new session using
     another PUT.

   • You can choose to reuse a token or create a new token with every request. For a small number of
     requests, it might be easier to generate and immediately use a token each time you need to access
     the instance metadata service. But for efficiency, you can specify a longer duration for the token
     and reuse it rather than having to write a PUT request every time you need to request instance
     metadata. There is no practical limit on the number of concurrent tokens, each representing its own
     session. IMDSv2 is, however, still constrained by normal instance metadata service connection and
     throttling limits. For more information, see Query throttling (p. 593).

HTTP GET and HEAD methods are allowed in IMDSv2 instance metadata requests. PUT requests are
rejected if they contain an X-Forwarded-For header.

By default, the response to PUT requests has a response hop limit (time to live) of 1 at the IP protocol
level. You can adjust the hop limit using the modify-instance-metadata-options command if you
need to make it larger. For example, you might need a larger hop limit for backward compatibility with
container services running on the instance. For more information, see modify-instance-metadata-options
in the AWS CLI Command Reference.

Transition to using Instance Metadata Service Version 2

Use of Instance Metadata Service Version 2 (IMDSv2) is optional. Instance Metadata Service Version
1 (IMDSv1) will continue to be supported indefinitely. If you choose to migrate to using IMDSv2, we
recommend that you use the following tools and transition path.

Tools for helping with the transition to IMDSv2

If your software uses IMDSv1, use the following tools to help reconfigure your software to use IMDSv2.
AWS software: The latest versions of the AWS SDKs and CLIs support IMDSv2. To use IMDSv2, make sure that your EC2 instances have the latest versions of the AWS SDKs and CLIs. For information about updating the CLI, see Installing, updating, and uninstalling the AWS CLI in the AWS Command Line Interface User Guide.

CloudWatch: IMDSv2 uses token-backed sessions, while IMDSv1 does not. The MetadataNoToken CloudWatch metric tracks the number of calls to the instance metadata service that are using IMDSv1. By tracking this metric to zero, you can determine if and when all of your software has been upgraded to use IMDSv2. For more information, see Instance metrics (p. 842).

Updates to EC2 APIs and CLIs: For existing instances, you can use the modify-instance-metadata-options CLI command (or the ModifyInstanceMetadataOptions API) to require the use of IMDSv2. For new instances, you can use the run-instances CLI command (or the RunInstances API) and the metadata-options parameter to launch new instances that require the use of IMDSv2.

To require the use of IMDSv2 on all new instances launched by Auto Scaling groups, your Auto Scaling groups can use either a launch template or a launch configuration. When you create a launch template or create a launch configuration, you must configure the MetadataOptions parameters to require the use of IMDSv2. After you configure the launch template or launch configuration, the Auto Scaling group launches new instances using the new launch template or launch configuration, but existing instances are not affected.

Use the modify-instance-metadata-options CLI command (or the ModifyInstanceMetadataOptions API) to require the use of IMDSv2 on the existing instances, or terminate the instances and the Auto Scaling group will launch new replacement instances with the instance metadata options settings that are defined in the launch template or launch configuration.

IAM policies and SCPs: You can use an IAM condition to enforce that IAM users can't launch an instance unless it uses IMDSv2. You can also use IAM conditions to enforce that IAM users can't modify running instances to re-enable IMDSv1, and to enforce that the instance metadata service is available on the instance.

The ec2:MetadataHttpTokens, ec2:MetadataHttpPutResponseHopLimit, and ec2:MetadataHttpEndpoint IAM condition keys can be used to control the use of the RunInstances and the ModifyInstanceMetadataOptions API and corresponding CLI. If a policy is created, and a parameter in the API call does not match the state specified in the policy using the condition key, the API or CLI call fails with an UnauthorizedOperation response. These condition keys can be used either in IAM policies or AWS Organizations service control policies (SCPs).

Furthermore, you can choose an additional layer of protection to enforce the change from IMDSv1 to IMDSv2. At the access management layer with respect to the APIs called via EC2 Role credentials, you can use a new condition key in either IAM policies or AWS Organizations service control policies (SCPs). Specifically, by using the policy condition key ec2:RoleDelivery with a value of 2.0 in your IAM policies, API calls made with EC2 Role credentials obtained from IMDSv1 will receive an UnauthorizedOperation response. The same thing can be achieved more broadly with that condition required by an SCP. This ensures that credentials delivered via IMDSv1 cannot actually be used to call APIs because any API calls not matching the specified condition will receive an UnauthorizedOperation error. For example IAM policies, see Work with instance metadata (p. 1102). For more information, see Service Control Policies in the AWS Organizations User Guide.

Recommended path to requiring IMDSv2 access

Using the above tools, we recommend that you follow this path for transitioning to IMDSv2:

Step 1: At the start

Update the SDKs, CLIs, and your software that use Role credentials on their EC2 instances to IMDSv2-compatible versions. For information about updating the CLI, see Upgrading to the latest version of the AWS CLI in the AWS Command Line Interface User Guide.
Then, change your software that directly accesses instance metadata (in other words, that does not use an SDK) using the IMDSv2 requests.

**Step 2: During the transition**

Track your transition progress by using the CloudWatch metric `MetadataNoToken`. This metric shows the number of calls to the instance metadata service that are using IMDSv1 on your instances. For more information, see Instance metrics (p. 842).

**Step 3: When everything is ready on all instances**

Everything is ready on all instances when the CloudWatch metric `MetadataNoToken` records zero IMDSv1 usage. At this stage, you can do the following:

- For existing instances: You can require IMDSv2 use through the `modify-instance-metadata-options` command. You can make these changes on running instances; you do not need to restart your instances.
- For new instances: When launching a new instance, you can do one of the following:
  - In the Amazon EC2 console launch instance wizard, set **Metadata accessible** to **Enabled** and **Metadata version** to **V2**. For more information, see Step 3: Configure Instance Details (p. 394).
  - Use the `run-instances` command to specify that only IMDSv2 is to be used.

Updating instance metadata options for existing instances is available only through the API or AWS CLI. It is currently not available in the Amazon EC2 console. For more information, see Configure the instance metadata options (p. 583).

**Step 4: When all of your instances are transitioned to IMDSv2**

The `ec2:MetadataHttpTokens`, `ec2:MetadataHttpPutResponseHopLimit`, and `ec2:MetadataHttpEndpoint` IAM condition keys can be used to control the use of the `RunInstances` and the `ModifyInstanceMetadataOptions` API and corresponding CLI. If a policy is created, and a parameter in the API call does not match the state specified in the policy using the condition key, the API or CLI call fails with an `UnauthorizedOperation` response. For example IAM policies, see Work with instance metadata (p. 1102).

**Configure the instance metadata options**

Instance metadata options allow you to configure new or existing instances to do the following:

- Require the use of IMDSv2 when requesting instance metadata
- Specify the **PUT** response hop limit
- Turn off access to instance metadata

You can also use IAM condition keys in an IAM policy or SCP to do the following:

- Allow an instance to launch only if it's configured to require the use of IMDSv2
- Restrict the number of allowed hops
- Turn off access to instance metadata

**Note**

If your PowerShell version is earlier than 4.0, you must update to Windows Management Framework 4.0 to require the use of IMDSv2.

**Note**

You should proceed cautiously and conduct careful testing before making any changes. Take note of the following:
• If you enforce the use of IMDSv2, applications or agents that use IMDSv1 for instance metadata access will break.
• If you turn off all access to instance metadata, applications or agents that rely on instance metadata access to function will break.
• For IMDSv2, you must use /latest/api/ token when retrieving the token.

Topics
• Configure instance metadata options for new instances (p. 584)
• Modify instance metadata options for existing instances (p. 585)

Configure instance metadata options for new instances

You can require the use of IMDSv2 on an instance when you launch it. You can also create an IAM policy that prevents users from launching new instances unless they require IMDSv2 on the new instance.

Console

To require the use of IMDSv2 on a new instance

• When launching a new instance in the Amazon EC2 console, select the following options on the Configure Instance Details page:
  • Under Advanced Details, for Metadata accessible, select Enabled.
  • For Metadata version, select V2 (token required).

For more information, see Step 3: Configure Instance Details (p. 394).

AWS CLI

To require the use of IMDSv2 on a new instance

The following run-instances example launches a c3.large instance with --metadata-options set to HttpTokens=required. When you specify a value for HttpTokens, you must also set HttpEndpoint to enabled. Because the secure token header is set to required for metadata retrieval requests, this opts in the instance to require using IMDSv2 when requesting instance metadata.

```bash
aws ec2 run-instances
  --image-id ami-0abcdef1234567890
  --instance-type c3.large
  ...
  --metadata-options "HttpEndpoint=enabled,HttpTokens=required"
```

To enforce the use of IMDSv2 on all new instances

To ensure that IAM users can only launch instances that require the use of IMDSv2 when requesting instance metadata, you can specify that the condition to require IMDSv2 must be met before an instance can be launched. For the example IAM policy, see Work with instance metadata (p. 1102).

Console

To turn off access to instance metadata

• To ensure that access to your instance metadata is turned off, regardless of which version of the instance metadata service you are using, launch the instance in the Amazon EC2 console with the following option selected on the Configure Instance Details page:
Instance metadata and user data

- Under **Advanced Details**, for **Metadata accessible**, select **Disabled**.

For more information, see Step 3: Configure Instance Details (p. 394).

**AWS CLI**

**To turn off access to instance metadata**

To ensure that access to your instance metadata is turned off, regardless of which version of the instance metadata service you are using, launch the instance with `--metadata-options` set to `HttpEndpoint=disabled`. You can turn access on later by using the `modify-instance-metadata-options` command.

```
aws ec2 run-instances
  --image-id ami-0abcdef1234567890
  --instance-type c3.large
  ...
  --metadata-options "HttpEndpoint=disabled"
```

**Modify instance metadata options for existing instances**

You can require the use IMDSv2 on an existing instance. You can also change the PUT response hop limit and turn off access to instance metadata on an existing instance. You can also create an IAM policy that prevents users from modifying the instance metadata options on an existing instance.

Currently only the AWS SDK or AWS CLI support modifying the instance metadata options on existing instances. You can't use the Amazon EC2 console for modifying instance metadata options.

**To require the use of IMDSv2**

You can opt in to require that IMDSv2 is used when requesting instance metadata. Use the `modify-instance-metadata-options` CLI command and set the `http-tokens` parameter to `required`. When you specify a value for `http-tokens`, you must also set `http-endpoint` to `enabled`.

```
aws ec2 modify-instance-metadata-options
  --instance-id i-123456789abcdef0
  --http-tokens required
  --http-endpoint enabled
```

**To change the PUT response hop limit**

For existing instances, you can change the settings of the PUT response hop limit. Use the `modify-instance-metadata-options` CLI command and set the `http-put-response-hop-limit` parameter to the required number of hops. In the following example, the hop limit is set to 3. Note that when specifying a value for `http-put-response-hop-limit`, you must also set `http-endpoint` to `enabled`.

```
aws ec2 modify-instance-metadata-options
  --instance-id i-123456789abcdef0
  --http-put-response-hop-limit 3
  --http-endpoint enabled
```

**To restore the use of IMDSv1 on an instance using IMDSv2**

You can use the `modify-instance-metadata-options` CLI command with `http-tokens` set to `optional` to restore the use of IMDSv1 when requesting instance metadata.
To turn off access to instance metadata

You can turn off access to your instance metadata by disabling the HTTP endpoint of the instance metadata service, regardless of which version of the instance metadata service you are using. You can reverse this change at any time by enabling the HTTP endpoint. Use the `modify-instance-metadata-options` CLI command and set the `http-endpoint` parameter to `disabled`.

```bash
aws ec2 modify-instance-metadata-options \
--instance-id i-1234567898abcdef0 \
--http-endpoint disabled
```

To control the use of `modify-instance-metadata-options`

To control which IAM users can modify the instance metadata options, specify a policy that prevents all users other than users with a specified role to use the `ModifyInstanceMetadataOptions` API. For the example IAM policy, see Work with instance metadata (p. 1102).

Retrieve instance metadata

Because your instance metadata is available from your running instance, you do not need to use the Amazon EC2 console or the AWS CLI. This can be helpful when you're writing scripts to run from your instance. For example, you can access the local IP address of your instance from instance metadata to manage a connection to an external application.

Instance metadata is divided into categories. For a description of each instance metadata category, see Instance metadata categories (p. 596).

To view all categories of instance metadata from within a running instance, use the following IPv4 or IPv6 URIs:

```
```

```
http://[fd00:ec2::254]/latest/meta-data/
```

The IP addresses are link-local address and are valid only from the instance. For more information, see Link-local address on Wikipedia.

**Note**

The examples in this section use the IPv4 address of the instance metadata service: 169.254.169.254. If you are retrieving instance metadata for EC2 instances over the IPv6 address, ensure that you enable and use the IPv6 address instead: fd00:ec2::254. The IPv6 address of the instance metadata service is compatible with IMDSv2 commands. The IPv6 address is only accessible on Instances built on the Nitro System (p. 146).

The command format is different, depending on whether you use IMDSv1 or IMDSv2. By default, you can use both instance metadata services. To require the use of IMDSv2, see Use IMDSv2 (p. 580).

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.
If you don't want to use PowerShell, you can install a third-party tool such as GNU Wget or cURL.

**Important**
If you 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.

Note that you are not billed for HTTP requests used to retrieve instance metadata and user data.

**Considerations**

To avoid problems with instance metadata retrieval, consider the following:

- The AWS SDKs use IMDSv2 calls by default. If the IMDSv2 call receives no response, the SDK retries the call and, if still unsuccessful, uses IMDSv1. This can result in a delay. In a container environment, if the hop limit is 1, the IMDSv2 response does not return because going to the container is considered an additional network hop. To avoid the process of falling back to IMDSv1 and the resultant delay, in a container environment we recommend that you set the hop limit to 2. For more information, see Configure the instance metadata options (p. 583).
- If you launch a Windows instance using a custom Windows AMI, to ensure that the instance metadata service works on the instance, the AMI must be a standardized image created using Sysprep (p. 40). Otherwise, the instance metadata service won't work.
- For IMDSv2, you must use /latest/api/token when retrieving the token. Issuing PUT requests to any version-specific path, for example /2021-03-23/api/token, will result in the metadata service returning 403 Forbidden errors. This behavior is intended.

**Responses and error messages**

All instance metadata is returned as text (HTTP content type text/plain).

A request for a specific metadata resource returns the appropriate value, or a 404 - Not Found HTTP error code if the resource is not available.

A request for a general metadata resource (the URI ends with a /) returns a list of available resources, or a 404 - Not Found HTTP error code if there is no such resource. The list items are on separate lines, terminated by line feeds (ASCII 10).

For requests made using Instance Metadata Service Version 2, the following HTTP error codes can be returned:

- 400 - Missing or Invalid Parameters - The PUT request is not valid.
Amazon Elastic Compute Cloud
User Guide for Windows Instances
Instance metadata and user data

• 401 - Unauthorized – The GET request uses an invalid token. The recommended action is to
generate a new token.
• 403 - Forbidden – The request is not allowed or the instance metadata service is turned oﬀ.

Examples of retrieving instance metadata
Examples
• Get the available versions of the instance metadata (p. 588)
• Get the top-level metadata items (p. 589)
• Get the list of available public keys (p. 591)
• Show the formats in which public key 0 is available (p. 591)
• Get public key 0 (in the OpenSSH key format) (p. 591)
• Get the subnet ID for an instance (p. 592)

Get the available versions of the instance metadata
This example gets the available versions of the instance metadata. These versions do not necessarily
correlate with an Amazon EC2 API version. The earlier versions are available to you in case you have
scripts that rely on the structure and information present in a previous version.
IMDSv2
PS C:\> $token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" =

1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
2016-06-30
2016-09-02
latest

IMDSv1
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15

588


Amazon Elastic Compute Cloud
User Guide for Windows Instances
Instance metadata and user data
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
2016-06-30
2016-09-02
latest

Get the top-level metadata items
This example gets the top-level metadata items. For more information, see Instance metadata
categories (p. 596).
IMDSv2
PS C:\> $token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" =

ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
iam/
instance-action
instance-id
instance-life-cycle
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/

IMDSv1
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
iam/
instance-action
instance-id

589


Instance metadata and user data

The following examples get the values of some of the top-level metadata items that were obtained in the preceding example. The IMDSv2 requests use the stored token that was created in the preceding example command, assuming it has not expired.

**IMDSv2**

```powershell
ami-0abcdef1234567890
```

**IMDSv1**

```powershell
ami-0abcdef1234567890
```

**IMDSv2**

```powershell
r-0efghijk987654321
```

**IMDSv1**

```powershell
r-0efghijk987654321
```

**IMDSv2**

```powershell
ip-10-251-50-12.ec2.internal
```

**IMDSv1**

```powershell
ip-10-251-50-12.ec2.internal
```
Amazon Elastic Compute Cloud
User Guide for Windows Instances
Instance metadata and user data

IMDSv2
ec2-203-0-113-25.compute-1.amazonaws.com

IMDSv1
ec2-203-0-113-25.compute-1.amazonaws.com

Get the list of available public keys
This example gets the list of available public keys.
IMDSv2
PS C:\> $token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" =
0=my-public-key

IMDSv1

Show the formats in which public key 0 is available
This example shows the formats in which public key 0 is available.
IMDSv2
PS C:\> $token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" =
openssh-key

IMDSv1
openssh-key
openssh-key

Get public key 0 (in the OpenSSH key format)
This example gets public key 0 (in the OpenSSH key format).

591


Get the subnet ID for an instance

This example gets the subnet ID for an instance.

IMDSv2

```powershell

```
Query throttling

We throttle queries to the instance metadata service on a per-instance basis, and we place limits on the number of simultaneous connections from an instance to the instance metadata service.

If you're using the instance metadata service to retrieve AWS security credentials, avoid querying for credentials during every transaction or concurrently from a high number of threads or processes, as this might lead to throttling. Instead, we recommend that you cache the credentials until they start approaching their expiry time.

If you are throttled while accessing the instance metadata service, retry your query with an exponential backoff strategy.

Limit instance metadata service access

You can consider using local firewall rules to disable access from some or all processes to the instance metadata service.

**Note**

For *Instances built on the Nitro System (p. 146)*, IMDS can be reached from your own network when a network appliance within your VPC, such as a virtual router, forwards packets to the IMDS address, and the default *source/destination check* on the instance is disabled. To prevent a source from outside your VPC reaching IMDS, we recommend that you modify the configuration of the network appliance to drop packets with the destination IPv4 address of IMDS 169.254.169.254 and, if you enabled the IPv6 endpoint, the IPv6 address of IMDS fd00:ec2::254.

Using Windows firewall to limit access

The following PowerShell example uses the built-in Windows firewall to prevent the Internet Information Server webserver (based on its default installation user ID of *NT AUTHORITY\IUSR*) from accessing 169.254.169.254. It uses a *deny rule* to reject all instance metadata requests (whether IMDSv1 or IMDSv2) from any process running as that user.

```
PS C:\> $blockPrincipal = New-Object -TypeName System.Security.Principal.NTAccount ("NT AUTHORITY\IUSR")
PS C:\> $BlockPrincipalSID = $blockPrincipal.Translate([System.Security.Principal.SecurityIdentifier]).Value
PS C:\> $BlockPrincipalSDDL = "D:(A;;CC;;;$BlockPrincipalSID)"
PS C:\> New-NetFirewallRule -DisplayName "Block metadata service from IIS" -Action block -Direction out ` -Protocol TCP -RemoteAddress 169.254.169.254 -LocalUser $BlockPrincipalSDDL
```

Or, you can consider only allowing access to particular users or groups, by using *allow rules*. Allow rules might be easier to manage from a security perspective, because they require you to make a decision about what software needs access to instance metadata. If you use *allow rules*, it's less likely you will accidentally allow software to access the metadata service (that you did not intend to have access) if you later change the software or configuration on an instance. You can also combine group usage with allow rules, so that you can add and remove users from a permitted group without needing to change the firewall rule.

The following example prevents access to instance metadata by all processes running as an OS group specified in the variable `blockPrincipal` (in this example, the Windows group *Everyone*), except for processes specified in `exceptionPrincipal` (in this example, a group called *trustworthy-users*). You must specify both deny and allow principals because Windows Firewall, unlike the *--uid-owner trusted-user* rule in Linux `iptables`, does not provide a shortcut mechanism to allow only a particular principal (user or group) by denying all the others.
Instance metadata and user data

Note
To use local firewall rules, you need to adapt the preceding example commands to suit your needs.

Using netsh rules to limit access
You can consider blocking all software using netsh rules, but those are much less flexible.

C:\> netsh advfirewall firewall add rule name="Block metadata service altogether" dir=out protocol=TCP remoteip=169.254.169.254 action=block

Note
• To use local firewall rules, you need to adapt the preceding example commands to suit your needs.
• netsh rules must be set from an elevated command prompt, and can't be set to deny or allow particular principals.

Work with instance user data
When working with instance user data, keep the following in mind:

• User data must be base64-encoded. The Amazon EC2 console can perform the base64-encoding for you or accept base64-encoded input.
• User data is limited to 16 KB, in raw form, before it is base64-encoded. The size of a string of length $n$ after base64-encoding is $\text{ceil}(n/3)*4$.
• User data must be base64-decoded when you retrieve it. If you retrieve the data using instance metadata or the console, it's decoded for you automatically.
• User data is treated as opaque data: what you give is what you get back. It is up to the instance to be able to interpret it.
• If you stop an instance, modify its user data, and start the instance, the updated user data is not run automatically when you start the instance. However, you can configure settings so that updated user data scripts are run 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 run one time at launch, or every time you reboot or start the instance. For more information, see Run commands on your Windows instance at launch (p. 572).
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. 576).

Retrieve instance user data

**Note**
The examples in this section use the IPv4 address of the instance metadata service: 169.254.169.254. If you are retrieving instance metadata for EC2 instances over the IPv6 address, ensure that you enable and use the IPv6 address instead: fd00:ec2::254. The IPv6 address of the instance metadata service is compatible with IMDSv2 commands. The IPv6 address is only accessible on Instances built on the Nitro System (p. 146).

To retrieve user data from within a running instance, use the following URI.

http://169.254.169.254/latest/user-data

A request for user data returns the data as it is (content type application/octet-stream).

This example returns user data that was provided as comma-separated text.

**IMDSv2**

```

1234,john,reboot,true | 4512,richard, | 173,,,
```

**IMDSv1**

```
1234,john,reboot,true | 4512,richard, | 173,,,
```

This example returns user data that was provided as a script.

**IMDSv2**

```

<powershell>
$file = $env:SystemRoot + "$\Temp\" + (Get-Date).ToString("MM-dd-yyyy-mm")
New-Item $file -ItemType file
</powershell>
<persist>true</persist>
```

**IMDSv1**

```
```
To retrieve user data for an instance from your own computer, see User data and the Tools for Windows PowerShell (p. 577).

**Retrieve dynamic data**

To retrieve dynamic data from within a running instance, use the following URI.


**Note**

The examples in this section use the IPv4 address of the instance metadata service: 169.254.169.254. If you are retrieving instance metadata for EC2 instances over the IPv6 address, ensure that you enable and use the IPv6 address instead: fd00:ec2::254. The IPv6 address of the instance metadata service is compatible with IMDSv2 commands. The IPv6 address is only accessible on Instances built on the Nitro System (p. 146).

This example shows how to retrieve the high-level instance identity categories.

**IMDSv2**

```powershell
```

```powershell
```

**IMDSv1**

```powershell
```

For more information about dynamic data and examples of how to retrieve it, see Instance identity documents (p. 604).

**Instance metadata categories**

Instance metadata is divided into categories. When you retrieve instance metadata, these are the top-level items.

When Amazon EC2 releases a new instance metadata category, the instance metadata for the new category might not be available for existing instances. With instances built on the Nitro system (p. 146),
you can retrieve instance metadata only for the categories that were available at launch. For instances with the Xen hypervisor, you can **stop and then start** (p. 425) the instance to update the categories that are available for the instance.

The following table lists the categories of instance metadata. Some of the category names include 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 actual values when you retrieve the instance metadata.

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>ami-id</td>
<td>The AMI ID used to launch the instance.</td>
<td>1.0</td>
</tr>
<tr>
<td>ami-launch-index</td>
<td>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.</td>
<td>1.0</td>
</tr>
<tr>
<td>ami-manifest-path</td>
<td>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.</td>
<td>1.0</td>
</tr>
<tr>
<td>ancestor-ami-ids</td>
<td>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.</td>
<td>2007-10-10</td>
</tr>
<tr>
<td>block-device-mapping/ami</td>
<td>The virtual device that contains the root/boot file system.</td>
<td>2007-12-15</td>
</tr>
<tr>
<td>block-device-mapping/ebs N</td>
<td>The virtual devices associated with any Amazon EBS volumes. Amazon EBS volumes are only available in metadata if they were present at launch time or when the instance was last started. The <code>N</code> indicates the index of the Amazon EBS volume (such as ebs1 or ebs2).</td>
<td>2007-12-15</td>
</tr>
<tr>
<td>block-device-mapping/eph N</td>
<td>The virtual devices for any non-NVMe instance store volumes. The <code>N</code> indicates the index of each volume. The number of instance store volumes in the block device mapping might not match the actual number of instance store volumes for the instance. The instance type determines the number of instance store volumes that are available to an instance. If the number of instance store volumes in a block device mapping exceeds the number available to an instance, the</td>
<td>2007-12-15</td>
</tr>
<tr>
<td>Data</td>
<td>Description</td>
<td>Version</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>block-device-mapping/root</td>
<td>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.</td>
<td>2007-12-15</td>
</tr>
<tr>
<td>block-device-mapping/swap</td>
<td>The virtual devices associated with swap. Not always present.</td>
<td>2007-12-15</td>
</tr>
<tr>
<td>elastic-gpus/associations/elastic-gpu-id</td>
<td>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.</td>
<td>2016-11-30</td>
</tr>
<tr>
<td>elastic-inference/associations/eia-id</td>
<td>If there is an Elastic Inference accelerator attached to the instance, contains a JSON string with information about the Elastic Inference accelerator, including its ID and type.</td>
<td>2018-11-29</td>
</tr>
<tr>
<td>events/maintenance/history</td>
<td>If there are completed or canceled maintenance events for the instance, contains a JSON string with information about the events. For more information, see View event history about completed or canceled events (p. 820).</td>
<td>2018-08-17</td>
</tr>
<tr>
<td>events/maintenance/scheduled</td>
<td>If there are active maintenance events for the instance, contains a JSON string with information about the events. For more information, see View scheduled events (p. 816).</td>
<td>2018-08-17</td>
</tr>
<tr>
<td>events/recommendations/rebalance</td>
<td>The approximate time, in UTC, when the EC2 instance rebalance recommendation notification is emitted for the instance. The following is an example of the metadata for this category: {&quot;noticeTime&quot;: &quot;2020-11-05T08:22:00Z&quot;}. This category is available only after the notification is emitted. For more information, see EC2 instance rebalance recommendations (p. 314).</td>
<td>2020-11-04</td>
</tr>
<tr>
<td>Data</td>
<td>Description</td>
<td>Version</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>hostname</td>
<td>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).</td>
<td>Version 1.0</td>
</tr>
<tr>
<td>iam/info</td>
<td>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.</td>
<td>2012-01-12</td>
</tr>
<tr>
<td>iam/security-credentials/role-name</td>
<td>If there is an IAM role associated with the instance, <code>role-name</code> is the name of the role, and <code>role-name</code> contains the temporary security credentials associated with the role (for more information, see Retrieve security credentials from instance metadata (p. 1115)). Otherwise, not present.</td>
<td>2012-01-12</td>
</tr>
<tr>
<td>identity-credentials/ec2/info</td>
<td>[Internal use only] Information about the credentials in identity-credentials/ec2/security-credentials/ec2-instance. These credentials are used by AWS features such as EC2 Instance Connect, and do not have any additional AWS API permissions or privileges beyond identifying the instance.</td>
<td>2018-05-23</td>
</tr>
<tr>
<td>identity-credentials/ec2/security-credentials/ec2-instance</td>
<td>[Internal use only] Credentials that allow on-instance software to identify itself to AWS to support features such as EC2 Instance Connect. These credentials do not have any additional AWS API permissions or privileges.</td>
<td>2018-05-23</td>
</tr>
<tr>
<td>instance-action</td>
<td>Notifies the instance that it should reboot in preparation for bundling. Valid values: none</td>
<td>shutdown</td>
</tr>
<tr>
<td>instance-id</td>
<td>The ID of this instance.</td>
<td>Version 1.0</td>
</tr>
<tr>
<td>instance-life-cycle</td>
<td>The purchasing option of this instance. For more information, see Instance purchasing options (p. 239).</td>
<td>2019-10-01</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td><strong>Description</strong></td>
<td><strong>Version</strong></td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>instance-type</code></td>
<td>The type of instance. For more information, see Instance types (p. 141).</td>
<td>2007-08-29</td>
</tr>
<tr>
<td><code>kernel-id</code></td>
<td>The ID of the kernel launched with this instance, if applicable.</td>
<td>2008-02-01</td>
</tr>
<tr>
<td><code>local-hostname</code></td>
<td>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).</td>
<td>2007-01-19</td>
</tr>
<tr>
<td><code>local-ipv4</code></td>
<td>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).</td>
<td>Version 1.0</td>
</tr>
<tr>
<td><code>mac</code></td>
<td>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).</td>
<td>2011-01-01</td>
</tr>
<tr>
<td><code>metrics/vhostmd</code></td>
<td>No longer available.</td>
<td>2011-05-01</td>
</tr>
<tr>
<td><code>network/interfaces/macs/mac/device-number</code></td>
<td>The unique device number associated with that interface. The device number corresponds to the device name; for example, a <code>device-number</code> of 2 is for the eth2 device. This category corresponds to the DeviceIndex and device-index fields that are used by the Amazon EC2 API and the EC2 commands for the AWS CLI.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td><code>network/interfaces/macs/mac/interface-id</code></td>
<td>The ID of the network interface.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td><code>network/interfaces/macs/mac/ipv4-associations/public-ip</code></td>
<td>The private IPv4 addresses that are associated with each public IP address and assigned to that interface.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td><code>network/interfaces/macs/mac/ipv6s</code></td>
<td>The IPv6 addresses associated with the interface. Returned only for instances launched into a VPC.</td>
<td>2016-06-30</td>
</tr>
<tr>
<td><code>network/interfaces/macs/mac/local-hostname</code></td>
<td>The interface's local hostname.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td><code>network/interfaces/macs/mac/local-ipv4s</code></td>
<td>The private IPv4 addresses associated with the interface.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td>Data</td>
<td>Description</td>
<td>Version</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>network/interfaces/macs/mac/mac</td>
<td>The instance's MAC address.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td>network/interfaces/macs/mac/network-card-index</td>
<td>The index of the network card. Some instance types support multiple network cards.</td>
<td>2020-11-01</td>
</tr>
<tr>
<td>network/interfaces/macs/mac/owner-id</td>
<td>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.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td>network/interfaces/macs/mac/public-hostname</td>
<td>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.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td>network/interfaces/macs/mac/public-ipv4s</td>
<td>The public IP address or Elastic IP addresses associated with the interface. There may be multiple IPv4 addresses on an instance.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td>network/interfaces/macs/mac/security-groups</td>
<td>Security groups to which the network interface belongs.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td>network/interfaces/macs/mac/security-group-ids</td>
<td>The IDs of the security groups to which the network interface belongs.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td>network/interfaces/macs/mac/subnet-id</td>
<td>The ID of the subnet in which the interface resides.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td>network/interfaces/macs/mac/subnet-ipv4-cidr-block</td>
<td>The IPv4 CIDR block of the subnet in which the interface resides.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td>network/interfaces/macs/mac/subnet-ipv6-cidr-blocks</td>
<td>The IPv6 CIDR block of the subnet in which the interface resides.</td>
<td>2016-06-30</td>
</tr>
<tr>
<td>network/interfaces/macs/mac/vpc-id</td>
<td>The ID of the VPC in which the interface resides.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td>network/interfaces/macs/mac/vpc-ipv4-cidr-block</td>
<td>The primary IPv4 CIDR block of the VPC.</td>
<td>2011-01-01</td>
</tr>
<tr>
<td>network/interfaces/macs/mac/vpc-ipv4-cidr-blocks</td>
<td>The IPv4 CIDR blocks for the VPC.</td>
<td>2016-06-30</td>
</tr>
<tr>
<td>network/interfaces/macs/mac/vpc-ipv6-cidr-blocks</td>
<td>The IPv6 CIDR block of the VPC in which the interface resides.</td>
<td>2016-06-30</td>
</tr>
<tr>
<td>placement/availability-zone</td>
<td>The Availability Zone in which the instance launched.</td>
<td>2008-02-01</td>
</tr>
<tr>
<td>Data</td>
<td>Description</td>
<td>Version</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>placement/availability-zone-id</td>
<td>The static Availability Zone ID in which the instance is launched. The Availability Zone ID is consistent across accounts. However, it might be different from the Availability Zone, which can vary by account.</td>
<td>2020-08-24</td>
</tr>
<tr>
<td>placement/group-name</td>
<td>The name of the placement group in which the instance is launched.</td>
<td>2020-08-24</td>
</tr>
<tr>
<td>placement/host-id</td>
<td>The ID of the host on which the instance is launched. Applicable only to Dedicated Hosts.</td>
<td>2020-08-24</td>
</tr>
<tr>
<td>placement/partition-number</td>
<td>The number of the partition in which the instance is launched.</td>
<td>2020-08-24</td>
</tr>
<tr>
<td>placement/region</td>
<td>The AWS Region in which the instance is launched.</td>
<td>2020-08-24</td>
</tr>
<tr>
<td>product-codes</td>
<td>AWS Marketplace product codes associated with the instance, if any.</td>
<td>2007-03-01</td>
</tr>
<tr>
<td>public-hostname</td>
<td>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.</td>
<td>2007-01-19</td>
</tr>
<tr>
<td>public-ipv4</td>
<td>The public IPv4 address. If an Elastic IP address is associated with the instance, the value returned is the Elastic IP address.</td>
<td>2007-01-19</td>
</tr>
<tr>
<td>public-keys/0/openssh-key</td>
<td>Public key. Only available if supplied at instance launch time.</td>
<td>Version 1.0</td>
</tr>
<tr>
<td>ramdisk-id</td>
<td>The ID of the RAM disk specified at launch time, if applicable.</td>
<td>2007-10-10</td>
</tr>
<tr>
<td>reservation-id</td>
<td>The ID of the reservation.</td>
<td>Version 1.0</td>
</tr>
<tr>
<td>security-groups</td>
<td>The names of the security groups applied to the instance. After launch, you can change the security groups of the instances. Such changes are reflected here and in network/interfaces/macs/mac/security-groups.</td>
<td>Version 1.0</td>
</tr>
<tr>
<td>services/domain</td>
<td>The domain for AWS resources for the Region.</td>
<td>2014-02-25</td>
</tr>
<tr>
<td>Data</td>
<td>Description</td>
<td>Version</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>services/partition</td>
<td>The partition that the resource is in. For standard AWS Regions, the partition is <code>aws</code>. If you have resources in other partitions, the partition is <code>aws-partitionname</code>. For example, the partition for resources in the China (Beijing) Region is <code>aws-cn</code>.</td>
<td>2015-10-20</td>
</tr>
<tr>
<td>spot/instance-action</td>
<td>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 <code>instance-action</code> (p. 322).</td>
<td>2016-11-15</td>
</tr>
<tr>
<td>spot/termination-time</td>
<td>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 <code>termination-time</code> (p. 323).</td>
<td>2014-11-05</td>
</tr>
</tbody>
</table>

### Dynamic data categories

The following table lists the categories of dynamic data.

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>fws/instance-monitoring</td>
<td>Value showing whether the customer has enabled detailed one-minute monitoring in CloudWatch. Valid values: enabled</td>
<td>2009-04-04</td>
</tr>
<tr>
<td>instance-identity/document</td>
<td>JSON containing instance attributes, such as instance-id, private IP address, etc. See Instance identity documents (p. 604).</td>
<td>2009-04-04</td>
</tr>
<tr>
<td>instance-identity/pkcs7</td>
<td>Used to verify the document's authenticity and content against the signature. See Instance identity documents (p. 604).</td>
<td>2009-04-04</td>
</tr>
<tr>
<td>instance-identity/signature</td>
<td>Data that can be used by other parties to verify its origin and authenticity. See Instance identity documents (p. 604).</td>
<td>2009-04-04</td>
</tr>
</tbody>
</table>
## Instance identity documents

Each instance that you launch has an instance identity document that provides information about the instance itself. You can use the instance identity document to validate the attributes of the instance.

The instance identity document is generated when the instance is launched and it is exposed (in plaintext JSON format) through the Instance Metadata Service. The IPv4 address 169.254.169.254 is a link-local address and is valid only from the instance. For more information, see [Link-local address](https://en.wikipedia.org/wiki/Link-local_address) on Wikipedia. The IPv6 address fd00:ec2::254 is a unique local address and is valid only from the instance. For more information, see [Unique local address](https://en.wikipedia.org/wiki/Unique_local_address) on Wikipedia.

**Note**

The examples in this section use the IPv4 address of the instance metadata service: 169.254.169.254. If you are retrieving instance metadata for EC2 instances over the IPv6 address, ensure that you enable and use the IPv6 address instead: fd00:ec2::254. The IPv6 address of the instance metadata service is compatible with IMDSv2 commands. The IPv6 address is only accessible on [Instances built on the Nitro System](https://docs.aws.amazon.com/en_us/instances/latest/nitro/index.html) (p. 146).

You can retrieve the instance identity document from a running instance at any time. The instance identity document includes the following information:

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>devpayProductCodes</td>
<td>Deprecated.</td>
</tr>
<tr>
<td>marketplaceProductCodes</td>
<td>The AWS Marketplace product code of the AMI used to launch the instance.</td>
</tr>
<tr>
<td>availabilityZone</td>
<td>The Availability Zone in which the instance is running.</td>
</tr>
<tr>
<td>privateIp</td>
<td>The private IPv4 address of the instance.</td>
</tr>
<tr>
<td>version</td>
<td>The version of the instance identity document format.</td>
</tr>
<tr>
<td>instanceId</td>
<td>The ID of the instance.</td>
</tr>
<tr>
<td>billingProducts</td>
<td>The billing product code of the AMI used to launch the instance.</td>
</tr>
<tr>
<td>instanceType</td>
<td>The instance type of the instance.</td>
</tr>
<tr>
<td>accountId</td>
<td>The ID of the AWS account that launched the instance.</td>
</tr>
<tr>
<td>imageId</td>
<td>The ID of the AMI used to launch the instance.</td>
</tr>
<tr>
<td>pendingTime</td>
<td>The date and time that the instance was launched.</td>
</tr>
<tr>
<td>architecture</td>
<td>The architecture of the AMI used to launch the instance (i386</td>
</tr>
<tr>
<td>kernelId</td>
<td>The ID of the kernel associated with the instance, if applicable.</td>
</tr>
<tr>
<td>ramdiskId</td>
<td>The ID of the RAM disk associated with the instance, if applicable.</td>
</tr>
<tr>
<td>region</td>
<td>The Region in which the instance is running.</td>
</tr>
</tbody>
</table>

### Retrieve the plaintext instance identity document

**To retrieve the plaintext instance identity document**

Connect to the instance and run one of the following commands depending on the Instance Metadata Service (IMDS) version used by the instance.
IMDSv2

```powershell
```

IMDSv1

```powershell
```

The following is example output.

```json
{
    "devpayProductCodes" : null,
    "marketplaceProductCodes" : [ "1abc2defghijklm3nopqrs4tu" ],
    "availabilityZone" : "us-west-2b",
    "privateIp" : "10.158.112.84",
    "version" : "2017-09-30",
    "instanceId" : "i-1234567890abcdef0",
    "billingProducts" : null,
    "instanceType" : "t2.micro",
    "accountId" : "123456789012",
    "imageId" : "ami-5fb8c835",
    "pendingTime" : "2016-11-19T16:32:11Z",
    "architecture" : "x86_64",
    "kernelId" : null,
    "ramdiskId" : null,
    "region" : "us-west-2"
}
```

**Verify the instance identity document**

If you intend to use the contents of the instance identity document for an important purpose, you should verify its contents and authenticity before using it.

The plaintext instance identity document is accompanied by three hashed and encrypted signatures. You can use these signatures to verify the origin and authenticity of the instance identity document and the information that it includes. The following signatures are provided:

- **Base64-encoded signature**—This is a base64-encoded SHA256 hash of the instance identity document that is encrypted using an RSA key pair.
- **PKCS7 signature**—This is a SHA1 hash of the instance identity document that is encrypted using a DSA key pair.
- **RSA-2048 signature**—This is a SHA256 hash of the instance identity document that is encrypted using an RSA-2048 key pair.

Each signature is available at a different endpoint in the instance metadata. You can use any one of these signatures depending on your hashing and encryption requirements. To verify the signatures, you must use the corresponding AWS public certificate.

**Important**

To validate the instance identity document using the base64-encoded signature or RSA2048 signature, you must request the corresponding AWS public certificate from [AWS Support](https://aws.amazon.com/support/).
The following topics provide detailed steps for validating the instance identity document using each signature.

- Use the PKCS7 signature to verify the instance identity document (p. 606)
- Use the base64-encoded signature to verify the instance identity document (p. 610)
- Use the RSA-2048 signature to verify the instance identity document (p. 613)

Use the PKCS7 signature to verify the instance identity document

This topic explains how to verify the instance identity document using the PKCS7 signature and the AWS DSA public certificate.

Prerequisites

This procedure requires the System.Security Microsoft .NET Core class. To add the class to your PowerShell session, run the following command.

```
PS C:\> Add-Type -AssemblyName System.Security
```

**Note**

The command adds the class to the current PowerShell session only. If you start a new session, you must run the command again.

To verify the instance identity document using the PKCS7 signature and the AWS DSA public certificate

1. Connect to the instance.
2. Retrieve the PKCS7 signature from the instance metadata, convert it to a byte array, and add it to a variable named `$Signature`. Use one of the following commands depending on the IMDS version used by the instance.

   **IMDSv2**

   ```
   ```

   **IMDSv1**

   ```
   ```

3. Retrieve the plaintext instance identity document from the instance metadata, convert it to a byte array, and add it to a variable named `$Document`. Use one of the following commands depending on the IMDS version used by the instance.

   **IMDSv2**

   ```
   ```

---

606
IMDSv1


4. Create a new file named certificate.pem and add one of the following AWS DSA public certificates, depending on your Region.

Other AWS Regions

The following AWS public certificate is for all AWS Regions, except Hong Kong, Bahrain, China, and GovCloud.

```
-----BEGIN CERTIFICATE-----
MIIC77CCAgqCCQCCQwkJ25V4a2zAJBgcqhkjOAQADMFwxCzAJBgnVBAYTA1vTMRkw
FwYDVQQIEhBXYXNoaW5nd9uIFN0YXR1MRawgDyVQHwZTwF0dGlMsAwHgYD
VQQSEkJ1WBF6b24gV2V1IFN1cn2y2zIEXMQz2FwXjMnIDMnIjMrMjIjF0eVoW0
NTAYMDMwMjJ1MjFMFWZXCA2AJBgnVBAYTA1vTMRkwFwYDVQQIEhBXYXNoaW5nd9u
IFN0YXR1MRawgDyVQHwZTwF0dGlMsAwHgYD

Other AWS Regions

Hong Kong Region

The AWS public certificate for the Hong Kong Region is as follows.

```
-----BEGIN CERTIFICATE-----
MIIC77CCAgqCCQCCQwkJ25V4a2zAJBgcqhkjOAQADMFwxCzAJBgnVBAYTA1vTMRkw
FwYDVQQIEhBXYXNoaW5nd9uIFN0YXR1MRawgDyVQHwZTwF0dGlMsAwHgYD
VQQSEkJ1WBF6b24gV2V1IFN1cn2y2zIEXMQz2FwXjMnIDMnIjMrMjIjF0eVoW0
NTAYMDMwMjJ1MjFMFWZXCA2AJBgnVBAYTA1vTMRkwFwYDVQQIEhBXYXNoaW5nd9u
IFN0YXR1MRawgDyVQHwZTwF0dGlMsAwHgYD

Bahrain Region

The AWS public certificate for the Bahrain Region is as follows.

```
-----BEGIN CERTIFICATE-----
Capetown Region

The AWS public certificate for the Cape Town Region is as follows.

-----BEGIN CERTIFICATE-----
MIIC7DCCAq4CCCMQwXgZDQYBDAgHHRwExaDQYJKoZIhvcNAQELBzBOB2RJFQ0r
X2tXawnvD2Jj6d9Gh6qX0N2JmKZj8d7FjOhDZ79+Q0e3w84kX86E4b2V33e
7bh2Y//A4R2uGIZm1rXyjOgbcJk5+Cb14n9Xc4Pd123OJfYXwQQcKOaJnG
2ukxJyBD8vW9hHqtNN43qEBsQpD4e/gf50GprEJrH5UJyj94/2j0Fb+J
3U0K3/wK99W38k01Q6mRe3Z7H0968C178vG0CzuIw9/9LmZJ2/9
-----END CERTIFICATE-----

Milan Region

The AWS public certificate for the Milan Region is as follows.

-----BEGIN CERTIFICATE-----
MIIC7DCCaqwCCCMQwXgZDQYBDAgHHRwExaDQYJKoZIhvcNAQELBzBOB2RJFQ0r
X2tXawnvD2Jj6d9Gh6qX0N2JmKZj8d7FjOhDZ79+Q0e3w84kX86E4b2V33e
7bh2Y//A4R2uGIZm1rXyjOgbcJk5+Cb14n9Xc4Pd123OJfYXwQQcKOaJnG
2ukxJyBD8vW9hHqtNN43qEBsQpD4e/gf50GprEJrH5UJyj94/2j0Fb+J
3U0K3/wK99W38k01Q6mRe3Z7H0968C178vG0CzuIw9/9LmZJ2/9
-----END CERTIFICATE-----
China Regions

The AWS public certificate for the China (Beijing) and China (Ningxia) Regions is as follows.

```
-----BEGIN CERTIFICATE-----
MIIDNjCCAh4CCQD3yZ1w1AVkTzAkgqkhiG9w0BAqfADBeCMqGwCyDQbQGEBWJ
UzEZMBGCAgAUECBMQVzFzaGlzQr3Vrb1TgdFQzE6MQA1UEBwMUH2VhHmRzTEg
MB4AUGEChMQWihem9uIFdIbiBTZXJ2aWNIcyBMTMEwIBcNMToWkNEZdMKdIOTE1
WhgPMkJ5ENDEwMTYOTU5MTVAMFxwCzAJBgNVBAYTAlVTMRkwFwYDVQQIEBQXYYXNo
aw5dNg9uIFN0YXR1MDAwGyDQbQGHwTZWFD0gxiMSAWhgDQbQHEkdBWFQ5bWrh
Z2V1IFhIcn2pyYXVzIHBXMQV4CZASi1dDQJkOcvhNAQEBBAADgEpFAIAQAcQogEB
AMVk9yvppSuD3AaAXyy2bVkeK3FlUqHpMyerii+i+NTs8tQqtNloaqCqhto/lg
lw9+QSnEwEyNmvi:WOBdn9CyDpN7cphVMeGgNJNI2vImMyWe12f2Kq/B9I7s7NC
PZ12z/t99orlck1nziZ08xFp17MTlshQwU3oxBeQ6A8wDucjXJHtGchuuip1uJ31
jxTPk3ztxp/xsF8BbAmwPnHbBghw/8lHe4393sMwL7v0Dvdn4y3bPQFwuw3r
vtBj/S5M/4/gQ3Xa1fCo1r90TzQbgx1i88R/gWtBegSgy7t2P2stU0yLdJ7hFzdK
/AIzAKh0vDWTWoadyo+0OaECaWGAATANBgkqhkiG9w0BAQsFAAOCAQEdsZn2+0E
VB1rJDF3PMWjHwrF17b1+1X/zaeW2hYEsreYxRLv+1VFp/L5l6KB7ETqHzUqteY7
zaACoerLu7/07oyNh42QcIhaaXLMN1cr6kcx0oObw+WQbX4cB3kekYHQ4RX
KHB1rwLXwa+2panSROIzJq25jhcFC9090D0TjlypV57n/Z9iQ+dqVfPncd3Bk
5pZJ1nDvVxqRike7B7f3tKjh7HpeF5m9hfrI1SvU+61mMVV/qJpnKysf9
c96en9a884BTJ2v4WxwS4q6h9fIVmFMq0p1ePG191dOwFxasNWQX0oaYysP
95yWLMKwa1MuA==
-----END CERTIFICATE-----
```

GovCloud Regions

The AWS public certificate for the AWS GovCloud Regions is as follows.

```
-----BEGIN CERTIFICATE-----
MIIC7TCCAq0CCQKfAMk9vyppSmDU3AaAXyy2bVkeK3FlUqHpMyerii+i+NTs8tQqtNloaqCqhto/lg
lw9+QSnEwEyNmvi:WOBdn9CyDpN7cphVMeGgNJNI2vImMyWe12f2Kq/B9I7s7NC
PZ12z/t99orlck1nziZ08xFp17MTlshQwU3oxBeQ6A8wDucjXJHtGchuuip1uJ31
jxTPk3ztxp/xsF8BbAmwPnHbBghw/8lHe4393sMwL7v0Dvdn4y3bPQFwuw3r
vtBj/S4M/4/gQ3Xa1fCo1r90TzQbgx1i88R/gWtBegSgy7t2P2stU0yLdJ7hFzdK
/AIzAKh0vDWTWoadyo+0OaECaWGAATANBgkqhkiG9w0BAQsFAAOCAQEdsZn2+0E
VB1rJDF3PMWjHwrF17b1+1X/zaeW2hYEsreYxRLv+1VFp/L5l6KB7ETqHzUqteY7
zaACoerLu7/07oyNh42QcIhaaXLMN1cr6kcx0oObw+WQbX4cB3kekYHQ4RX
KHB1rwLXwa+2panSROIzJq25jhcFC9090D0TjlypV57n/Z9iQ+dqVfPncd3Bk
5pZJ1nDvVxqRike7B7f3tKjh7HpeF5m9hfrI1SvU+61mMVV/qJpnKysf9
c96en9a884BTJ2v4WxwS4q6h9fIVmFMq0p1ePG191dOwFxasNWQX0oaYysP
95yWLMKwa1MuA==
-----END CERTIFICATE-----
```

5. Extract the certificate from the certificate file and store it in a variable named $Store.

```
```

6. Verify the signature.

```
```
If the signature is valid, the command returns no output. If the signature cannot be verified, the command returns Exception calling "CheckSignature" with "2" argument(s): "Cannot find the original signer. If your signature cannot be verified, contact AWS Support.

7. Validate the content of the instance identity document.

```powershell
PS C:\> [System.Linq.Enumerable]::SequenceEqual($SignatureDocument.ContentInfo.Content, $Document)
```

If the content of the instance identity document is valid, the command returns True. If instance identity document cannot be validated, contact AWS Support.

Use the base64-encoded signature to verify the instance identity document

This topic explains how to verify the instance identity document using the base64-encoded signature and the AWS RSA public certificate.

To validate the instance identity document using the base64-encoded signature and the AWS RSA public certificate

1. Connect to the instance.
2. Retrieve the base64-encoded signature from the instance metadata, convert it to a byte array, and add it to variable named $Signature. Use one of the following commands depending on the IMDS version used by the instance.

**IMDSv2**

```powershell
```

**IMDSv1**

```powershell
```

3. Retrieve the plaintext instance identity document from the instance metadata, convert it to a byte array, and add it to a variable named $Document. Use one of the following commands depending on the IMDS version used by the instance.

**IMDSv2**

```powershell
```
IMDSv1


4. Add one of the following AWS RSA public certificates to a new file named certificate.pem, depending on the Region of your instance.

Other AWS Regions

The following AWS public certificate is for all AWS Regions, except Hong Kong, Bahrain, China, and GovCloud.

-----BEGIN CERTIFICATE-----
MIIDjCCouagAwIBAgIJAJKAN4EDMN/FMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCA8+CjV7aG4+dD/cb16f7u4BxV2V22CIyoaZfSg1iY1lujSEqVJnInBOaj7nAFQyXqgV+iIjWSSAVaCwHHEwBwyk/M6mF5jBDuPwLYl7d0YxQ0aD2VwLkT3Afw4wCp7cV35ENNj5no3am7v8lyXEk8AS6Pv3b8f79/6aUG57vJwU6RwG00N621cHrJZuG6WVrBQCIU5A6+W4e+KjTPP1kDzXz66rE1vH0+g8YcBh71e2N7ZMc2aKQlYxGxWV3c52pyV7G8HxiDy8zjYkD+5T8G0oR0aX5z1Qo9C45Lh389d1a3QVZAhWkZw7nH7bSwUsMVv7Q8K0B1886T5y2z5p2Xh9Bp49+S2sE7WzQlVoewfK coacheswO6Fus96FlVjYn5Q+V5yGIYcJ820+eKpReJkoG8h976h4fHfJfnV7JgL6E5Pju1a1AFA85LM3dH6UZ69xQZPZZfECv+HdGDK5x6hZ8M6VFK+Jw==
-----END CERTIFICATE-----

Hong Kong Region

The AWS public certificate for the Hong Kong Region is as follows.

-----BEGIN CERTIFICATE-----
MIIEjCCabAgAwIBAgIJAJKAN4EDMN/FMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQwEjHZI0u20r7azw/hXyPy827RqIrRj5z9yXX9WTw5+1qgk4+8mU6fucTopJ55w6ocERsHx6CNgO6B5iA9O3p7n1cJSqebZfWn+PdiSqwG95hUYrsMT+jQ4YQp6f3+4Qe691Y6/paP0d2YqODyOB3wH4rF4QGf9ZlJ+X6b2fG8nuh5PvflEB2e7jx7nCjDj0bM69ypfT0Z24XuACLk8BvXH6Es4Zr2Es4/b1ySHZx3dy5GzF9Cz5iaIrSBEiD0obGd9GzD+K51f4x67FG8L5+5wu56v855S7Mjaj/hyXUEKx6+MVZ+i+sR+/Cg==
-----END CERTIFICATE-----

Bahrain Region

The AWS public certificate for the Bahrain Region is as follows.

-----BEGIN CERTIFICATE-----
MIIEjCCabAgAwIBAgIJAJKAN4EDMN/FMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQwEjHZI0u20r7azw/hXyPy827RqIrRj5z9yXX9WTw5+1qgk4+8mU6fucTopJ55w6ocERsHx6CNgO6B5iA9O3p7n1cJSqebZfWn+PdiSqwG95hUYrsMT+jQ4YQp6f3+4Qe691Y6/paP0d2YqODyOB3wH4rF4QGf9ZlJ+X6b2fG8nuh5PvflEB2e7jx7nCjDj0bM69ypfT0Z24XuACLk8BvXH6Es4Zr2Es4/b1ySHZx3dy5GzF9Cz5iaIrSBEiD0obGd9GzD+K51f4x67FG8L5+5wu56v855S7Mjaj/hyXUEKx6+MVZ+i+sR+/Cg==
-----END CERTIFICATE-----
Cape Town Region

The AWS public certificate for the Cape Town Region is as follows.

-----BEGIN CERTIFICATE-----
MIICNjCCAz+gAwIBAgIJAKumfZiRrNvHMA0GCSqGSIb3DQEBCuAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTExMjM5MzUzMzY3OVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAoTF0FtYXpvbmF3cy5jb20wIzEaMBgGA1UdEwEB/wQFAwQGMBQwJswwLjEYDQYJKoZIhvcNAQEBBQADggEPADCCAQoCAQEE...
-----END CERTIFICATE-----

Milan Region

The AWS public certificate for the Milan Region is as follows.

-----BEGIN CERTIFICATE-----
MIICNjCCAz+gAwIBAgIJAOZ3GEIaDcugMA0GCSqGSIb3DQEBCuAMFwxCzAJBgNVBAYTAlVTMRkwFwYDVQQIExBXYXNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0dGxlMSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnZpY2VzIExMQzAgFw0xOTExMjQxNTIyMDUyNzUxOVowXDELMAkGA1UEBhMCVVMxGTAXBgNVBAoTF0FtYXpvbmF3cy5jb20wIzEaMBgGA1UdEwEB/wQFAwQGMBQwJswwLjEYDQYJKoZIhvcNAQEBBQADggEPADCCAQoCAQEE...
-----END CERTIFICATE-----

China Regions

The AWS public certificate for the China (Beijing) and China (Ningxia) Regions is as follows.

-----BEGIN CERTIFICATE-----
MIICsCAQCBCCQUu97teKrR4zANBkgkqkhki9g0wDBQUFFADBqMgswCQYDVQQGEwJV...
-----END CERTIFICATE-----
GovCloud Regions

The AWS public certificate for the AWS GovCloud Regions is as follows.

```
-----BEGIN CERTIFICATE-----
MIIDCzCCAnSgAwIBAgIJAIe9Hnq82O7UMA0GCSqGSIb3DQEBCwUwYW1hem9uLmNvbSBJbmMuMRwj
……
-----END CERTIFICATE-----
```

5. Verify the instance identity document.

```
```

If the signature is valid, the command returns True. If the signature cannot be verified, contact AWS Support.

Use the RSA-2048 signature to verify the instance identity document

This topic explains how to verify the instance identity document using the RSA-2048 signature and the AWS RSA-2048 public certificate.

Prerequisites

This procedure requires the System.Security.Cryptography.NetCore class. To add the class to your PowerShell session, run the following command.

```
PS C:\> Add-Type -AssemblyName System.Security
```
To verify the instance identity document using the RSA-2048 signature and the AWS RSA-2048 public certificate

1. Connect to the instance.
2. Retrieve the RSA-2048 signature from the instance metadata, convert it to a byte array, and add it to a variable named $Signature. Use one of the following commands depending on the IMDS version used by the instance.

**IMDSv2**

```powershell
PS C:\>
```

**IMDSv1**

```powershell
```

3. Retrieve the plaintext instance identity document from the instance metadata, convert it to a byte array, and add it to a variable named $Document. Use one of the following commands depending on the IMDS version used by the instance.

**IMDSv2**

```powershell
```

**IMDSv1**

```powershell
```

4. Create a new file named `certificate.pem` and add one of the following AWS RSA-2048 public certificates, depending on your Region.

**North America Regions**

- **Northern Virginia**

```plaintext
-----BEGIN CERTIFICATE-----
MIIEEjCCAvqgAwIBAgIJA6ojcLp8pEAVwAGCQGSM49BAMFBwYDVQQD...XELw4NvBAGT

-----END CERTIFICATE-----
```
Instance metadata and user data

Ohio

Oregon

Amazon Elastic Compute Cloud
User Guide for Windows Instances
Instance metadata and user data
Amazon Elastic Compute Cloud
User Guide for Windows Instances
Instance metadata and user data

• Northern California

-----BEGIN CERTIFICATE-----
MIIEjCICAqvgAwIBAgIJAJNNPKIpcyRTiMA0GCSqGSIb3DQEBCwUAaBIGA1UdEwEB/wQFMAoGA1UdIwQx
-----END CERTIFICATE-----

• Canada (Central)

-----BEGIN CERTIFICATE-----
MID0zCCAjogAwIBAgIJAJNNKJhawkJaJ0uYMA0GCSqGSIb3DQEBCwUAaBIGA1UdEwEB/wQDAgKCC
-----END CERTIFICATE-----

South America Regions

• São Paulo

-----BEGIN CERTIFICATE-----
MIIEjCICAqvgAwIBAgIJAJNNPKIpcyRTiMA0GCSqGSIb3DQEBCwUAaBIGA1UdEwEB/wQFMAoGA1UdIwQx
-----END CERTIFICATE-----
Asia Pacific Regions

- Sydney

- Tokyo
Instance metadata and user data

---

**Seoul**

---BEGIN CERTIFICATE---

MIID0zCAICAgIBAgJIANgCgCht0jOHAoGCSqGSIb3DQEBCwUCAUJBGNV
BAYTALVMRkwY2VydQVEQIX6ZXYNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
50wxkIGAwOgCIASIkJBNMRA1wDQYJKoZIhvcNAQELBQADggEAAM0GCSqGSIb3DQEBC
---END CERTIFICATE---

• Osaka

---BEGIN CERTIFICATE---

MIID0zCAICAgIBAgJIANgCgCht0jOHAoGCSqGSIb3DQEBCwUCAUJBGNV
BAYTALVMRkwY2VydQVEQIX6ZXYNoaW5ndG9uIFN0YXRlMRAwDgYDVQQHEwdTZWF0
50wxkIGAwOgCIASIkJBNMRA1wDQYJKoZIhvcNAQELBQADggEAAM0GCSqGSIb3DQEBC
---END CERTIFICATE---
Amazon Elastic Compute Cloud
User Guide for Windows Instances
Instance metadata and user data

- Mumbai

-----BEGIN CERTIFICATE-----
MIID0zCCAmIgAwIBAgIJA8zP4YDIT7tmcCMAoGCSqGSIb3DQEBCwUAUxIFAgEAMl4f
BAYTA1VMRkwfDVQYDQYJKoZIhvcNAQELBQAD应该是数字证书的内容，因为它是加密的。”

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

- Hong Kong

-----BEGIN CERTIFICATE-----
MIID0zCCAmIgAwIBAgIJA8zP4YDIT7tmcCMAoGCSqGSIb3DQEBCwUAUxIFAgEAMl4f
BAYTA1VMRkwfDVQYDQYJKoZIhvcNAQELBQAD应该是数字证书的内容，因为它是加密的。”

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

- Singapore

-----BEGIN CERTIFICATE-----
MIIF4TCCAvwgAwIBAgIJA8zP4YDIT7tmcCMAoGCSqGSIb3DQEBCwUAUxIFAgEAMl4f
BAYTA1VMRkwfDVQYDQYJKoZIhvcNAQELBQAD应该是数字证书的内容，因为它是加密的。”

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

622
• Ningxia

-----BEGIN CERTIFICATE-----
MIIDOzCQAigAwIABAgIJAo4usxY3B1ZmaO0GCSqGSIb3DQEBCwUMFxw2aCAJBJqN
BAYTAlVTm8kwFyDVQQExBBXZXNoaW5ndG9uMgYDVQQHMQYDVQQIExBXYXNoaW5nd
GA1UdE8QH/4QIwIBADCCADF8AwIBAgIi8SgXgBAAECAgYEAMgq+UzCMA0GCSqGSIb
3DQEBCwUAA4IBAhgZPrh579MbH84FbRud2s2Q/yE/T6WrdjY3o9Gf74y/Xc37D5Qs
Pa3SPuMBIiAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIB
AAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAI
-----END CERTIFICATE-----

• Beijing

-----BEGIN CERTIFICATE-----
MIIDOzCQAigAwIABAgIJAo4usxY3B1ZmaO0GCSqGSIb3DQEBCwUMFxw2aCAJBJqN
BAYTAlVTm8kwFyDVQQExBBXZXNoaW5ndG9uMgYDVQQHMQYDVQQIExBXYXNoaW5nd
GA1UdE8QH/4QIwIBADCCADF8AwIBAgIi8SgXgBAAECAgYEAMgq+UzCMA0GCSqGSIb
3DQEBCwUAA4IBAhgZPrh579MbH84FbRud2s2Q/yE/T6WrdjY3o9Gf74y/Xc37D5Qs
Pa3SPuMBIiAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIB
AAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAI
-----END CERTIFICATE-----

AWS GovCloud Regions

• AWS GovCloud (US-West) Region

-----BEGIN CERTIFICATE-----
MIIDOzCQAigAwIABAgIJAo4usxY3B1ZmaO0GCSqGSIb3DQEBCwUMFxw2aCAJBJqN
BAYTAlVTm8kwFyDVQQExBBXZXNoaW5ndG9uMgYDVQQHMQYDVQQIExBXYXNoaW5nd
GA1UdE8QH/4QIwIBADCCADF8AwIBAgIi8SgXgBAAECAgYEAMgq+UzCMA0GCSqGSIb
3DQEBCwUAA4IBAhgZPrh579MbH84FbRud2s2Q/yE/T6WrdjY3o9Gf74y/Xc37D5Qs
Pa3SPuMBIiAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIB
AAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAIBAAI
-----END CERTIFICATE-----
5. Extract the certificate from the certificate file and store it in a variable named $Store.

```
```

6. Verify the signature.

```
PS C:\> $SignatureDocument.Decode($Signature)
PS C:\> $SignatureDocument.CheckSignature($Store, $true)
```

If the signature is valid, the command returns no output. If the signature cannot be verified, the command returns

```
Exception calling "CheckSignature" with "2" argument(s):
"Cannot find the original signer.
```

If the signature cannot be verified, contact AWS Support.
7. Validate the content of the instance identity document.

```powershell
PS C:\> [Linq.Enumerable]::SequenceEqual($SignatureDocument.ContentInfo.Content, $Document)
```

If the content of the instance identity document is valid, the command returns True. If instance identity document cannot be validated, contact AWS Support.

**Best practices and recommendations for SQL Server clustering on EC2**

SQL Always On clustering offers high availability without the requirement for shared storage. The list of practices in this topic, in addition to the prerequisites listed at Prerequisites, Restrictions, and Recommendations for Always On availability groups, can help you get the best results when operating a SQL Server Always On cluster on AWS. The practices listed in this topic also offer a method to gather logs.

**Note**
When nodes are deployed in different Availability Zones, or in different subnets within the same zone, they should be treated as a multi-subnet cluster. Keep this in mind as you apply best practices and when you address possible failure scenarios.

**Contents**
- Assign IP addresses (p. 625)
- Cluster properties (p. 626)
- Cluster quorum votes and 50/50 splits in a multi-site cluster (p. 626)
- DNS registration (p. 626)
- Elastic Network Adapters (ENAs) (p. 627)
- Multi-site clusters and EC2 instance placement (p. 627)
- Instance type selection (p. 627)
- Assign elastic network interfaces and IPs to the instance (p. 627)
- Heartbeat network (p. 628)
- Configure the network adapter in the OS (p. 628)
- IPv6 (p. 628)
- Host record TTL for SQL Availability Group Listeners (p. 628)
- Logging (p. 629)
- NetBIOS over TCP (p. 629)
- NetFT Virtual Adapter (p. 629)
- Set possible owners (p. 629)
- Tune the failover thresholds (p. 630)
- Witness importance and Dynamic Quorum Architecture (p. 631)
- Troubleshoot (p. 631)

**Assign IP addresses**

Each cluster node should have one elastic network interface assigned that includes three private IP addresses on the subnet: a primary IP address, a cluster IP address, and an Availability Group IP address. The operating system (OS) should have the NIC configured for DHCP. It should not be set for a static IP
address because the IP addresses for the cluster IP and Availability Group will be handled virtually in the Failover Cluster Manager. The NIC can be configured for a static IP as long as it is configured to only use the primary IP of eth0. If the other IPs are assigned to the NIC, it can cause network drops for the instance during failover events.

When the network drops because the IPs are incorrectly assigned, or when there is a failover event or network failure, it is not uncommon to see the following event log entries at the time of failure.

```
Isatap interface isatap.{9468661C-0AEB-41BD-BB8C-1F85981D5482} is no longer active.
Isatap interface isatap.{9468661C-0AEB-41BD-BB8C-1F85981D5482} with address fe80::5efe:169.254.1.105 has been brought up.
```

Because these messages seem to describe network issues, it is easy to mistake the cause of the outage or failure as a network error. However, these errors describe a symptom, rather than cause, of the failure. ISATAP is a tunneling technology that uses IPv6 over IPv4. When the IPv4 connection fails, the ISATAP adapter also fails. When the network issues are resolved, these entries should no longer appear in the event logs. Alternately, you can eliminate network errors by safely disabling ISATAP with the following command.

```
netsh int ipv6 isatap set state disabled
```

When you run this command, the adapter is removed from Device Manager. This command should be run on all nodes. It does not impact the ability of the cluster to function. Instead, when the command has been run, ISATAP is no longer used. However, because this command might cause unknown impacts on other applications that leverage ISATAP, you should test it.

**Cluster properties**

To see the complete cluster configuration, run the following PowerShell command.

```
Get-Cluster | Format-List -Property *
```

**Cluster quorum votes and 50/50 splits in a multi-site cluster**

To learn how the cluster quorum works and what to expect if a failure occurs, see Understanding Cluster and Pool Quorum.

**DNS registration**

In Windows Server 2012, Failover Clustering, by default, attempts to register each DNS node under the cluster name. This is acceptable for applications that are aware the SQL target is configured for multi-site. However, when the client is not configured this way, it can result in timeouts, delays, and application errors due to attempts to connect to each individual node and failing on the inactive ones. To prevent these problems, the Cluster Resource parameter RegisterAllProvidersIp must be changed to 0. For more information, see RegisterAllProvidersIp Setting and Multi-subnet Clustered SQL + RegisterAllProvidersIp + SharePoint 2013.

The RegisterAllProvidersIp can be modified with the following PowerShell script.

```
Import-Module FailoverClusters
#cluster = (Get-ClusterResource | where {($_.ResourceType -eq "Network Name") -and
  ($_.OwnerGroup -ne "Cluster Group")}).Name
Get-ClusterResource $cluster | Set-ClusterParameter RegisterAllProvidersIP 0
```
Get-ClusterResource $cluster | Set-ClusterParameter HostRecordTTL 300
Stop-ClusterResource $cluster
Start-ClusterResource $cluster

In addition to setting the Cluster Resource parameter to 0, you must ensure that the cluster has permissions to modify the DNS entry for your cluster name.

1. Log into the Domain Controller (DC) for the domain, or a server that hosts the forward lookup zone for the domain.
2. Launch the DNS Management Console and locate the A record for the cluster.
3. Right-click the A record and choose Properties.
5. Choose Add.
6. Choose Object Types..., select the box for Computers, and choose OK.
7. Enter the name of the cluster resource object and choose Check name and OK if resolve.
8. Select the check box for Full Control.
9. Choose OK.

Elastic Network Adapters (ENAs)

AWS has identified known issues with some clustering workloads running on ENA driver version 1.2.3. We recommend upgrading to version 1.5.0 or later and adjusting settings on the NIC in the OS. For the latest versions, see Amazon ENA Driver Versions. The first setting, which applies to all systems, increases Receive Buffers, which can be done with the following example PowerShell command.

Set-NetAdapterAdvancedProperty -Name (Get-NetAdapter | Where-Object {$_._InterfaceDescription -like '*Elastic*'}).Name -DisplayName "Receive Buffers" -DisplayValue 8192

For instances with more than 16 vCPUs, we recommend preventing RSS from running on CPU 0.

Run the following command.

Set-NetAdapterRss -name (Get-NetAdapter | Where-Object {$_._InterfaceDescription -like '*Elastic*'}).Name -Baseprocessorgroup 0 -BaseProcessorNumber 1

Multi-site clusters and EC2 instance placement

Each cluster is considered a multi-site cluster. The EC2 service does not share IP addresses virtually. Each node must be in a unique subnet. Though not required, we recommend that each node also be in a unique Availability Zone.

Instance type selection

The type of instance recommended for Windows Server Failover Clustering depends on the workload. For production workloads, we recommend instances that support EBS optimization (p. 1344) and Enhanced networking (p. 960).

Assign elastic network interfaces and IPs to the instance

Each node in an EC2 cluster should have only one attached elastic network interface. The network interface should have a minimum of two assigned private IP addresses. However, for workloads that
use Availability Groups, such as SQL Always On, you must include an additional IP address for each Availability Group. The primary IP address is used for accessing and managing the server, the secondary IP address is used as the cluster IP address, and each additional IP address is assigned to Availability Groups, as needed.

**Heartbeat network**

Some Microsoft documentation recommends using a dedicated heartbeat network. However, this recommendation is not applicable to EC2. With EC2, while you can assign and use a second elastic network interface for the heartbeat network, it uses the same infrastructure and shares bandwidth with the primary network interface. Therefore, traffic within the infrastructure cannot be prioritized, and cannot benefit from a dedicated network interface.

**Configure the network adapter in the OS**

The NIC in the OS can keep using DHCP as long as the DNS servers that are being retrieved from the DHCP Options Set allow for the nodes to resolve each other. You can set the NIC to be configured statically. When completed, you then manually configure only the primary IP address for the elastic network interface. Failover Clustering manages and assigns additional IP addresses, as needed.

For all instance types, you can increase the maximum transmission unit (MTU) on the network adapter to 9001 to support Jumbo Frames. This configuration reduces fragmentation of packets wherever Jumbo Frames are supported. The following example shows how to use PowerShell to configure Jumbo Frames for an Elastic Network Adapter.

```powershell
Get-NetAdapter | Set-NetAdapterAdvancedProperty -DisplayName "MTU" -DisplayValue 9001
```

**IPv6**

Microsoft does not recommend disabling IPv6 in a Windows Cluster. While Failover Clustering works in an IPv4-only environment, Microsoft tests clusters with IPv6 enabled. See Failover Clustering and IPv6 in Windows Server 2012 R2 for details.

**Host record TTL for SQL Availability Group Listeners**

Set the host record TTL to 300 seconds instead of the default 20 minutes (1200 seconds). For legacy client comparability, set `RegisterAllProvidersIP` to 0 for SQL Availability Group Listeners. This is not required in all environments. These settings are important because some legacy client applications cannot use MultiSubnetFailover in their connection strings. See HostRecordTTL Setting for more information. When you change these settings, the Cluster Resource must be restarted. The Cluster Group for the listener stops when the Cluster Resource is restarted, so it must be started. If you do not start the Cluster Group, the Availability Group remains offline in a RESOLVING state. The following are example PowerShell scripts for changing the TTL and `RegisterAllProvidersIP` settings.

```powershell
Get-ClusterResource yourListenerName | Set-ClusterParameter RegisterAllProvidersIP 0
Get-ClusterResource yourListenerName | Set-ClusterParameter HostRecordTTL 300
Stop-ClusterResource yourListenerName
Start-ClusterResource yourListenerName
```
Start-ClusterGroup yourListenerGroupName

Logging

The default logging level for the cluster log is 3. To increase the detail of log information, set the logging level to 5. See Set-ClusterLog for more information about the PowerShell cmdlet.

Set-ClusterLog –Level 5

NetBIOS over TCP

On Windows Server 2012 R2, you can increase the speed of the failover process by disabling NetBIOS over TCP. This feature was removed from Windows Server 2016. You should test this procedure if you are using older operating systems in your environment. For more information, see Speeding Up Failover Tips-n-Tricks. The following is an example PowerShell command to disable NetBIOS over TCP.

Get-ClusterResource “Cluster IP Address” | Set-ClusterParameter EnableNetBIOS 0

NetFT Virtual Adapter

For Windows Server versions earlier than 2016 and non-Hyper-V workloads, Microsoft recommends you enable the NetFT Virtual Adapter Performance Filter on the adapter in the OS. When you enable the NetFT Virtual Adapter, internal cluster traffic is routed directly to the NetFT Virtual Adapter. For more information, see NetFT Virtual Adapter Performance Filter. You can enable NetFT Virtual Adapter by selecting the check box in the NIC properties, or by using the following PowerShell command.

Get-NetAdapter | Set-NetAdapterBinding –ComponentID ms_netftflt –Enable $true

Set possible owners

The Failover Cluster Manager can be configured so that each IP address specified on the Cluster Core Resources and Availability Group resources can be brought online only on the node to which the IP belongs. When the Failover Cluster Manager is not configured for this and a failure occurs, there will be some delay in failover as the cluster attempts to bring up the IPs on nodes that do not recognize the address. For more information, see SQL Server Manages Preferred and Possible Owner Properties for AlwaysOn Availability Group/Role.

Each resource in a cluster has a setting for Possible Owners. This setting tells the cluster which nodes are permitted to “online” a resource. Each node is running on a unique subnet in a VPC. Because EC2 cannot share IPs between instances, the IP resources in the cluster can be brought online only by specific nodes. By default, each IP address that is added to the cluster as a resource has every node listed as a Possible Owner. This does not result in failures. However, during expected and unexpected failures, you can see errors in the logs about conflicting IPs and failures to bring IPs online. These errors can be ignored. If you set the Possible Owner property, you can eliminate these errors entirely, and also prevent down time while the services are moved to another node.
Tune the failover thresholds

In Server 2012 R2, the network thresholds for the failover heartbeat network default to high values. See Tuning Failover Cluster Network Thresholds for details. This potentially unreliable configuration (for clusters with some distance between them) was addressed in Server 2016 with an increase in the number of heartbeats. It was discovered that clusters would fail over due to very brief transient network issues. The heartbeat network is maintained with UDP 3343, which is traditionally far less reliable than TCP and more prone to incomplete conversations. Although there are low-latency connections between AWS Availability Zones, there are still geographic separations with a number of "hops" separating resources. Within an Availability Zone, there may be some distance between clusters unless the customer is using Placement Groups or Dedicated Hosts. As a result, there is a higher possibility for heartbeat failure with UDP than with TCP-based heartbeats.

The only time a cluster should fail over is when there is a legitimate outage, such as a service or node that experiences a hard failover, as opposed to a few UDP packets lost in transit. To ensure legitimate outages, we recommend that you adjust the thresholds to match, or even exceed, the settings for Server 2016 listed in Tuning Failover Cluster Network Thresholds. You can change the settings with the following PowerShell commands.

```powershell
(get-cluster).SameSubnetThreshold = 10
(get-cluster).CrossSubnetThreshold = 20
```

When you set these values, unexpected failovers should be dramatically reduced. You can fine-tune these settings by increasing the delays between heartbeats. However, we recommend that you send the
heartbeats more frequently with greater thresholds. Setting these thresholds even higher ensures that failovers occur only for hard failover scenarios, with longer delays before failing over. You must decide how much down time is acceptable for your applications.

After increasing the `SameSubnetThreshold` or `CrossSubnetThreshold`, we recommend that you increase the `RouteHistoryLength` to double the higher of the two values. This ensures that there is sufficient logging for troubleshooting. You can set the `RouteHistoryLength` with the following PowerShell command.

```
(Get-Cluster).RouteHistoryLength = 20
```

**Witness importance and Dynamic Quorum Architecture**

There is a difference between Disk Witness and File Share Witness. Disk Witness keeps a backup of the cluster database while File Share Witness does not. Both add a vote to the cluster (p. 626). You can use Disk Witness if you use iSCSI-based storage. For more about witness options, see File Share witness vs Disk witness for local clusters.

**Troubleshoot**

If you experience unexpected failovers, first make sure that you are not experiencing networking, service, or infrastructure issues.

1. Check that your nodes are not experiencing network-related issues.
2. Check driver updates. If you are using outdated drivers on your instance, you should update them. Updating your drivers might address bugs and stability issues that might be present in your currently installed version.
3. Check for any possible resource bottlenecks that could cause an instance to become unresponsive, such as CPU and disk I/O. If the node cannot service requests, it might appear to be down by the cluster service.

**Upgrade 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**

- Perform an in-place upgrade (p. 632)
- Perform an automated upgrade (p. 636)
- Migrate to latest generation instance types (p. 642)
- Windows to Linux replatforming assistant for Microsoft SQL Server Databases (p. 648)
- Troubleshoot an upgrade (p. 655)
Perform 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. 519).

Automated upgrades

For steps on how to use AWS Systems Manager to automate the upgrade of your Windows Server 2008 R2 to Server 2012 R2 or from SQL Server 2008 R2 on Windows Server 2012 R2 to SQL Server 2016, see Upgrade Your End of Support Microsoft 2008 Workloads in AWS with Ease.

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 R2
  - Upgrade Options for Windows Server 2012
  - Upgrade Options for Windows Server 2012 R2
  - Upgrade and conversion options for Windows Server 2016
  - Upgrade and conversion options for Windows Server 2019
  - Windows Server Upgrade Center
- We recommend performing an operating system upgrade on instances with at least 2 vCPUs and 4GB of RAM. If needed, you can change the instance to a larger size of the same type (t2.small to t2.large, for example), perform the upgrade, and then resize it back to the original size. If you are required to retain the instance size, you can monitor the progress using the instance console screenshot (p. 1475). For more information, see Change the instance type (p. 231).
- 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 Amazon EBS Elastic Volumes (p. 1315).
- 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.
- Update to the latest drivers as described in Migrate to latest generation instance types (p. 642).
- 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. 524) 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.
Perform an 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 Create a custom Windows AMI (p. 37).

2. Ensure that your Windows Server instance is using the latest network drivers. See Upgrade PV drivers on Windows instances (p. 524) 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 or 2019, do the following on your instance before proceeding:
   a. Uninstall the EC2Config service. For more information, see Stop, restart, delete, or uninstall EC2Config (p. 496).
   b. Install the EC2Launch service. For more information, see Install the latest version of EC2Launch (p. 487).
   c. Install the AWS Systems Manager SSM Agent. For more information, see Working with SSM Agent in the AWS Systems Manager User Guide.

6. Create a new volume from a Windows Server installation media snapshot.
   a. In the left navigation pane, under Elastic Block Store, choose Snapshots. In the search bar filter, choose Public Snapshots.
   b. Add the Owner filter to the search bar and choose Amazon images.
   c. Add the Description filter and enter Windows. Select Enter.
   d. Select the snapshot that matches the system architecture and language preference you are upgrading to. For example, select Windows 2019 English Installation Media to upgrade to Windows Server 2019.
   e. Choose Actions, Create Volume.
   f. In the Create Volume dialog box, choose the Availability Zone that matches your Windows instance, and choose Create Volume.

7. In the Create Volume Request Succeeded message, choose the volume that you just created.

8. Choose Actions, Attach Volume.

9. In the Attach Volume dialog box, enter the instance ID of your Windows instance and choose Attach.

10. Make the new volume available for use by following the steps at Make an Amazon EBS volume available for use on Windows.

    Important
    Do not initialize the disk because doing so will delete the existing data.

11. In Windows PowerShell, switch to the new volume drive. Begin the upgrade by opening the installation media volume you attached to the instance.
   a. If you are upgrading to Windows Server 2016 or later, run the following:

```
./setup.exe /auto upgrade
```

   If you are upgrading to an earlier version of Windows Server, run the following:

```
Sources/setup.exe
```
Perform an in-place upgrade

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 2019, 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 [Troubleshoot an upgrade](p. 655).

**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 runs 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:

  ```powershell
  pnputil -i -a "C:\Program Files (x86)\Citrix\XenTools\*.inf"
  ```

- Installs the missing device by executing the following command:

  ```powershell
  C:\Temp\EC2DriverUtils.exe install "C:\Program Files (x86)\Citrix\XenTools\xevtchn.inf" ROOT\XENEVTCHN
  ```

- Automatically removes **UpgradeHelperService** when complete.
Perform 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, and next to the filter field, select Public Snapshots.
   b. Add the Owner filter and choose Amazon images.
   c. Add the Description filter and enter Windows. Press Enter.
   d. Select the snapshot that matches the system architecture of your instance. For example, Windows 2012 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, enter 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 (Windows 2012 R2 and earlier) or EC2Launch (Windows 2016 and later). For more information, see Install the latest version of EC2Config (p. 494) or Install the latest version of EC2Launch (p. 487).
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. If you upgraded on a Nitro-based instance, we recommend that you install or upgrade the NVME and ENA drivers. For more information, see Windows Server 2012 R2, Install or upgrade AWS NVMe drivers (p. 538), or Enabling Enhanced Networking on Windows.
6. Re-enable antivirus and anti-spyware software and firewalls.

Perform an automated upgrade


The Systems Manager Automation documents provide two upgrade paths:

- SQL Server 2008 R2 on Windows Server 2012 R2 to SQL Server 2016 using the Systems Manager document for Automation named AWSEC2-CloneInstanceAndUpgradeSQLServer

Related services

The following AWS services are used in the automated upgrade process:

- **AWS Systems Manager.** AWS Systems Manager is a powerful, unified interface for centrally managing your AWS resources. For more information, see the [AWS Systems Manager User Guide](https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-what-is-systems-manager.html).
- **AWS Systems Manager Agent (SSM Agent)** is Amazon software that can be installed and configured on an Amazon EC2 instance, an on-premises server, or a virtual machine (VM). SSM Agent makes it possible for Systems Manager to update, manage, and configure these resources. The agent processes requests from the Systems Manager service in the AWS Cloud, and then runs them as specified in the request. For more information, see [Working with SSM Agent](https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-working-with-ssm-agent.html) in the [AWS Systems Manager User Guide](https://docs.aws.amazon.com/systems-manager/latest/userguide/)
- **AWS Systems Manager SSM documents.** An SSM document defines the actions that Systems Manager performs on your managed instances. SSM documents use JavaScript Object Notation (JSON) or YAML, and they include steps and parameters that you specify. This topic uses two Systems Manager SSM documents for Automation. For more information, see [AWS Systems Manager Documents](https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-ssm-documents.html) in the [AWS Systems Manager User Guide](https://docs.aws.amazon.com/systems-manager/latest/userguide/).
Prerequisites

In order to automate your upgrade with AWS Systems Manager Automation documents, you must perform the following tasks:

• Create an IAM role with the specified IAM policies (p. 637) to allow Systems Manager to perform automation tasks on your Amazon EC2 instances and verify that you meet the prerequisites to use Systems Manager.
• Select the option for how you want the automation to be run (p. 637). The options for execution are Simple execution, Rate control, Multi-account and Region, and Manual execution.

Create IAM role with specified permissions

For steps on how to create an IAM role in order to allow AWS Systems Manager to access resources on your behalf, see Creating a Role to Delegate Permissions to an AWS Service in the IAM User Guide. This topic also contains information on how to verify that your account meets the prerequisites to use Systems Manager.

Select execution option

When you select Automation on the Systems Manager console, select Execute. After you select an Automation document, you are then prompted to choose an automation execution option. You choose from the following options. In the steps for the paths provided later in this topic, we use the Simple execution option.

Simple execution

Choose this option if you want to update a single instance but do not want to go through each automation step to audit the results. This option is explained in further detail in the upgrade steps that follow.

Rate control

Choose this option if you want to apply the upgrade to more than one instance. You define the following settings.

• Parameter

  This setting, which is also set in Multi-Account and Region settings, defines how your automation branches out.

• Targets

  Select the target to which you want to apply the automation. This setting is also set in Multi-Account and Region settings.

• Parameter Values

  Use the values defined in the automation document parameters.

• Resource Group

  In AWS, a resource is an entity you can work with. Examples include Amazon EC2 instances, AWS CloudFormation stacks, or Amazon S3 buckets. If you work with multiple resources, it might be useful to manage them as a group as opposed to moving from one AWS service to another for every task. In some cases, you may want to manage large numbers of related resources, such as EC2 instances that make up an application layer. In this case, you will likely need to perform bulk actions on these resources at one time.

• Tags
Tags help you categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This categorization is useful when you have many resources of the same type. You can quickly identify a specific resource using the assigned tags.

- **Rate Control**

  Rate Control is also set in Multi-Account and Region settings. When you set the rate control parameters, you define how many of your fleet to apply the automation to, either by target count or by percentage of the fleet.

- **Multi-Account and Region**

  In addition to the parameters specified under Rate Control that are also used in the Multi-Account and Region settings, there are two additional settings:

  - **Accounts and organizational units (OUs)**

    Specify multiple accounts on which you want to run the automation.

  - **AWS Regions**

    Specify multiple AWS Regions where you want to run the automation.

- **Manual execution**

  This option is similar to Simple execution, but allows you to step through each automation step and audit the results.

- **Upgrade paths**

  There are two upgrade paths, which use two different AWS Systems Manager Automation documents.

  - **AWSEC2-CloneInstanceAndUpgradeWindows**. This script creates an Amazon Machine Image (AMI) from a Windows Server 2008 R2, 2012 R2, or 2016 instance in your account and upgrades this AMI to a supported version of your choice (Windows Server 2012 R2, 2016, or 2019). This multi-step process can take up to two hours to complete.

    To upgrade your Windows Server 2008 R2 instance to Windows Server 2016 or 2019, an in-place upgrade is performed twice, first from Windows Server 2008 R2 to Windows Server 2012 R2, and then from Windows Server 2012 R2 to Windows Server 2016 or 2019. Directly upgrading Windows Server 2008 R2 to Windows Server 2016 or 2019 is not supported.

    In this workflow, the automation creates an AMI from the instance and then launches the new AMI in the subnet you provide. The automation workflow performs an in-place upgrade from Windows Server 2008 R2, 2012 R2 or 2016 to the selected version (Windows Server 2012 R2, 2016, or 2019). It also updates or installs the AWS drivers required by the upgraded instance. After the upgrade is complete, the workflow creates a new AMI and terminates the upgraded instance. If you upgrade from Windows Server 2008 R2 to Windows Server 2016 or 2019, the automation creates two AMIs because the in-place upgrade is performed twice.

  - **AWSEC2-CloneInstanceAndUpgradeSQLServer**. This script creates an AMI from an Amazon EC2 instance running SQL Server 2008 R2 SP3 in your account, and then upgrades the AMI to SQL Server 2016 SP2. This multi-step process can take up to two hours to complete.

    In this workflow, the automation creates an AMI from the instance and then launches the new AMI in the subnet you provide. The automation then performs an in-place upgrade of SQL Server 2008 R2 to SQL Server 2016 SP2. After the upgrade is complete, the automation creates a new AMI before terminating the upgraded instance.
Perform an automated upgrade

There are two AMIs included in the automated upgrade process:

- **Current running instance.** The first AMI is the current running instance, which is not upgraded. This AMI is used to launch another instance to run the in-place upgrade. When the process is complete, this AMI is deleted from your account, unless you specifically request to keep the original instance. This setting is handled by the parameter `KeepPreUpgradeImageBackUp` (default value is `false`, which means the AMI is deleted by default).

- **Upgraded AMI.** This AMI is the outcome of the automation process. The second AMI includes SQL Server 2016 SP2 instead of SQL Server 2008 R2.

The final result is one AMI, which is the upgraded instance of the AMI.

When the upgrade is complete, you can test your application functionality by launching the new AMI in your VPC. After testing, and before you perform another upgrade, schedule application downtime before completely switching to the upgraded instance.

**Steps for performing an automated upgrade**

**Upgrade paths**

- Upgrade Windows Server 2008 R2, 2012 R2, or 2016 to Windows Server 2012 R2, 2016, or 2019 (p. 639)
- Upgrade SQL Server 2008 R2 to SQL Server 2016 (p. 640)

**Upgrade Windows Server 2008 R2, 2012 R2, or 2016 to Windows Server 2012 R2, 2016, or 2019**

This upgrade path requires additional prerequisites to work successfully. These prerequisites can be found in the automation document details for `AWSEC2-CloneInstanceAndUpgradeWindows` in the AWS Systems Manager User Guide.

After you have verified the additional prerequisite tasks, follow these steps to upgrade your Windows 2008 R2 instance to Windows 2012 R2 by using the automation document on AWS Systems Manager.

1. Open Systems Manager from the AWS Management Console.
2. From the left navigation pane, choose Automation.
3. Choose Execute automation.
4. Search for the automation document called `AWSEC2-CloneInstanceAndUpgradeWindows`.
5. When the document name appears, select it. When you select it, the document details appear.
6. Select Next to input the parameters for this document. Leave Simple execution selected at the top of the page.
7. Enter the requested parameters based on the following guidance.

   - **InstanceID**
     
     **Type**: String
     
     (Required) The instance running Windows Server 2008 R2, 2012 R2, or 2016 with the SSM agent installed.

   - **InstanceProfile**
     
     **Type**: String
Perform an automated upgrade

(Required) The IAM instance profile. This is the IAM role used to perform the Systems Manager automation against the Amazon EC2 instance and AWS AMIs. For more information, see Create an IAM Instance Profile for Systems Manager in the AWS Systems Manager User Guide.

- **TargetWindowsVersion**
  
  **Type:** String
  
  (Required) Select the target Windows version.

- **SubnetId**
  
  **Type:** String
  
  (Required) This is the subnet for the upgrade process and where your source EC2 instance resides. Verify that the subnet has outbound connectivity to AWS services, including Amazon S3, and also to Microsoft (in order to download patches).

- **KeepPreUpgradedBackUp**
  
  **Type:** String
  
  (Optional) If this parameter is set to `true`, the automation retains the image created from the instance. The default setting is `false`.

- **RebootInstanceBeforeTakingImage**
  
  **Type:** String
  
  (Optional) The default is `false` (no reboot). If this parameter is set to `true`, Systems Manager reboots the instance before creating an AMI for the upgrade.

8. After you have entered the parameters, select **Execute**. When the automation begins, you can monitor the execution progress.

9. When the automation completes, you will see the AMI ID. You can launch the AMI to verify that the Windows OS is upgraded.

**Note**

It is not necessary for the automation to run all of the steps. The steps are conditional based on the behavior of the automation and instance. Systems Manager might skip some steps that are not required.

Additionally, some steps may time out. Systems Manager attempts to upgrade and install all of the latest patches. Sometimes, however, patches time out based on a definable timeout setting for the given step. When this happens, the Systems Manager automation continues to the next step to ensure that the internal OS is upgraded to the target Windows Server version.

10. After the automation completes, you can launch an Amazon EC2 instance using the AMI ID to review your upgrade. For more information about how to create an Amazon EC2 instance from an AWS AMI, see How do I launch an EC2 instance from a custom Amazon Machine Image (AMI)?

**Upgrade SQL Server 2008 R2 to SQL Server 2016**

This upgrade path requires additional prerequisites to work successfully. These prerequisites can be found in the automation document details for AWSEC2-CloneInstanceAndUpgradeSQLServer in the AWS Systems Manager User Guide.

After you have verified the additional prerequisite tasks, follow these steps to upgrade your SQL Server 2008 R2 database engine to SQL Server 2016 using the automation document on AWS Systems Manager.

1. If you haven't already, download the SQL Server 2016 .iso file and mount it to the source server.
2. After the .iso file is mounted, copy all of the component files and place them on any volume of your choice.

3. Take an EBS snapshot of the volume and copy the snapshot ID onto a clipboard for later use. For more information about creating an EBS snapshot, see Creating an EBS Snapshot in the Amazon Elastic Compute Cloud User Guide.

4. Attach the instance profile to the EC2 source instance. This allows Systems Manager to communicate with the EC2 instance and run commands on it after it is added to the AWS Systems Manager service. For this example, we named the role SSM-EC2-Profile-Role with the AmazonSSMManagedInstanceCore policy attached to the role. See Create an IAM Instance Profile for Systems Manager in the AWS Systems Manager User Guide.

5. In the AWS Systems Manager console, in the left navigation pane, choose Managed Instances. Verify that your EC2 instance is in the list of managed instance. If you don't see your instance after a few minutes, see Where Are My Instances? in the AWS Systems Manager User Guide.

6. In the left navigation pane, choose Automation.

7. Choose Execute automation.

8. Choose the button beside the AWSEC2-CloneInstanceAndUpgradeSQLServer SSM document, and then choose Next.

9. Ensure that the Simple execution option is selected.

10. Enter the requested parameters based on the following guidance.

   • InstanceId
     
     **Type:** String
     
     (Required) The instance running SQL Server 2008 R2 (or later).

   • IamInstanceProfile
     
     **Type:** String
     
     (Required) The IAM instance profile.

   • SnapshotId
     
     **Type:** String
     
     (Required) The Snapshot ID for SQL Server 2016 installation media.

   • SubnetId
     
     **Type:** String
     
     (Required) This is the subnet for the upgrade process and where your source EC2 instance resides. Verify that the subnet has outbound connectivity to AWS services, including Amazon S3, and also to Microsoft (in order to download patches).

   • KeepPreUpgradedBackUp
     
     **Type:** String
     
     (Optional) If this parameter is set to `true`, the automation retains the image created from the instance. The default setting is `false`.

   • RebootInstanceBeforeTakingImage
     
     **Type:** String
     
     (Optional) The default is `false` (no reboot). If this parameter is set to `true`, Systems Manager reboots the instance before creating an AMI for the upgrade.

11. After you have entered the parameters, choose Execute. When the automation begins, you can monitor the execution progress.
12. When **Execution status** shows **Success**, expand **Outputs** to view the AMI information. You can use the AMI ID to launch your SQL Server 2016 instance for the VPC of your choice.

13. Open the EC2 console. In the left navigation pane, choose **AMIs**. You should see the new AMI.

14. To verify that SQL Server 2016 has been successfully installed, choose the new AMI and choose **Launch**.

15. Choose the type of instance that you want for the AMI, the VPC and subnet that you want to deploy to, and the storage that you want to use. Because you're launching the new instance from an AMI, the volumes are presented to you as an option to include within the new EC2 instance you are launching. You can remove any of these volumes, or you can add volumes.

16. Add a tag to help you identify your instance.

17. Add the security group or groups to the instance.

18. Choose **Launch Instance**.

19. Choose the tag name for the instance and select **Connect** under the **Actions** dropdown.

20. Verify that SQL Server 2016 is the new database engine on the new instance.

---

### Migrate to latest generation instance types

The AWS Windows AMIs are configured with the default settings used by the Microsoft installation media, with some customizations. The customizations include drivers and configurations that support the latest generation instance types, which are instances built on the Nitro System (p. 146), such as an M5 or C5.

When migrating to Nitro-based (p. 146) instances, including bare metal instances, we recommend that you follow the steps in this topic in the following cases:

- If you are launching instances from custom Windows AMIs
- If you are launching instances from Windows AMIs provided by Amazon that were created before August 2018

For more information, see [Amazon EC2 Update — Additional Instance Types, Nitro System, and CPU Options](https://aws.amazon.com/documentation/ec2/instance-types/upgrade-windows/).  

**Note**  
The following migration procedures can be performed on Windows Server version 2008 R2 and later.

**Contents**

- Part 1: Install and upgrade AWS PV drivers (p. 643)  
- Part 2: Install and upgrade ENA (p. 644)  
- Part 3: Upgrade AWS NVMe drivers (p. 644)  
- Part 4: Update EC2Config and EC2Launch (p. 644)  
- Part 5: Install the serial port driver for bare metal instances (p. 645)  
- Part 6: Update power management settings (p. 645)  
- Part 7: Update Intel chipset drivers for new instance types (p. 645)  
- (Alternative) Upgrade the AWS PV, ENA, and NVMe drivers using AWS Systems Manager (p. 646)  
- Migrate to Xen instance types from Nitro instance types (p. 647)

**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) Upgrade the AWS PV, ENA, and NVMe drivers using AWS Systems Manager (p. 646), 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 an instance based on the Nitro System (p. 146), such as an M5 or C5.

You must use PowerShell version 3.0 or later to successfully perform the upgrade.

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: Install and upgrade 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, Windows Server 2016, or Windows Server 2019. For more information, see Upgrade PV drivers on Windows instances (p. 524).

To upgrade a Domain Controller, see Upgrade a domain controller (AWS PV upgrade) (p. 526).

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 package history** (p. 521).

**Part 2: Install and upgrade 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 Enable enhanced networking with the Elastic Network Adapter (ENA) on Windows instances (p. 961).

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 the `install.ps1` script as an administrator.

4. Check if your AMI has enaSupport activated. If not, continue by following the documentation at Enable enhanced networking with the Elastic Network Adapter (ENA) on Windows instances (p. 961).

**Part 3: Upgrade 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 `dpinst.exe`.
4. Open a PowerShell session and run the following command:

   ```powershell
   start rundll32.exe sppnp.dll,Sysprep_Generalize_Pnp -wait
   ```

   **Note**
   To apply the command, you must run the PowerShell session as an administrator. PowerShell (x86) versions will result in an error. This command only runs sysprep on the device drivers. It does not run the full sysprep preparation.

5. For Windows Server 2008 R2 and Windows Server 2012, 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 anytime after the device sysprep.

**Part 4: Update 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 and later 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 Install the latest version of EC2Config (p. 494).

**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 install.ps1 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 Configure a Windows instance using EC2Launch (p. 486).

**Part 5: Install 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: Update power management settings**

The following update to power management settings sets 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:

```cmd
powercfg /setacvalueindex 381b4222-f694-41f0-9685-ff5bb260df2e 7516b95f-f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbb2b7e 0
powercfg /setacvalueindex 8c5e7fda-e8bf-4a96-9a85-a6e23a8c035c 7516b95f-f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbb2b7e 0
powercfg /setacvalueindex a1841308-3541-4fab-bc81-f7155f20b4a 7516b95f-f776-4464-8c53-06167f40cc99 3c0bc021-c8a8-4e07-a973-6b14cbb2b7e 0
```

Part 7: Update Intel chipset drivers for new instance types

The u-6tb1.metal, u-9tb1.metal, and u-12tb1.metal instance types use hardware that requires chipset drivers that were not previously installed on Windows AMIs. If you are not using an instance launched from an Amazon-provided Windows AMI dated 2018.11.19 or later, you must install the drivers using the Intel Chipset INF Utility.

To install the chipset drivers

1. Download the chipset utility to the instance.
2. Extract the files.
3. Run `SetupChipset.exe`.
4. Accept the Intel software license agreement and install the chipset drivers.
5. Reboot the instance.

(Alternative) Upgrade 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 SSM Agent is not contactable, the document can perform an offline installation of the AWS drivers if explicitly requested.

Note

This procedure will fail on a domain controller. To update drivers on a domain controller, see Upgrade a domain controller (AWS PV upgrade) (p. 526).

To automatically upgrade the AWS PV, ENA, and NVMe drivers using AWS 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 Systems Manager endpoints (ssm.*).

- **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.

### Migrate to Xen instance types from Nitro instance types

The following procedure assumes that you are currently running on a Nitro-based instance type, such as M5 or C5, and that you are migrating to an instance based on the Xen System, such as M4 or C4. For instance type specifications, see Amazon EC2 Instance Types. Perform the following steps before the migration to avoid errors during the booting process.

1. AWS PV drivers must be installed and upgraded on a Nitro instance before you migrate to a Xen instance. For steps to install and upgrade AWS PV drivers, see Part 1: Install and upgrade AWS PV drivers (p. 643).
2. Update to the latest EC2Launch v2 version. See Migrate to EC2Launch v2 (p. 453) for steps.
3. Open a PowerShell session and run the following command as an administrator to sysprep the device drivers. Running sysprep ensures that early boot storage drivers required for booting on Xen instances are properly registered with Windows.

   **Note**
   Running the command using PowerShell (x86) versions will result in an error. This command adds only the boot-critical device drivers to the critical device database. It does not run the full sysprep preparation.

   ```
   Start-Process rundll32.exe sppnp.dll,Sysprep_Generalize_Pnp -wait
   ```
4. Perform the migration to a Xen instance type when the sysprep process completes.

Windows to Linux replatforming assistant for Microsoft SQL Server Databases

The Windows to Linux replatforming assistant for Microsoft SQL Server Databases service is a scripting tool. It helps you move existing Microsoft SQL Server workloads from a Windows to a Linux operating system. You can use the replatforming assistant with any Windows Server virtual machines (VMs) hosted in the cloud, or with on-premises environments running Microsoft SQL Server 2008 and later. The tool checks for common incompatibilities, exports databases from the Windows VM, and imports into an EC2 instance running Microsoft SQL Server 2017 on Ubuntu 16.04. The automated process results in a ready-to-use Linux VM configured with your selected SQL Server databases that can be used for experimenting and testing.

Contents

• Concepts (p. 648)
• Related services (p. 648)
• How Windows to Linux replatforming assistant for Microsoft SQL Server works (p. 649)
• Components (p. 649)
• Setting up (p. 649)
• Get started (p. 651)

Concepts

The following terminology and concepts are central to your understanding and use of the Windows to Linux replatforming assistant for Microsoft SQL Server Databases.

Backup

A Microsoft SQL Server backup copies data or log records from a Microsoft SQL Server database or its transaction log to a backup device, such as a disk. For more information, see Backup Overview (Microsoft SQL Server).

Restore

A logical and meaningful sequence for restoring a set of Microsoft SQL Server backups. For more information, see Restore and Recovery Overview (Microsoft SQL Server).

Replatform

A Microsoft SQL Server database can be replatformed from an EC2 Windows instance to an EC2 Linux instance running Microsoft SQL Server. It can also be replatformed to the VMware Cloud running Microsoft SQL Server Linux on AWS.

Related services

AWS Systems Manager (Systems Manager) gives you visibility and control of your infrastructure on AWS. The Windows to Linux replatforming assistant for Microsoft SQL Server Databases uses Systems Manager to move your Microsoft SQL databases to Microsoft SQL Server on EC2 Linux. For more information about Systems Manager, see the AWS Systems Manager User Guide.
How Windows to Linux replatforming assistant for Microsoft SQL Server works

Windows to Linux replatforming assistant for Microsoft SQL Server Databases allows you to migrate your Microsoft SQL Server databases from an on-premises environment or from an EC2 Windows instance to Microsoft SQL Server 2017 on EC2 Linux using backup and restore. For the destination EC2 Linux instance, you provide either the EC2 instance ID or the EC2 instance type with the subnet ID and EC2 Key Pair.

When you run the PowerShell script for the Windows to Linux replatforming assistant for Microsoft SQL Server Databases on the source Microsoft SQL Server databases, the Windows instance backs up the databases to an encrypted Amazon Simple Storage Service (S3) storage bucket. It then restores the backups to an existing Microsoft SQL Server on EC2 Linux instance, or it launches a new Microsoft SQL Server on EC2 Linux instance and restores the backups to the newly created instance. This process can be used to replatform your 2-tier databases running enterprise applications. It also enables you to replicate your database to Microsoft SQL Server on Linux to test the application while the source Microsoft SQL Server remains online. After testing, you can schedule application downtime and rerun the PowerShell backup script during your final cutover.

The entire replatforming process can also be automated and run unattended. You can run the Systems Manager SSM document AWSEC2-SQLServerDBRestore to import your existing database backup files into Microsoft SQL Server on EC2 Linux without using the PowerShell backup script.

Components

The Windows to Linux replatforming assistant for Microsoft SQL Server Databases script consists of two main components:

1. A PowerShell backup script, which backs up on-premises Microsoft SQL Server databases to an Amazon S3 storage bucket. It then invokes the SSM Automation document AWSEC2-SQLServerDBRestore to restore the backups to a Microsoft SQL Server on EC2 Linux instance.

2. An SSM Automation document named AWSEC2-SQLServerDBRestore, which restores database backups to Microsoft SQL Server on EC2 Linux. This automation restores Microsoft SQL Server database backups stored in Amazon S3 to Microsoft SQL Server 2017 running on an EC2 Linux instance. You can provide your own EC2 instance running Microsoft SQL Server 2017 Linux, or the automation launches and configures a new EC2 instance with Microsoft SQL Server 2017 on Ubuntu 16.04. The automation supports the restoration of full, differential, and transactional log backups, and accepts multiple database backup files. The automation automatically restores the most recent valid backup of each database in the files provided. For more information, see AWSEC2-SQLServerDBRestore.

Setting up

This section covers the steps necessary to run the Windows to Linux replatforming script.

Contents

- Prerequisites (p. 649)
- Prerequisites for replatforming to an existing EC2 instance (p. 650)

Prerequisites

In order to run the Windows to Linux replatforming assistant for Microsoft SQL Server Databases script, you must do the following:
1. **Install the AWS PowerShell module**

To install the AWS PowerShell module, follow the steps listed in Setting up the AWS Tools for PowerShell on a Windows-Based Computer. We recommend that you use PowerShell 3.0 or later for the backup script to work properly.

2. **Install the Windows to Linux replatforming assistant PowerShell backup script**

To run the Windows to Linux replatforming assistant, download the PowerShell backup script: `MigrateSQLServerToEC2Linux.ps1`.

3. **Add an AWS user profile to the AWS SDK store**

To add and configure the AWS user profile, see the steps listed in Managing Profiles in the AWS Tools for PowerShell User Guide. Set the following IAM policy for your user profile. You can also add these permissions as an inline policy under your AWS user account using the IAM console.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "ec2:RebootInstances",
            "ec2:DescribeInstanceStatus",
            "ec2:DescribeInstances",
            "ec2:CreateTags",
            "ec2:RunInstances",
            "ec2:DescribeImages",
            "iam:PassRole",
            "ssm:StartAutomationExecution",
            "ssm:DescribeInstanceInformation",
            "ssm:ListCommandInvocations",
            "ssm:ListCommands",
            "ssm:GetAutomationExecution",
            "ssm:GetCommandInvocation",
            "s3:PutEncryptionConfiguration",
            "s3:CreateBucket",
            "s3:ListBucket",
            "s3:PutObject",
            "s3:GetObject",
            "s3:DeleteObject",
            "s3:DeleteBucket"
         ],
         "Resource": "*"
      }
   ]
}
```

4. **Create an IAM instance profile role**

To create an IAM instance profile role in order to run Systems Manager on EC2 Linux, see the steps listed under Create an Instance Profile for Systems Manager in the AWS Systems Manager User Guide.

**Prerequisites for replatforming to an existing EC2 instance**

To replatform to an existing instance running Microsoft SQL Server 2017 on Linux, you must:

1. Configure the EC2 instance with an AWS Identity and Access Management (IAM) instance profile and attach the AmazonSSMManagedInstanceCore managed policy.
For information about creating an IAM instance profile for Systems Manager and attaching it to an instance, see the following topics in the AWS Systems Manager User Guide:

- Create an Instance Profile for Systems Manager
- Attach an IAM Instance Profile to an Amazon EC2 Instance

2. Verify that SSM Agent is installed on your EC2 instance. For more information, see Installing and Configuring SSM Agent on Windows Instances in the AWS Systems Manager User Guide.

3. Verify that the EC2 instance has enough free disk space to download and restore the Microsoft SQL Server backups.

Get started

This section contains the PowerShell parameter definitions and scripts for replatforming your databases. For more information about how to use PowerShell scripts, see PowerShell.

Topics

- Run the Windows to Linux replatforming assistant for Microsoft SQL Server script (p. 651)
- Parameters (p. 652)

Run the Windows to Linux replatforming assistant for Microsoft SQL Server script

The following common scenarios and example PowerShell scripts demonstrate how to replatform your Microsoft SQL Server databases using Windows to Linux replatforming assistant for Microsoft SQL Server Databases.

Important

The Windows to Linux Replatforming Assistant for Microsoft SQL Server Databases resets the SQL Server server administrator (SA) user password on the target instance every time that it is run. After the replatform process is complete, you must set your own SA user password before you can connect to the target SQL Server instance.

Syntax

The Windows to Linux replatforming assistant for Microsoft SQL Server Databases script adheres to the syntax shown in the following example.

```
PS C:\> C:\MigrateSQLServerToEC2Linux.ps1 [-SqlServerInstanceName <String>] [[-DBNames] <Object[]>] [-MigrateAllDBs] [PathForBackup] <String> [-SetSourceDBModeReadOnly] [-IAMInstanceProfileName] <String> [-AWSRegion] <String> [-EC2InstanceId] <String> [-EC2InstanceType] <String> [-EC2KeyPair] <String> [-SubnetId] <String> [-AWSProfileName] <String> [-AWSProfileLocation] <String> [-GeneratePresignedUrls] [<CommonParameters>]
```

Example 1: Move a database to an EC2 instance

The following example shows how to move a database named AdventureDB to an EC2 Microsoft SQL Server on Linux instance, with an instance ID of i-024689abcdef, from the Microsoft SQL Server instance named MSSQLSERVER. The backup directory to be used is D:\Backup and the AWS Region is us-east-2.

```
PS C:\> ./MigrateSQLServerToEC2Linux.ps1 - SQLServerInstanceName MSSQLSERVER - EC2InstanceId i-
```
Example 2: Move a database to an EC2 instance using the AWS credentials profile

The following example shows how to move the database in Example 1 using the AWS credentials profile:

```powershell
PS C:\> ./MigrateSQLServerToEC2Linux.ps1 - SQLServerInstanceName MSSQLSERVER -
024689abcdef -DBNames AdventureDB -PathForBackup D:\\Backup -AWSRegion us-east-2 -
IamInstanceProfileName AmazonSSMManagedInstanceCore
```

Example 3: Move a database to a new m5.large type instance

The following example shows how to create an m5.large type EC2 Linux instance in subnet-abc127 using the Key Pair customer-ec2-keypair and then moving AdventureDB and TestDB to the new instance from the database used in Examples 1 and 2.

```powershell
PS C:\> ./MigrateSQLServerToEC2Linux.ps1 -EC2InstanceType m5.large -SubnetId subnet-abc127
(customer-ec2-keypair -DBNames AdventureDB,TestDB -PathForBackup D:\\Backup -AWSRegion us-
east-2 -
AWSProfileName DBMigration -IamInstanceProfileName AmazonSSMManagedInstanceCore
```

Example 4: Move all databases to a new m5.large type instance

The following example shows how to create an m5.large type EC2 Linux instance in subnet-abc127 using the Key Pair customer-ec2-keypair and then migrating all databases to the instance from databases used in Examples 1 and 2.

```powershell
PS C:\> ./MigrateSQLServerToEC2Linux.ps1 -EC2InstanceType m5.large -SubnetId subnet-abc127
(customer-ec2-keypair -MigrateAllDBs -PathForBackup D:\\Backup -AWSRegion us-east-2 -
AWSProfileName DBMigration -IamInstanceProfileName AmazonSSMManagedInstanceCore
```

Parameters

The following parameters are used by the PowerShell script to replatform your Microsoft SQL Server databases.

```
-SqlServerInstanceName
```

The name of the Microsoft SQL Server instance to be backed up. If a value for SqlServerInstanceName is not provided, $env:ComputerName is used by default.

Type: String

Required: No

```
-DBNames
```

The names of the databases to be backed up and restored. Specify the names of the databases in a comma-separated list (for example, adventureDB,universityDB). Either the DBNames or MigrateAllDBs parameter is required.
Type: Object
Required: No

-MigrateAllDBs

This switch is disabled by default. If this switch is enabled, the automation migrates all databases except for the system databases (master, msdb, tempdb). Either the DBNames or MigrateAllDBs parameter is required.

Type: SwitchParameter
Required: No

-PathForBackup

The path where the full backup is stored.

Type: String
Required: Yes

-SetSourceDBModeReadOnly

This switch is disabled by default. If this switch is enabled, it makes the database read-only during migration.

Type: SwitchParameter
Required: No

-IamInstanceProfileName

Enter the AWS IAM instance role with permissions to run Systems Manager Automation on your behalf. See Getting Started with Automation in the AWS Systems Manager User Guide.

Type: String
Required: Yes

-AWSRegion

Enter the AWS Region where your Amazon S3 buckets are created to store database backups.

Type: String
Required: Yes

-EC2InstanceId

To restore Microsoft SQL Server databases to an existing EC2 instance running Microsoft SQL Server Linux, enter the instance ID of the instance. Make sure that the EC2 instance already has the AWS Systems Manager SSM Agent installed and running.

Type: String
Required: No

-EC2InstanceType
To restore Microsoft SQL Server databases to a new EC2 Linux instance, enter the instance type of the instance to be launched.

Type: String
Required: No

-EC2KeyPair

To restore Microsoft SQL Server databases to a new EC2 Linux instance, enter the name of the EC2 Key Pair to be used to access the instance. This parameter is recommended if you are creating a new EC2 Linux instance.

Type: String
Required: No

-SubnetId

This parameter is required when creating a new EC2 Linux instance. When creating a new EC2 Linux instance, if SubnetId is not provided, the AWS user default subnet is used to launch the EC2 Linux instance.

Type: String
Required: No

-AWSPortalName

The name of the AWS profile that the automation uses when connecting to AWS services. For more information on the required IAM user permissions, see Getting Started with Automation in the AWS Systems Manager User Guide. If a profile is not entered, the automation uses your default AWS profile.

Type: String
Required: No

-AWSPortalLocation

The location of the AWS Profile if the AWS Profile is not stored in the default location.

Type: String
Required: No

-GeneratePresignedUrls

This parameter is only used when replatforming to non-EC2 instances, such as to VMware Cloud on AWS or on-premises VMs.

Type: SwitchParameter
Required: No

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see About Common Parameters in the Microsoft PowerShell documentation.

Required: No
Troubleshoot 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. 1531).
- 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. 632).

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.

Inspect 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. 604).

Inspect 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-012345ABCDEF

For instances using SMBIOS 2.4, the UUID might be represented in little-endian format; for example:

45E12AEC-DCD1-B213-94ED-012345ABCDEF

Tutorial: Set 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 1: Create your security groups (p. 656)
- Step 2: Set up your Active Directory domain controller (p. 659)
- Step 3: Configure your head node (p. 660)
- Step 4: Set up the compute node (p. 661)
- Step 5: Scale your HPC compute nodes (optional) (p. 662)

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 your 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.

```powershell
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.

```powershell
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.

```powershell
PS C:\> $sg_dm = New-Object Amazon.EC2.Model.UserIdGroupPair
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"; UserIdGroupPairs= $sg_dm }
PS C:\> $r17 = @{ IpProtocol="TCP"; FromPort="3389"; ToPort="3389"; UserIdGroupPairs= $sg_dm }
PS C:\> $r18 = @{ IpProtocol="UDP"; FromPort="53"; ToPort="53"; IpRanges="203.0.113.25/32" }
PS C:\> $r19 = @{ 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.

```powershell
PS C:\> Grant-EC2SecurityGroupIngress -GroupId $g_id -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.

```powershell
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.

```powershell
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.

```powershell
#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 }
```
Step 2: Set up your Active Directory domain controller

6. Use the Grant-EC2SecurityGroupIngress cmdlet to add the rules to the HPC cluster security group.

```powershell
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" )
```

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.
11. Select the Elastic IP address you created, and choose Actions, Associate address. For Instance, choose the domain controller instance. Choose Associate.
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.
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:

   ```
   IPCConfig /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.
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.

```xml
<component name="Microsoft-Windows-UnattendedJoin" processorArchitecture="amd64"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
xmns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <Identification>
      <UnsecureJoin>false</UnsecureJoin>
```

662

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 and templates, 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.
EC2 Fleet and Spot Fleet

You can use an EC2 Fleet or a Spot Fleet to launch a fleet of instances. In a single API call, a fleet can launch multiple instance types across multiple Availability Zones, using the On-Demand Instance, Reserved Instance, and Spot Instance purchasing options together.

Topics
- EC2 Fleet (p. 664)
- Spot Fleet (p. 711)
- Monitor fleet events using Amazon EventBridge (p. 746)
- Tutorials for EC2 Fleet and Spot Fleet (p. 760)
- Example configurations for EC2 Fleet and Spot Fleet (p. 766)
- Fleet quotas (p. 791)

EC2 Fleet

An EC2 Fleet contains the configuration information to launch a fleet—or group—of instances. In a single API call, a fleet can launch multiple instance types across multiple Availability Zones, using the On-Demand Instance, Reserved Instance, and Spot Instance purchasing options together. Using EC2 Fleet, you can:

- Define separate On-Demand and Spot capacity targets and the maximum amount you’re willing to pay per hour
- Specify the instance types that work best for your applications
- Specify how Amazon EC2 should distribute your fleet capacity within each purchasing option

You can also set a maximum amount per hour that you're willing to pay for your fleet, and EC2 Fleet launches instances until it reaches the maximum amount. When the maximum amount you’re willing to pay is reached, the fleet stops launching instances even if it hasn't met the target capacity.

The EC2 Fleet attempts to launch the number of instances that are required to meet the target capacity specified in your request. If you specified a total maximum price per hour, it fulfills the capacity until it reaches the maximum amount that you’re willing to pay. The fleet can also attempt to maintain its target Spot capacity if your Spot Instances are interrupted. For more information, see How Spot Instances work (p. 287).
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. 665)
- Burstable performance instances (p. 665)
- EC2 Fleet request types (p. 666)
- EC2 Fleet configuration strategies (p. 684)
- Work with EC2 Fleets (p. 692)

EC2 Fleet limitations

The following limitations apply to EC2 Fleet:

- EC2 Fleet is available only through the API or AWS CLI.
- An EC2 Fleet request can't span AWS Regions. You need to create a separate EC2 Fleet for each Region.
- An EC2 Fleet request can't span different subnets from the same Availability Zone.

Burstable performance instances

If you launch your Spot Instances using a burstable performance instance type (p. 160), and if you plan to use your burstable performance Spot Instances immediately and for a short duration, with no idle time for accruing CPU credits, we recommend that you launch them in Standard mode (p. 175) to avoid paying higher costs. If you launch burstable performance Spot Instances in Unlimited mode (p. 168) and burst CPU immediately, you'll spend surplus credits for bursting. If you use the instance for a short duration, the instance doesn't have time to accrue CPU credits to pay down the surplus credits, and you are charged for the surplus credits when you terminate the instance.

Unlimited mode is suitable for burstable performance Spot Instances only if the instance runs long enough to accrue CPU credits for bursting. Otherwise, paying for surplus credits makes burstable performance Spot Instances more expensive than using other instances. For more information, see When to use unlimited mode versus fixed CPU (p. 169).

Launch credits are meant to provide a productive initial launch experience for T2 instances by providing sufficient compute resources to configure the instance. Repeated launches of T2 instances to access new launch credits is not permitted. If you require sustained CPU, you can earn credits (by idling over some period), use Unlimited mode (p. 168) for T2 Spot Instances, or use an instance type with dedicated CPU.
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. For more information, see `Use an EC2 Fleet of type 'instant'` (p. 666).

**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 capacity pools if capacity is unavailable.

**maintain**

(Default) If you configure the request type as `maintain`, EC2 Fleet places an asynchronous request for your desired capacity, and maintains capacity by automatically replenishing any interrupted Spot Instances.

All three types of requests benefit from an allocation strategy. For more information, see `Allocation strategies for Spot Instances` (p. 685).

**Use an EC2 Fleet of type 'instant'**

The EC2 Fleet of type `instant` is a synchronous one-time request that makes only one attempt to launch your desired capacity. The API response lists the instances that launched, along with errors for those instances that could not be launched. There are several benefits to using an EC2 Fleet of type `instant`, which are described in this article. Example configurations are provided at the end of the article.

For workloads that need a launch-only API to launch EC2 instances, you can use the RunInstances API. However, with RunInstances, you can only launch On-Demand Instances or Spot Instances, but not both in the same request. Furthermore, when you use RunInstances to launch Spot Instances, your Spot Instance request is limited to one instance type and one Availability Zone. This targets a single Spot capacity pool (a set of unused instances with the same instance type and Availability Zone). If the Spot capacity pool does not have sufficient Spot Instance capacity for your request, the RunInstances call fails.

Instead of using RunInstances to launch Spot Instances, we recommend that you rather use the CreateFleet API with the `type` parameter set to `instant` for the following benefits:

- **Launch On-Demand Instances and Spot Instances in one request.** An EC2 Fleet can launch On-Demand Instances, Spot Instances, or both. The request for Spot Instances is fulfilled if there is available capacity and the maximum price per hour for your request exceeds the Spot price.

- **Increase the availability of Spot Instances.** By using an EC2 Fleet of type `instant`, you can launch Spot Instances following Spot best practices with the resulting benefits:
  - **Spot best practice: Be flexible about instance types and Availability Zones.**
    
    Benefit: By specifying several instance types and Availability Zones, you increase the number of Spot capacity pools. This gives the Spot service a better chance of finding and allocating your desired Spot compute capacity. A good rule of thumb is to be flexible across at least 10 instance types for each workload and make sure that all Availability Zones are configured for use in your VPC.
  - **Spot best practice: Use the capacity-optimized allocation strategy.**
Benefit: The capacity-optimized allocation strategy automatically provisions instances from the most-available Spot capacity pools. Because your Spot Instance capacity is sourced from pools with optimal capacity, this decreases the possibility that your Spot Instances will be interrupted when Amazon EC2 needs the capacity back.

- **Get access to a wider set of capabilities.** For workloads that need a launch-only API, and where you prefer to manage the lifecycle of your instance rather than let EC2 Fleet manage it for you, use the EC2 Fleet of type `instant` instead of the `RunInstances` API. EC2 Fleet provides a wider set of capabilities than RunInstances, as demonstrated in the following examples. For all other workloads, you should use Amazon EC2 Auto Scaling because it supplies a more comprehensive feature set for a wide variety of workloads, like ELB-backed applications, containerized workloads, and queue processing jobs.

AWS services like Amazon EC2 Auto Scaling and Amazon EMR use EC2 Fleet of type `instant` to launch EC2 instances.

**Prerequisites for EC2 Fleet of type instant**

For the prerequisites for creating an EC2 Fleet, see [EC2 Fleet prerequisites](p. 693).

**How instant EC2 Fleet works**

When working with an EC2 Fleet of type `instant`, the sequence of events is as follows:

1. Configure the `CreateFleet` request type as `instant`. For more information, see [Create an EC2 Fleet](p. 701). Note that after you make the API call, you can't modify it.
2. When you make the API call, EC2 Fleet places a synchronous one-time request for your desired capacity.
3. The API response lists the instances that launched, along with errors for those instances that could not be launched.
4. You can describe your EC2 Fleet, list the instances associated with your EC2 Fleet, and view the history of your EC2 Fleet.
5. After your instances have launched, you can **delete the fleet request**. When deleting the fleet request, you can also choose to terminate the associated instances, or leave them running.
6. You can terminate the instances at any time.

**Examples**

The following examples show how to use EC2 Fleet of type `instant` for different use cases. For more information about using the EC2 CreateFleet API parameters, see [CreateFleet](p. 693) in the [Amazon EC2 API Reference](p. 693).

**Examples**

- Example 1: Launch Spot Instances with the capacity-optimized allocation strategy (p. 668)
- Example 2: Launch a single Spot Instance with the capacity-optimized allocation strategy (p. 669)
- Example 3: Launch Spot Instances using instance weighting (p. 670)
- Example 4: Launch Spot Instances within single Availability zone (p. 672)
- Example 5: Launch Spot Instances of single instance type within single Availability zone (p. 673)
- Example 6: Launch Spot Instances only if minimum target capacity can be launched (p. 674)
- Example 7: Launch Spot Instances only if minimum target capacity can be launched of same Instance Type in a single Availability Zone (p. 676)
- Example 8: Launch instances with multiple Launch Templates (p. 677)
- Example 9: Launch Spot Instance with a base of On-Demand Instances (p. 679)
Example 1: Launch Spot Instances with the capacity-optimized allocation strategy

The following example specifies the parameters required in an EC2 Fleet of type `instance`: a launch template, target capacity, default purchasing option, and launch template overrides.

- The launch template is identified by its launch template name and version number.
- The 12 launch template overrides specify 4 different instance types and 3 different subnets, each in a separate Availability Zone. Each instance type and subnet combination defines a Spot capacity pool, resulting in 12 Spot capacity pools.
- The target capacity for the fleet is 20 instances.
- The default purchasing option is `spot`, which results in the fleet attempting to launch 20 Spot Instances into the Spot capacity pool with optimal capacity for the number of instances that are launching.

```json
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized"
    },
    "LaunchTemplateConfig": [
        {
            "LaunchTemplateName": "ec2-fleet-lt1",
            "Version": "$Latest"
        },
        "Overrides": [
            {
                "InstanceType": "c5.large",
                "SubnetId": "subnet-fae8c380"
            },
            {
                "InstanceType": "c5.large",
                "SubnetId": "subnet-e7188bab"
            },
            {
                "InstanceType": "c5.large",
                "SubnetId": "subnet-49e41922"
            },
            {
                "InstanceType": "c5d.large",
                "SubnetId": "subnet-fae8c380"
            },
            {
                "InstanceType": "c5d.large",
                "SubnetId": "subnet-e7188bab"
            },
            {
                "InstanceType": "c5d.large",
                "SubnetId": "subnet-49e41922"
            },
            {
                "InstanceType": "m5.large",
                "SubnetId": "subnet-fae8c380"
            },
            {
                "InstanceType": "m5.large",
                "SubnetId": "subnet-e7188bab"
            },
            {
                "InstanceType": "m5.large",
                "SubnetId": "subnet-49e41922"
            }
        ]
    }
}
```
Example 2: Launch a single Spot Instance with the capacity-optimized allocation strategy

You can optimally launch one Spot Instance at a time by making multiple EC2 Fleet API calls of type instant, by setting the TotalTargetCapacity to 1.

The following example specifies the parameters required in an EC2 Fleet of type instant: a launch template, target capacity, default purchasing option, and launch template overrides. The launch template is identified by its launch template name and version number. The 12 launch template overrides have 4 different instance types and 3 different subnets, each in a separate Availability Zone. The target capacity for the fleet is 1 instance, and the default purchasing option is spot, which results in the fleet attempting to launch a Spot Instance from one of the 12 Spot capacity pools based on the capacity-optimized allocation strategy, to launch a Spot Instance from the most-available capacity pool.
Example 3: Launch Spot Instances 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 based on how many units of the workload can run on the instance assuming a unit of the workload requires a 15 GB of memory and 4 vCPUs. For example an m5.xlarge (4 vCPUs and 16 GB of memory) can run one unit and is weighted 1, m5.2xlarge (8 vCPUs and 32 GB of memory) can run 2 units and is weighted 2, and so on. The total target capacity is set to 40 units. The default purchasing option is spot, and the allocation strategy is capacity-optimized, which results in either 40 m5.xlarge (40 divided by 1), 20 m5.2xlarge (40 divided by 2), 10 m5.4xlarge (40 divided by 4), 5 m5.8xlarge (40 divided by 8), or a mix of the instance types with weights adding up to the desired capacity based on the capacity-optimized allocation strategy.

For more information, see EC2 Fleet instance weighting (p. 690).
"AllocationStrategy":"capacity-optimized"
},
"LaunchTemplateConfigs": [
{
  "LaunchTemplateSpecification":{
    "LaunchTemplateName":"ec2-fleet-lt1",
    "Version":"$Latest"
  },
  "Overrides": [
    {
      "InstanceType":"m5.xlarge",
      "SubnetId":"subnet-fae8c380",
      "WeightedCapacity":1
    },
    {
      "InstanceType":"m5.xlarge",
      "SubnetId":"subnet-e7188bab",
      "WeightedCapacity":1
    },
    {
      "InstanceType":"m5.xlarge",
      "SubnetId":"subnet-49e41922",
      "WeightedCapacity":1
    },
    {
      "InstanceType":"m5.2xlarge",
      "SubnetId":"subnet-fae8c380",
      "WeightedCapacity":2
    },
    {
      "InstanceType":"m5.2xlarge",
      "SubnetId":"subnet-e7188bab",
      "WeightedCapacity":2
    },
    {
      "InstanceType":"m5.2xlarge",
      "SubnetId":"subnet-49e41922",
      "WeightedCapacity":2
    },
    {
      "InstanceType":"m5.4xlarge",
      "SubnetId":"subnet-fae8c380",
      "WeightedCapacity":4
    },
    {
      "InstanceType":"m5.4xlarge",
      "SubnetId":"subnet-e7188bab",
      "WeightedCapacity":4
    },
    {
      "InstanceType":"m5.4xlarge",
      "SubnetId":"subnet-49e41922",
      "WeightedCapacity":4
    },
    {
      "InstanceType":"m5.8xlarge",
      "SubnetId":"subnet-fae8c380",
      "WeightedCapacity":8
    },
    {
      "InstanceType":"m5.8xlarge",
      "SubnetId":"subnet-e7188bab",
      "WeightedCapacity":8
    },
    {
      "InstanceType":"m5.8xlarge",
      "SubnetId":null,
      "WeightedCapacity":8
    }
  ]
}
Example 4: Launch Spot Instances within single Availability zone

You can configure a fleet to launch all instances in a single Availability Zone by setting the Spot options SingleAvailabilityZone to true.

The 12 launch template overrides have different instance types and subnets (each in a separate Availability Zone) but the same weighted capacity. The total target capacity is 20 instances, the default purchasing option is spot, and the Spot allocation strategy is capacity-optimized. The EC2 Fleet launches 20 Spot Instances all in a single AZ, from the Spot capacity pool(s) with optimal capacity using the launch specifications.

```json
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleAvailabilityZone": true
    },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "ec2-fleet-lt1",
                "Version": "$Latest"
            },
            "Overrides": [
                { "InstanceType": "c5.4xlarge", "SubnetId": "subnet-fae8c380" },
                { "InstanceType": "c5.4xlarge", "SubnetId": "subnet-e7188bab" },
                { "InstanceType": "c5.4xlarge", "SubnetId": "subnet-49e41922" },
                { "InstanceType": "c5d.4xlarge", "SubnetId": "subnet-fae8c380" },
                { "InstanceType": "c5d.4xlarge", "SubnetId": "subnet-e7188bab" },
                { "InstanceType": "c5d.4xlarge", "SubnetId": "subnet-49e41922" },
                { "InstanceType": "m5.4xlarge", "SubnetId": "subnet-fae8c380" }
            ]
        }...
}
```
Example 5: Launch Spot Instances of single instance type within single Availability zone

You can configure a fleet to launch all instances of the same instance type and in a single Availability Zone by setting the SpotOptions SingleInstanceType to true and SingleAvailabilityZone to true.

The 12 launch template overrides have different instance types and subnets (each in a separate Availability Zone) but the same weighted capacity. The total target capacity is 20 instances, the default purchasing option is spot, the Spot allocation strategy is capacity-optimized. The EC2 Fleet launches 20 Spot Instances of the same instance type all in a single AZ from the Spot Instance pool with optimal capacity using the launch specifications.

```json
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true
    },
    "LaunchTemplateConfig": [
        {
            "LaunchTemplateName": "ec2-fleet-lt1",
            "Version": "$Latest"
        },
        "Overrides": [
            {
                "InstanceType": "c5.4xlarge",
                "SubnetId": "subnet-fae8c380"
            },
            {
                "InstanceType": "c5.4xlarge",
                "SubnetId": "subnet-e7188bab"
            },
            {
                "InstanceType": "c5.4xlarge",
                "SubnetId": "subnet-49e41922"
            }
        ]
    }
}
```
Example 6: Launch Spot Instances only if minimum target capacity can be launched

You can configure a fleet to launch instances only if the minimum target capacity can be launched by setting the Spot options MinTargetCapacity to the minimum target capacity you want to launch together.

The 12 launch template overrides have different instance types and subnets (each in a separate Availability Zone) but the same weighted capacity. The total target capacity and the minimum target capacity are both set to 20 instances, the default purchasing option is spot, the Spot allocation strategy is capacity-optimized. The EC2 Fleet launches 20 Spot Instances from the Spot capacity pool with optimal capacity using the launch template overrides, only if it can launch all 20 instances at the same time.
"AllocationStrategy": "capacity-optimized",
"MinTargetCapacity": 20
},
"LaunchTemplateConfigs": [
  {
    "LaunchTemplateSpecification": {
      "LaunchTemplateName": "ec2-fleet-lt1",
      "Version": "$Latest"
    },
    "Overrides": [
      {
        "InstanceType": "c5.4xlarge",
        "SubnetId": "subnet-fae8c380"
      },
      {
        "InstanceType": "c5.4xlarge",
        "SubnetId": "subnet-e7188bab"
      },
      {
        "InstanceType": "c5.4xlarge",
        "SubnetId": "subnet-49e41922"
      },
      {
        "InstanceType": "c5d.4xlarge",
        "SubnetId": "subnet-fae8c380"
      },
      {
        "InstanceType": "c5d.4xlarge",
        "SubnetId": "subnet-e7188bab"
      },
      {
        "InstanceType": "c5d.4xlarge",
        "SubnetId": "subnet-49e41922"
      },
      {
        "InstanceType": "m5.4xlarge",
        "SubnetId": "subnet-fae8c380"
      },
      {
        "InstanceType": "m5.4xlarge",
        "SubnetId": "subnet-e7188bab"
      },
      {
        "InstanceType": "m5d.4xlarge",
        "SubnetId": "subnet-49e41922"
      },
      {
        "InstanceType": "m5d.4xlarge",
        "SubnetId": "subnet-fae8c380"
      },
      {
        "InstanceType": "m5d.4xlarge",
        "SubnetId": "subnet-e7188bab"
      },
      {
        "InstanceType": "m5d.4xlarge",
        "SubnetId": "subnet-49e41922"
      }
    ]
  },
  "TargetCapacitySpecification": {
    "TotalTargetCapacity": 20,
    "DefaultTargetCapacityType": "spot"
  },
  "Type": "instant"}
Example 7: Launch Spot Instances only if minimum target capacity can be launched of same Instance Type in a single Availability Zone

You can configure a fleet to launch instances only if the minimum target capacity can be launched with a single instance type in a single Availability Zone by setting the Spot options MinTargetCapacity to the minimum target capacity you want to launch together along with SingleInstanceType and SingleAvailabilityZone options.

The 12 launch specifications which override the launch template, have different instance types and subnets (each in a separate Availability Zone) but the same weighted capacity. The total target capacity and the minimum target capacity are both set to 20 instances, the default purchasing option is spot, the Spot allocation strategy is capacity-optimized, the SingleInstanceType is true and SingleAvailabilityZone is true. The EC2 Fleet launches 20 Spot Instances of the same Instance type all in a single AZ from the Spot capacity pool with optimal capacity using the launch specifications, only if it can launch all 20 instances at the same time.

```json
{
  "SpotOptions": {
    "AllocationStrategy": "capacity-optimized",
    "SingleInstanceType": true,
    "SingleAvailabilityZone": true,
    "MinTargetCapacity": 20
  },
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateName": "ec2-fleet-lt1",
        "Version": "$Latest"
      },
      "Overrides": [
        {
          "InstanceType": "c5.4xlarge",
          "SubnetId": "subnet-fae8c380"
        },
        {
          "InstanceType": "c5.4xlarge",
          "SubnetId": "subnet-e7188bab"
        },
        {
          "InstanceType": "c5.4xlarge",
          "SubnetId": "subnet-49e41922"
        },
        {
          "InstanceType": "c5d.4xlarge",
          "SubnetId": "subnet-fae8c380"
        },
        {
          "InstanceType": "c5d.4xlarge",
          "SubnetId": "subnet-e7188bab"
        },
        {
          "InstanceType": "c5d.4xlarge",
          "SubnetId": "subnet-49e41922"
        },
        {
          "InstanceType": "m5.4xlarge",
          "SubnetId": "subnet-fae8c380"
        },
        {
          "InstanceType": "m5.4xlarge",
          "SubnetId": "subnet-e7188bab"
        },
        {
          "InstanceType": "m5.4xlarge",
          "SubnetId": "subnet-49e41922"
        },
        {
          "InstanceType": "m5.4xlarge",
          "SubnetId": "subnet-fae8c380"
        },
        {
          "InstanceType": "m5.4xlarge",
          "SubnetId": "subnet-e7188bab"
        }
    ]
  }
}
```
Example 8: Launch instances with multiple Launch Templates

You can configure a fleet to launch instances with different launch specifications for different instance
types or a group of instance types, by specifying multiple launch templates. In this example we want
to have different EBS volume sizes for different instance types and we have that configured in the launch
templates ec2-fleet-lt-4xl, ec2-fleet-lt-9xl and ec2-fleet-lt-18xl.

In this example, we are using 3 different launch templates for the 3 instance types based on their size.
The launch specification overrides on all the launch templates use instance weights based on the vCPUs
on the instance type. The total target capacity is 144 units, the default purchasing option is spot, and
the Spot allocation strategy is capacity-optimized. The EC2 Fleet can either launch 9 c5n.4xlarge (144
divided by 16) using the launch template ec2-fleet-lt-4xl or 4 c5n.9xlarge (144 divided by 36) using the
launch template ec2-fleet-lt-9xl, or 2 c5n.18xlarge (144 divided by 72) using the launch template ec2-
fleet-lt-18xl, or a mix of the instance types with weights adding up to the desired capacity based on the
capacity-optimized allocation strategy.

```json
{
  "SpotOptions": {
    "AllocationStrategy": "capacity-optimized"
  },
  "LaunchTemplateConfig": [
    {
      "LaunchTemplateName": "ec2-fleet-lt-18xl",
      "Version": "$Latest"
    },
    {
      "InstanceType": "c5n.18xlarge",
      "SubnetId": "subnet-fae8c380",
      "WeightedCapacity": 72
    },
    {
      "InstanceType": "c5n.18xlarge",
      "SubnetId": "subnet-e7188bab",
      "WeightedCapacity": 72
    }
  ]
}
```

677
Amazon Elastic Compute Cloud
User Guide for Windows Instances
EC2 Fleet request types

```json
{
  "LaunchTemplateSpecifications": [
    {
      "LaunchTemplateName": "ec2-fleet-la-18x",
      "Version": "$Latest"
    },
    {
      "LaunchTemplateName": "ec2-fleet-la-9x",
      "Version": "$Latest"
    },
    {
      "LaunchTemplateName": "ec2-fleet-la-4x",
      "Version": "$Latest"
    }
  ],
  "TargetCapacitySpecification": {
    "TotalTargetCapacity": 144,
    "DefaultTargetCapacityType": "spot"
  },
  "Type": "instant"
}
```
Example 9: Launch Spot Instance with a base of On-Demand Instances

The following example specifies the total target capacity of 20 instances for the fleet, and a target capacity of 5 On-Demand Instances. The default purchasing option is spot. The fleet launches 5 On-Demand Instance as specified, but needs to launch 15 more instances to fulfill the total target capacity. The purchasing option for the difference is calculated as \( \text{TotalTargetCapacity} - \text{OnDemandTargetCapacity} = \text{DefaultTargetCapacityType} \), which results in the fleet launching 15 Spot Instances form one of the 12 Spot capacity pools based on the capacity-optimized allocation strategy.

```json
{
  "SpotOptions": {
    "AllocationStrategy": "capacity-optimized"
  },
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateName": "ec2-fleet-lt1",
      "Version": "$Latest"
    },
    "LaunchTemplateConfig": [
      {
        "InstanceType": "c5.large",
        "SubnetId": "subnet-fae8c380"
      },
      {
        "InstanceType": "c5.large",
        "SubnetId": "subnet-e7188bab"
      },
      {
        "InstanceType": "c5.large",
        "SubnetId": "subnet-49e41922"
      },
      {
        "InstanceType": "c5d.large",
        "SubnetId": "subnet-fae8c380"
      },
      {
        "InstanceType": "c5d.large",
        "SubnetId": "subnet-e7188bab"
      },
      {
        "InstanceType": "c5d.large",
        "SubnetId": "subnet-49e41922"
      },
      {
        "InstanceType": "m5.large",
        "SubnetId": "subnet-fae8c380"
      },
      {
        "InstanceType": "m5.large",
        "SubnetId": "subnet-e7188bab"
      },
      {
        "InstanceType": "m5d.large",
        "SubnetId": "subnet-fae8c380"
      },
      {
        "InstanceType": "m5d.large",
        "SubnetId": "subnet-e7188bab"
      }
    ]
  }
}``
Example 10: Launch Spot Instances using capacity-optimized allocation strategy with a base of On-Demand Instances using Capacity Reservations and the prioritized allocation strategy

You can configure a fleet to use On-Demand Capacity Reservations first when launching a base of On-Demand Instances with the default target capacity type as spot by setting the usage strategy for Capacity Reservations to use-capacity-reservations-first. And if multiple instance pools have unused Capacity Reservations, the chosen On-Demand allocation strategy is applied. In this example, the On-Demand allocation strategy is prioritized.

In this example, there are 6 available unused Capacity Reservations. This is less than the fleet's target On-Demand capacity of 10 On-Demand Instances.

The account has the following 6 unused Capacity Reservations in 2 pools. The number of Capacity Reservations in each pool is indicated by AvailableInstanceCount.

{  
  "CapacityReservationId": "cr-111",  
  "InstanceType": "m5.large",  
  "InstancePlatform": "Linux/UNIX",  
  "AvailabilityZone": "us-east-1a",  
  "AvailableInstanceCount": 3,  
  "InstanceMatchCriteria": "open",  
  "State": "active"  
}

{  
  "CapacityReservationId": "cr-222",  
  "InstanceType": "c5.large",  
  "InstancePlatform": "Linux/UNIX",  
  "AvailabilityZone": "us-east-1a",  
  "AvailableInstanceCount": 3,  
  "InstanceMatchCriteria": "open",  
  "State": "active"  
}

The following fleet configuration shows only the pertinent configurations for this example. The On-Demand allocation strategy is prioritized, and the usage strategy for Capacity Reservations is use-capacity-reservations-first. The Spot allocation strategy is capacity-optimized. The total target capacity is 20, the On-Demand target capacity is 10, and the default target capacity type is spot.

{  
  "SpotOptions": {  
    "AllocationStrategy": "capacity-optimized"  
  },  
  "OnDemandOptions": {  
    "CapacityReservationOptions": {  
    }  
  }  
}
"UsageStrategy": "use-capacity-reservations-first"
},
"AllocationStrategy": "prioritized"
},
"LaunchTemplateConfig": [
{
"LaunchTemplateSpecification": {
"LaunchTemplateName": "ec2-fleet-lt1",
"Version": "$Latest"
},
"Overrides": [
{
"InstanceType": "c5.large",
"SubnetId": "subnet-fae8c380",
"Priority": 1.0
},
{
"InstanceType": "c5.large",
"SubnetId": "subnet-e7188bab",
"Priority": 2.0
},
{
"InstanceType": "c5.large",
"SubnetId": "subnet-49e41922",
"Priority": 3.0
},
{
"InstanceType": "c5d.large",
"SubnetId": "subnet-fae8c380",
"Priority": 4.0
},
{
"InstanceType": "c5d.large",
"SubnetId": "subnet-e7188bab",
"Priority": 5.0
},
{
"InstanceType": "c5d.large",
"SubnetId": "subnet-49e41922",
"Priority": 6.0
},
{
"InstanceType": "m5.large",
"SubnetId": "subnet-fae8c380",
"Priority": 7.0
},
{
"InstanceType": "m5.large",
"SubnetId": "subnet-e7188bab",
"Priority": 8.0
},
{
"InstanceType": "m5.large",
"SubnetId": "subnet-49e41922",
"Priority": 9.0
},
{
"InstanceType": "m5d.large",
"SubnetId": "subnet-fae8c380",
"Priority": 10.0
},
{
"InstanceType": "m5d.large",
"SubnetId": "subnet-e7188bab",
"Priority": 11.0
},
After you create the instant fleet using the preceding configuration, the following 20 instances are launched to meet the target capacity:

- 7 c5.large On-Demand Instances in us-east-1a – c5.large in us-east-1a is prioritized first, and there are 3 available unused c5.large Capacity Reservations. The Capacity Reservations are used first to launch 3 On-Demand Instances plus 4 additional On-Demand Instances are launched according to the On-Demand allocation strategy, which is prioritized in this example.
- 3 m5.large On-Demand Instances in us-east-1a – m5.large in us-east-1a is prioritized second, and there are 3 available unused c5.large Capacity Reservations.
- 10 Spot Instances from one of the 12 Spot capacity pools that has the optimal capacity according to the capacity-optimized allocation strategy.

After the fleet is launched, you can run describe-capacity-reservations to see how many unused Capacity Reservations are remaining. In this example, you should see the following response, which shows that all of the c5.large and m5.large Capacity Reservations were used.

Example 11: Launch Spot Instances using capacity-optimized-prioritized allocation strategy

The following example specifies the parameters required in an EC2 Fleet of type instant: a launch template, target capacity, default purchasing option, and launch template overrides. The launch template is identified by its launch template name and version number. The 12 launch specifications which override the launch template have 4 different instance types with a priority assigned, and 3 different subnets, each in a separate Availability Zone. The target capacity for the fleet is 20 instances, and the default purchasing option is spot, which results in the fleet attempting to launch 20 Spot Instances from one of the 12 Spot capacity pools based on the capacity-optimized-prioritized allocation strategy, which implements priorities on a best-effort basis, but optimizes for capacity first.

```json
{
   "SpotOptions": {
      "AllocationStrategy": "capacity-optimized-prioritized"
   },
}
"LaunchTemplateConfigs": [
  {
    "LaunchTemplateSpecification": {
      "LaunchTemplateName": "ec2-fleet-lt1",
      "Version": "$Latest"
    },
    "Overrides": [
      {
        "InstanceType": "c5.large",
        "SubnetId": "subnet-fae8c380",
        "Priority": 1.0
      },
      {
        "InstanceType": "c5.large",
        "SubnetId": "subnet-e7188bab",
        "Priority": 1.0
      },
      {
        "InstanceType": "c5.large",
        "SubnetId": "subnet-49e41922",
        "Priority": 1.0
      },
      {
        "InstanceType": "c5d.large",
        "SubnetId": "subnet-fae8c380",
        "Priority": 2.0
      },
      {
        "InstanceType": "c5d.large",
        "SubnetId": "subnet-e7188bab",
        "Priority": 2.0
      },
      {
        "InstanceType": "c5d.large",
        "SubnetId": "subnet-49e41922",
        "Priority": 2.0
      },
      {
        "InstanceType": "m5.large",
        "SubnetId": "subnet-fae8c380",
        "Priority": 3.0
      },
      {
        "InstanceType": "m5.large",
        "SubnetId": "subnet-e7188bab",
        "Priority": 3.0
      },
      {
        "InstanceType": "m5.large",
        "SubnetId": "subnet-49e41922",
        "Priority": 3.0
      },
      {
        "InstanceType": "m5d.large",
        "SubnetId": "subnet-fae8c380",
        "Priority": 4.0
      },
      {
        "InstanceType": "m5d.large",
        "SubnetId": "subnet-e7188bab",
        "Priority": 4.0
      },
      {
        "InstanceType": "m5d.large",
        "SubnetId": "subnet-49e41922",
        "Priority": 4.0
      }
    ]
]
EC2 Fleet configuration strategies

An EC2 Fleet is a group of On-Demand Instances and Spot Instances.

The EC2 Fleet attempts to launch the number of instances that are required to meet the target capacity that you specify in the fleet request. The fleet can comprise only On-Demand Instances, only Spot Instances, or a combination of both On-Demand Instances and Spot Instances. The request for Spot Instances is fulfilled if there is available capacity and the maximum price per hour for your request exceeds the Spot price. The fleet also attempts to maintain its target capacity if your Spot Instances are interrupted.

You can also set a maximum amount per hour that you're willing to pay for your fleet, and EC2 Fleet launches instances until it reaches the maximum amount. When the maximum amount you're willing to pay is reached, the fleet stops launching instances even if it hasn't met the target capacity.

A Spot capacity pool is a set of unused EC2 instances with the same instance type and Availability Zone. 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 capacity pools that are used to fulfill the request, based on the launch specifications included in your request, and the configuration of the request. The Spot Instances come from the selected pools.

An EC2 Fleet enables you to provision large amounts of EC2 capacity that makes sense for your application based on number of cores or instances, or amount of memory. For example, you can specify an EC2 Fleet to launch a target capacity of 200 instances, of which 130 are On-Demand Instances and the rest are Spot Instances.

Use the appropriate configuration strategies to create an EC2 Fleet that meets your needs.

Contents
- Planning an EC2 Fleet (p. 684)
- Allocation strategies for Spot Instances (p. 685)
- Configure EC2 Fleet for On-Demand backup (p. 687)
- Capacity Rebalancing (p. 688)
- Maximum price overrides (p. 690)
- Control spending (p. 690)
- EC2 Fleet instance weighting (p. 690)

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. 666).
EC2 Fleet configuration strategies

- 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. 690).
- Determine what portion of the EC2 Fleet target capacity must be On-Demand capacity and Spot capacity. You can specify 0 for On-Demand capacity or Spot capacity, or both.
- Determine your price per unit, if you are using instance weighting. To calculate the price per unit, divide the price per instance hour by the number of units (or weight) that this instance represents. If you are not using instance weighting, the default price per unit is the price per instance hour.
- Determine the maximum amount per hour that you’re willing to pay for your fleet. For more information, see Control spending (p. 690).
- Review the possible options for your EC2 Fleet. For more information, see the EC2 Fleet JSON configuration file reference (p. 698). For EC2 Fleet configuration examples, see EC2 Fleet example configurations (p. 766).

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 capacity pools represented by its launch specifications. The following are the allocation strategies that you can specify in your fleet:

lowest-price

The Spot Instances come from the Spot capacity pool with the lowest price. This is the default strategy.

diversified

The Spot Instances are distributed across all Spot capacity pools.

capacity-optimized

The Spot Instances come from the Spot capacity pool with optimal capacity for the number of instances that are launching. You can optionally set a priority for each instance type in your fleet using capacity-optimized-prioritized. EC2 Fleet optimizes for capacity first, but honors instance type priorities on a best-effort basis.

With Spot Instances, pricing changes slowly over time based on long-term trends in supply and demand, but capacity fluctuates in real time. The capacity-optimized strategy automatically launches Spot Instances into the most available pools by looking at real-time capacity data and predicting which are the most available. This works well for workloads such as big data and analytics, image and media rendering, machine learning, and high performance computing that may have a higher cost of interruption associated with restarting work and checkpointing. By offering the possibility of fewer interruptions, the capacity-optimized strategy can lower the overall cost of your workload.

Alternatively, you can use the capacity-optimized-prioritized allocation strategy with a priority parameter to order instance types from highest to lowest priority. You can set the same priority for different instance types. EC2 Fleet will optimize for capacity first, but will honor instance type priorities on a best-effort basis (for example, if honoring the priorities will not significantly affect EC2 Fleet's ability to provision optimal capacity). This is a good option for workloads where the possibility of disruption must be minimized and the preference for certain instance types matters. Using priorities is supported only if your fleet uses a launch template. Note that when you set the priority for capacity-optimized-prioritized, the same priority is also applied to your On-Demand Instances if the On-Demand AllocationStrategy is set to prioritized.
InstancePoolsToUseCount

The Spot Instances are distributed across the number of Spot capacity pools that you specify. This parameter is valid only when used in combination with lowest-price.

Maintaining target capacity

After Spot Instances are terminated due to a change in the Spot price or available capacity of a Spot capacity pool, an EC2 Fleet of type maintain launches replacement Spot Instances. If the allocation strategy is lowest-price, the fleet launches replacement instances in the pool where the Spot price is currently the lowest. If the allocation strategy is lowest-price in combination with InstancePoolsToUseCount, the fleet selects the Spot capacity pools with the lowest price and launches Spot Instances across the number of Spot capacity pools that you specify. If the allocation strategy is capacity-optimized, the fleet launches replacement instances in the pool that has the most available Spot Instance capacity. If the allocation strategy is diversified, the fleet distributes the replacement Spot Instances across the remaining pools.

Choose the appropriate allocation strategy

You can optimize your fleet based on your use case.

If your fleet runs workloads that may have a higher cost of interruption associated with restarting work and checkpointing, then use the capacity-optimized strategy. This strategy offers the possibility of fewer interruptions, which can lower the overall cost of your workload. Use the capacity-optimized-prioritized strategy for workloads where the possibility of disruption must be minimized and the preference for certain instance types matters.

If your fleet is small or runs for a short time, the probability that your Spot Instances will be interrupted is low, even with all of the instances in a single Spot capacity pool. Therefore, the lowest-price strategy is likely to meet your needs while providing the lowest cost.

If your fleet is large or runs for a long time, you can improve the availability of your fleet by distributing the Spot Instances across multiple pools using the diversified strategy. For example, if your EC2 Fleet specifies 10 pools and a target capacity of 100 instances, the fleet launches 10 Spot Instances in each pool. If the Spot price for one pool exceeds your maximum price for this pool, only 10% of your fleet is affected. Using this strategy also makes your fleet less sensitive to increases in the Spot price in any one pool over time. With the diversified strategy, the EC2 Fleet does not launch Spot Instances into any pools with a Spot price that is equal to or higher than the On-Demand price.

To create a cheap and diversified fleet, use the lowest-price strategy in combination with InstancePoolsToUseCount. You can use a low or high number of Spot capacity pools across which to allocate your Spot Instances. For example, if you run batch processing, we recommend specifying a low number of Spot capacity 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 capacity pools (for example, InstancePoolsToUseCount=10) to minimize the impact if a Spot capacity pool becomes temporarily unavailable.

Configure EC2 Fleet for cost optimization

To optimize the costs for your use of Spot Instances, specify the lowest-price allocation strategy so that EC2 Fleet automatically deploys the least expensive combination of instance types and Availability Zones based on the current Spot price.

For On-Demand Instance target capacity, EC2 Fleet always selects the cheapest instance type based on the public On-Demand price, while continuing to follow the allocation strategy (either lowest-price, capacity-optimized, or diversified) for Spot Instances.
Configure EC2 Fleet for cost optimization and diversification

To create a fleet of Spot Instances that is both cheap and diversified, use the lowest-price allocation strategy in combination with InstancePoolsToUseCount. EC2 Fleet automatically deploys the least expensive combination of instance types and Availability Zones based on the current Spot price across the number of Spot capacity pools that you specify. This combination can be used to avoid the most expensive Spot Instances.

For example, if your target capacity is 10 Spot Instances, and you specify 2 Spot capacity pools (for InstancePoolsToUseCount), EC2 Fleet will draw on the two cheapest pools to fulfill your Spot capacity.

Note that EC2 Fleet attempts to draw Spot Instances from the number of pools that you specify on a best effort basis. If a pool runs out of Spot capacity before fulfilling your target capacity, EC2 Fleet will continue to fulfill your request by drawing from the next cheapest pool. To ensure that your target capacity is met, you might receive Spot Instances from more than the number of pools that you specified. Similarly, if most of the pools have no Spot capacity, you might receive your full target capacity from fewer than the number of pools that you specified.

Configure EC2 Fleet for capacity optimization

To launch Spot Instances into the most-available Spot capacity pools, use the capacity-optimized allocation strategy. For an example configuration, see Example 10: Launch Spot Instances in a capacity-optimized fleet (p. 779).

You can also express your pool priorities by using the capacity-optimized-prioritized allocation strategy and then setting the order of instance types to use from highest to lowest priority. Using priorities is supported only if your fleet uses a launch template. Note that when you set priorities for capacity-optimized-prioritized, the same priorities are also applied to your On-Demand Instances if the On-Demand AllocationStrategy is set to prioritized. For an example configuration, see Example 11: Launch Spot Instances in a capacity-optimized fleet with priorities (p. 779).

Configure EC2 Fleet for On-Demand backup

If you have urgent, unpredictable scaling needs, such as a news website that must scale during a major news event or game launch, we recommend that you specify alternative instance types for your On-Demand Instances, in the event that your preferred option does not have sufficient available capacity. For example, you might prefer c5.2xlarge On-Demand Instances, but if there is insufficient available capacity, you'd be willing to use some c4.2xlarge instances during peak load. In this case, EC2 Fleet attempts to fulfill all of your target capacity using c5.2xlarge instances, but if there is insufficient capacity, it automatically launches c4.2xlarge instances to fulfill the target capacity.

Prioritize instance types for On-Demand capacity

When EC2 Fleet attempts to fulfill your On-Demand capacity, it defaults to launching the lowest-priced instance type first. If AllocationStrategy is set to prioritized, EC2 Fleet uses priority to determine which instance type to use first in fulfilling On-Demand capacity. The priority is assigned to the launch template override, and the highest priority is launched first.

For example, you have configured three launch template overrides, each with a different instance type: c3.large, c4.large, and c5.large. The On-Demand price for c5.large is less than the price for c4.large. c3.large is the cheapest. If you do not use priority to determine the order, the fleet fulfills On-Demand capacity by starting with c3.large, and then c5.large. Because you often have unused Reserved Instances for c4.large, you can set the launch template override priority so that the order is c4.large, c3.large, and then c5.large.
Use Capacity Reservations for On-Demand Instances

You can configure a fleet to use On-Demand Capacity Reservations first when launching On-Demand Instances by setting the usage strategy for Capacity Reservations to `use-capacity-reservations-first`. You can use this setting in conjunction with the allocation strategy for On-Demand Instances (`lowest-price` or `prioritized`).

When unused Capacity Reservations are used to fulfil On-Demand capacity:

- The fleet uses unused Capacity Reservations to fulfill On-Demand capacity up to the target On-Demand capacity.
- If multiple instance pools have unused Capacity Reservations, the On-Demand allocation strategy (`lowest-price` or `prioritized`) is applied.
- If the number of unused Capacity Reservations is less than the On-Demand target capacity, the remaining On-Demand target capacity is launched according to the On-Demand allocation strategy (`lowest-price` or `prioritized`).

You can only use unused On-Demand Capacity Reservations for fleets of type `instant`.

For examples of how to configure a fleet to use Capacity Reservations to fulfil On-Demand capacity, see EC2 Fleet example configurations (p. 766). For more information, see On-Demand Capacity Reservations (p. 366) and the On-Demand Capacity Reservation FAQs.

Capacity Rebalancing

You can configure EC2 Fleet to launch a replacement Spot Instance when Amazon EC2 emits a rebalance recommendation to notify you that a Spot Instance is at an elevated risk of interruption. Capacity Rebalancing helps you maintain workload availability by proactively augmenting your fleet with a new Spot Instance before a running instance is interrupted by Amazon EC2. For more information, see EC2 instance rebalance recommendations (p. 314).

To configure EC2 Fleet to launch a replacement Spot Instance, use the `create-fleet` (AWS CLI) command and the relevant parameters in the `MaintenanceStrategies` structure. For more information, see the example launch configuration (p. 778).

Limitations

- Only available for fleets of type `maintain`.
- When the fleet is running, you can't modify the Capacity Rebalancing setting. To change the Capacity Rebalancing setting, you must delete the fleet and create a new fleet.

Considerations

If you configure an EC2 Fleet for Capacity Rebalancing, consider the following:

EC2 Fleet can launch new replacement Spot Instances until fulfilled capacity is double target capacity

When an EC2 Fleet is configured for Capacity Rebalancing, the fleet attempts to launch a new replacement Spot Instance for every Spot Instance that receives a rebalance recommendation. After a Spot Instance receives a rebalance recommendation, it is no longer counted as part of the fulfilled capacity, and EC2 Fleet does not automatically terminate the instance. This gives you the opportunity to perform rebalancing actions (p. 315) on the instance. Thereafter, you can terminate the instance, or you can leave it running.

If your fleet reaches double its target capacity, it stops launching new replacement instances even if the replacement instances themselves receive a rebalance recommendation.
For example, you create an EC2 Fleet with a target capacity of 100 Spot Instances. All the Spot Instances receive a rebalance recommendation, which causes EC2 Fleet to launch 100 replacement Spot Instances. This raises the number of fulfilled Spot Instances to 200, which is double the target capacity. Some of the replacement instances receive a rebalance recommendation, but no more replacement instances are launched because the fleet cannot exceed double its target capacity.

Note that you are charged for all of the instances while they are running.

We recommend that you manually terminate Spot Instances that receive a rebalance recommendation

If you configure your EC2 Fleet for Capacity Rebalancing, we recommend that you monitor the rebalance recommendation signal that is received by the Spot Instances in the fleet. By monitoring the signal, you can quickly perform rebalancing actions (p. 315) on the affected instances before Amazon EC2 interrupts them, and then you can manually terminate them. If you do not terminate the instances, you continue paying for them while they are running. EC2 Fleet does not automatically terminate the instances that receive a rebalance recommendation.

You can set up notifications using Amazon EventBridge or instance metadata. For more information, see Monitor rebalance recommendation signals (p. 315).

EC2 Fleet does not count instances that receive a rebalance recommendation when calculating fulfilled capacity during scale in or out

If your EC2 Fleet is configured for Capacity Rebalancing, and you change the target capacity to either scale in or scale out, the fleet does not count the instances that are marked for rebalance as part of the fulfilled capacity, as follows:

- **Scale in** – If you decrease your desired target capacity, the fleet terminates instances that are not marked for rebalance until the desired capacity is reached. The instances that are marked for rebalance are not counted towards the fulfilled capacity.

  For example, you create an EC2 Fleet with a target capacity of 100 Spot Instances. 10 instances receive a rebalance recommendation, so the fleet launches 10 new replacement instances, resulting in a fulfilled capacity of 110 instances. You then reduce the target capacity to 50 (scale in), but the fulfilled capacity is actually 60 instances because the 10 instances that are marked for rebalance are not terminated by the fleet. You need to manually terminate these instances, or you can leave them running.

- **Scale out** – If you increase your desired target capacity, the fleet launches new instances until the desired capacity is reached. The instances that are marked for rebalance are not counted towards the fulfilled capacity.

  For example, you create an EC2 Fleet with a target capacity of 100 Spot Instances. 10 instances receive a rebalance recommendation, so the fleet launches 10 new replacement instances, resulting in a fulfilled capacity of 110 instances. You then increase the target capacity to 200 (scale out), but the fulfilled capacity is actually 210 instances because the 10 instances that are marked for rebalance are not counted by the fleet as part of the target capacity. You need to manually terminate these instances, or you can leave them running.

Provide as many Spot capacity pools in the request as possible

Configure your EC2 Fleet to use multiple instance types and Availability Zones. This provides the flexibility to launch Spot Instances in various Spot capacity pools. For more information, see Be flexible about instance types and Availability Zones (p. 286).

Configure your EC2 Fleet to use the most optimal Spot capacity pools

Use the capacity-optimized allocation strategy to ensure that replacement Spot Instances are launched in the most optimal Spot capacity pools. For more information, see Use the capacity optimized allocation strategy (p. 286).
Maximum price overrides

Each EC2 Fleet can either include a global maximum price, or use the default (the On-Demand price). The fleet uses this as the default maximum price for each of its launch specifications.

You can optionally specify a maximum price in one or more launch specifications. This price is specific to the launch specification. If a launch specification includes a specific price, the EC2 Fleet uses this maximum price, overriding the global maximum price. Any other launch specifications that do not include a specific maximum price still use the global maximum price.

Control spending

EC2 Fleet stops launching instances when it has met one of the following parameters: the TotalTargetCapacity or the MaxTotalPrice (the maximum amount you're willing to pay). To control the amount you pay per hour for your fleet, you can specify the MaxTotalPrice. When the maximum total price is reached, EC2 Fleet stops launching instances even if it hasn't met the target capacity.

The following examples show two different scenarios. In the first, EC2 Fleet stops launching instances when it has met the target capacity. In the second, EC2 Fleet stops launching instances when it has reached the maximum amount you're willing to pay (MaxTotalPrice).

Example: Stop launching instances when target capacity is reached

Given a request for m4.large On-Demand Instances, where:

- On-Demand Price: $0.10 per hour
- OnDemandTargetCapacity: 10
- MaxTotalPrice: $1.50

EC2 Fleet launches 10 On-Demand Instances because the total of $1.00 (10 instances x $0.10) does not exceed the MaxTotalPrice of $1.50 for On-Demand Instances.

Example: Stop launching instances when maximum total price is reached

Given a request for m4.large On-Demand Instances, where:

- On-Demand Price: $0.10 per hour
- OnDemandTargetCapacity: 10
- MaxTotalPrice: $0.80

If EC2 Fleet launches the On-Demand target capacity (10 On-Demand Instances), the total cost per hour would be $1.00. This is more than the amount ($0.80) specified for MaxTotalPrice for On-Demand Instances. To prevent spending more than you're willing to pay, EC2 Fleet launches only 8 On-Demand Instances (below the On-Demand target capacity) because launching more would exceed the MaxTotalPrice for On-Demand Instances.

EC2 Fleet instance weighting

When you create an EC2 Fleet, you can define the capacity units that each instance type would contribute to your application's performance. You can then adjust your maximum price for each launch specification by using instance weighting.

By default, the price that you specify is per instance hour. When you use the instance weighting feature, the price that you specify is per unit hour. You can calculate your price per unit hour by dividing your price for an instance type by the number of units that it represents. EC2 Fleet calculates the number of instances to launch by dividing the target capacity by the instance weight. If the result isn't an integer,
the fleet rounds it up to the next integer, so that the size of your fleet is not below its target capacity. The fleet can select any pool that you specify in your launch specification, even if the capacity of the instances launched exceeds the requested target capacity.

The following table includes examples of calculations to determine the price per unit for an EC2 Fleet with a target capacity of 10.

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Instance weight</th>
<th>Target capacity</th>
<th>Number of instances launched</th>
<th>Price per instance hour</th>
<th>Price per unit hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>r3.xlarge</td>
<td>2</td>
<td>10</td>
<td>5</td>
<td>$0.05</td>
<td>$0.025</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(.05 divided by 2)</td>
<td></td>
</tr>
<tr>
<td>r3.8xlarge</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>$0.10</td>
<td>$0.0125</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(.10 divided by 8)</td>
<td></td>
</tr>
</tbody>
</table>

Use EC2 Fleet instance weighting as follows to provision the target capacity that you want in the pools with the lowest price per unit at the time of fulfillment:

1. Set the target capacity for your EC2 Fleet either in instances (the default) or in the units of your choice, such as virtual CPUs, memory, storage, or throughput.
2. Set the price per unit.
3. For each launch specification, specify the weight, which is the number of units that the instance type represents toward the target capacity.

Instance weighting example

Consider an EC2 Fleet request with the following configuration:

- A target capacity of 24
- A launch specification with an instance type r3.2xlarge and a weight of 6
- A launch specification with an instance type c3.xlarge and a weight of 5

The weights represent the number of units that instance type represents toward the target capacity. If the first launch specification provides the lowest price per unit (price for r3.2xlarge per instance hour divided by 6), the EC2 Fleet would launch four of these instances (24 divided by 6).

If the second launch specification provides the lowest price per unit (price for c3.xlarge per instance hour divided by 5), the EC2 Fleet would launch five of these instances (24 divided by 5, result rounded up).

Instance weighting and allocation strategy

Consider an EC2 Fleet request with the following configuration:

- A target capacity of 30 Spot Instances
- A launch specification with an instance type c3.2xlarge and a weight of 8
- A launch specification with an instance type m3.xlarge and a weight of 8
- A launch specification with an instance type r3.xlarge and a weight of 8
The EC2 Fleet would launch four instances (30 divided by 8, result rounded up). With the lowest-price strategy, all four instances come from the pool that provides the lowest price per unit. With the diversified strategy, the fleet launches one instance in each of the three pools, and the fourth instance in whichever of the three pools provides the lowest price per unit.

Work with EC2 Fleets

To start using 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 of type maintain or request remains active until it expires or you delete it. When you delete a fleet of type maintain or request, you can specify whether deletion terminates the instances in that fleet.

Contents

- EC2 Fleet request states (p. 692)
- EC2 Fleet prerequisites (p. 693)
- EC2 Fleet health checks (p. 696)
- Generate an EC2 Fleet JSON configuration file (p. 696)
- Create an EC2 Fleet (p. 701)
- Tag an EC2 Fleet (p. 703)
- Monitor your EC2 Fleet (p. 705)
- Modify an EC2 Fleet (p. 706)
- Delete an EC2 Fleet (p. 707)

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. The request 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 fleet 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 manually. The request remains in this state
until all instances are interrupted or terminated. Only an EC2 Fleet of type *maintain* or *request* can have running instances after the EC2 Fleet request is deleted. A deleted instant fleet with running instances is not supported. This state does not apply to instant fleets.

deprecated

The EC2 Fleet request is deleted and its instances are terminating. The request remains in this state until all instances are terminated.

deprecated

The EC2 Fleet is deleted and has no running instances. The request is deleted two days after its instances are terminated.

The following illustration represents the transitions between the EC2 Fleet request states. If you exceed your fleet limits, the request is deleted immediately.

**EC2 Fleet prerequisites**

To create an EC2 Fleet, the following prerequisites must be in place:

- Launch template (p. 693)
- Service-linked role for EC2 Fleet (p. 693)
- Grant access to customer managed keys for use with encrypted AMIs and EBS snapshots (p. 694)
- Permissions for EC2 Fleet IAM users (p. 695)

**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 Launch an instance from a launch template (p. 397).

**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:
Work with EC2 Fleets

• ec2:RunInstances – Launch instances.
• ec2:RequestSpotInstances – Request Spot Instances.
• ec2:TerminateInstances – Terminate instances.
• ec2:DescribeImages – Describe Amazon Machine Images (AMIs) for the Spot Instances.
• ec2:DescribeInstanceStatus – Describe the status of the Spot Instances.
• ec2:DescribeSubnets – Describe the subnets for Spot Instances.
• ec2:CreateTags – Add tags to the EC2 Fleet, instances, and volumes.

Ensure that this role exists before you use the AWS CLI or an API to create an EC2 Fleet.

Note
An instant EC2 Fleet does not require this role.

To create the AWSServiceRoleForEC2Fleet role for EC2 Fleet

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose Roles, and then choose Create role.
3. For Select type of trusted entity, choose AWS service.
4. For Choose the service that will use this role, choose EC2 - Fleet, and then choose Next: Permissions, Next: Tags, and Next: Review.
5. On the Review page, choose Create role.

If you no longer need to use EC2 Fleet, we recommend that you delete the AWSServiceRoleForEC2Fleet role. After this role is deleted from your account, you can create the role again if you create another fleet.

For more information, see Using service-linked roles in the IAM User Guide.

Grant access to customer managed keys for use with encrypted AMIs and EBS snapshots

If you specify an encrypted AMI (p. 127) or an encrypted Amazon EBS snapshot (p. 1327) in your EC2 Fleet and you use an AWS KMS key for encryption, you must grant the AWSServiceRoleForEC2Fleet role permission to use the customer managed key so that Amazon EC2 can launch instances on your behalf. To do this, you must add a grant to the customer managed key, as shown in the following procedure.

When providing permissions, grants are an alternative to key policies. For more information, see Using grants and Using key policies in AWS KMS in the AWS Key Management Service Developer Guide.

To grant the AWSServiceRoleForEC2Fleet role permissions to use the customer managed key

• Use the create-grant command to add a grant to the customer managed key and to specify the principal (the AWSServiceRoleForEC2Fleet service-linked role) that is given permission to perform the operations that the grant permits. The customer managed key is specified by the key-id parameter and the ARN of the customer managed key. The principal is specified by the grantee-principal parameter and the ARN of the AWSServiceRoleForEC2Fleet service-linked role.

```bash
aws kms create-grant 
--region us-east-1 
--key-id arn:aws:kms:us-east-1:44445556666:key/1234abcd-12ab-34cd-56ef-1234567890ab 
--grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2Fleet 
--operations "Decrypt" "Encrypt" "GenerateDataKey" "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom" "ReEncryptTo"
```

694
Permissions for EC2 Fleet 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.

```json
{
  "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.

EC2 Fleet determines the health status of an instance by using the status checks provided by Amazon EC2. An instance is determined as unhealthy when the status of either the instance status check or the system status check is impaired for three consecutive health status checks. For more information, see *Status checks for your instances (p. 809)*.

You can configure your fleet to replace unhealthy Spot Instances. After setting *ReplaceUnhealthyInstances* to true, a Spot Instance is replaced when it is reported as unhealthy. The fleet can go below its target capacity for up to a few minutes while an unhealthy Spot Instance is being replaced.

**Requirements**

- Health check replacement is supported only for EC2 Fleets that maintain a target capacity (fleets of type *maintain*), and not for fleets of type *request* or *instant*.
- Health check replacement is supported only for Spot Instances. This feature is not supported for On-Demand Instances.
- 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.

**To configure an EC2 Fleet to replace unhealthy Spot Instances**

1. Follow the steps for creating an EC2 Fleet. For more information, see *Create an EC2 Fleet (p. 701)*.
2. To configure the fleet to replace unhealthy Spot Instances, in the JSON file, for *ReplaceUnhealthyInstances*, enter true.

**Generate an EC2 Fleet JSON configuration file**

To create an EC2 Fleet, you need only specify the launch template, total target capacity, and whether the default purchasing option is On-Demand or Spot. If you do not specify a parameter, the fleet uses the default value. To view the full list of fleet configuration parameters, you can generate a JSON file as follows.

**To generate a JSON file with all possible EC2 Fleet parameters using the command line**

- Use the `create-fleet` (AWS CLI) command and the `--generate-cli-skeleton` parameter to generate an EC2 Fleet JSON file:

  ```shell
  aws ec2 create-fleet
  --generate-cli-skeleton
  ```

  The following EC2 Fleet parameters are available:

  ```json
  {
    "DryRun": true,
    "ClientToken": "",
  }
  ```
"SpotOptions": {
  "AllocationStrategy": "lowest-price",
  "InstanceInterruptionBehavior": "hibernate",
  "InstanacePoolsToUseCount": 0,
  "SingleInstanceType": true,
  "SingleAvailabilityZone": true,
  "MaxTotalPrice": 0,
  "MinTargetCapacity": 0
},
"OnDemandOptions": {
  "AllocationStrategy": "prioritized",
  "SingleInstanceType": true,
  "SingleAvailabilityZone": true,
  "MaxTotalPrice": 0,
  "MinTargetCapacity": 0
},
"ExcessCapacityTerminationPolicy": "termination",
"LaunchTemplateConfigs": [
  {
    "LaunchTemplateSpecification": {
      "LaunchTemplateId": "",
      "LaunchTemplateName": "",
      "Version": ""
    },
    "overrides": [
      {
        "InstanceType": "t2.micro",
        "MaxPrice": "",
        "SubnetId": "",
        "AvailabilityZone": "",
        "WeightedCapacity": null,
        "Priority": null,
        "Placement": {
          "AvailabilityZone": "",
          "Affinity": "",
          "GroupName": "",
          "PartitionNumber": 0,
          "HostId": "",
          "Tenancy": "dedicated",
          "SpreadDomain": ""
        }
      }
    ]
  }
],
"TargetCapacitySpecification": {
  "TotalTargetCapacity": 0,
  "OnDemandTargetCapacity": 0,
  "SpotTargetCapacity": 0,
  "DefaultCapacityType": "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 capacity pools specified by the EC2 Fleet. Valid values are lowest-price, diversified, capacity-optimized, capacity-optimized-prioritized. The default is lowest-price. Specify the allocation strategy that meets your needs. For more information, see Allocation strategies for Spot Instances (p. 685).

**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 capacity pools across which to allocate your target Spot capacity. Valid only when Spot AllocationStrategy is set to lowest-price. EC2 Fleet selects the cheapest Spot capacity pools and evenly allocates your target Spot capacity across the number of Spot capacity pools that you specify.

**SingleInstanceType**
Indicates that the fleet uses a single instance type to launch all Spot Instances in the fleet.

**SingleAvailabilityZone**
Indicates that the fleet launches all Spot Instances into a single Availability Zone.

**MaxTotalPrice**
The maximum amount per hour for Spot Instances that you’re willing to pay.

**MinTargetCapacity**
The minimum target capacity for Spot Instances in the fleet. If the minimum target capacity is not reached, the fleet launches no instances.

**AllocationStrategy (for OnDemandOptions)**
The order of the launch template overrides to use in fulfilling On-Demand capacity. If you specify lowest-price, EC2 Fleet uses price to determine the order, launching the lowest price first. If you specify prioritized, EC2 Fleet uses the priority that you assigned to each launch template override, launching the highest priority first. If you do not specify a value, EC2 Fleet defaults to lowest-price.

**SingleInstanceType**
Indicates that the fleet uses a single instance type to launch all On-Demand Instances in the fleet.

**SingleAvailabilityZone**
Indicates that the fleet launches all On-Demand Instances into a single Availability Zone.

**MaxTotalPrice**
The maximum amount per hour for On-Demand Instances that you’re willing to pay.
MinTargetCapacity

The minimum target capacity for On-Demand Instances in the fleet. If the minimum target capacity is not reached, the fleet launches no instances.

ExcessCapacityTerminationPolicy

(Optional) Indicates whether running instances should be terminated if the total target capacity of the EC2 Fleet is decreased below the current size of the EC2 Fleet. Valid values are no-termination and termination.

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 Launch an instance from a launch template (p. 397).

Version

The launch template version number, $Latest, or $Default. You must specify a value, otherwise the request fails. If the value is $Latest, Amazon EC2 uses the latest version of the launch template. If the value is $Default, Amazon EC2 uses the default version of the launch template. For more information, see Modify a launch template (manage launch template versions) (p. 405).

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. The highest priority is launched first.

If the On-Demand AllocationStrategy is set to prioritized, EC2 Fleet uses priority to determine which launch template override to use first in fulfilling On-Demand capacity.

If the Spot AllocationStrategy is set to capacity-optimized-prioritized, EC2 Fleet uses priority on a best-effort basis to determine which launch template override to use first in fulfilling Spot capacity, but optimizes for capacity first.

Valid values are whole numbers starting at 0. The lower the number, the higher the priority. If no number is set, the launch template override has the lowest priority. You can set the same priority for different launch template overrides.

TotalTargetCapacity
The number of instances to launch. You can choose instances or performance characteristics that are important to your application workload, such as vCPUs, memory, or storage. If the request type is maintain, you can specify a target capacity of 0 and add capacity later.

OnDemandTargetCapacity
(Optional) The number of On-Demand Instances to launch. This number must be less than the TotalTargetCapacity.

SpotTargetCapacity
(Optional) The number of Spot Instances to launch. This number must be less than the TotalTargetCapacity.

DefaultTargetCapacityType
If the value for TotalTargetCapacity is higher than the combined values for OnDemandTargetCapacity and SpotTargetCapacity, the difference is launched as the instance purchasing option specified here. Valid values are on-demand or spot.

TerminateInstancesWithExpiration
(Optional) By default, Amazon EC2 terminates your instances when the EC2 Fleet request expires. The default value is true. To keep them running after your request expires, do not enter a value for this parameter.

Type
(Optional) The type of request. Valid values are instant, request, and maintain. The default value is maintain.

- instant – The EC2 Fleet submits a synchronous one-time request for your desired capacity, and returns errors for any instances that could not be launched.
- request – The EC2 Fleet submits an asynchronous one-time request for your desired capacity, but does submit Spot requests in alternative Spot capacity pools if Spot capacity is unavailable, and does not maintain Spot capacity if Spot Instances are interrupted.
- maintain – The EC2 Fleet submits an asynchronous request for your desired capacity, and continues to maintain your desired Spot capacity by replenishing interrupted Spot Instances.

For more information, see EC2 Fleet request types (p. 666).

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. 399). For information about tagging after launch, see Tag your resources (p. 1451).

Create 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. 698).

EC2 Fleets can only be created using the AWS CLI.

To create an EC2 Fleet (AWS CLI)

- Use the create-fleet (AWS CLI) command to create an EC2 Fleet.

```bash
aws ec2 create-fleet \
    --cli-input-json file://file_name.json
```

For example configuration files, see EC2 Fleet example configurations (p. 766).

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"
                }
            }
        }
    ]
}
```
The following is example output for a fleet of type `instance` that launched part of the target capacity with errors for instances that were not launched.

```json
{
    "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",
            "InstanceIds": ["i-1234567890abcdef0", "i-9876543210abcdef9"],
            "InstanceType": "c5.large",
            "Platform": null
        },
        {
            "LaunchTemplateAndOverrides": {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
                    "Version": "1"
                },
                "Overrides": {
                    "InstanceType": "c4.xlarge",
                    "AvailabilityZone": "us-east-1a"
                }
            },
            "Lifecycle": "on-demand",
            "InstanceIds": ["i-5678901234abcdef0", "i-5432109876abcdef9"],
            "InstanceType": "c4.large",
            "Platform": null
        }
    ],
    "Instances": [
        {
            "LaunchTemplateAndOverrides": {
                "LaunchTemplateSpecification": {
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
                    "Version": "1"
                },
                "Overrides": {
                    "InstanceType": "c5.large",
                    "InstanceIds": ["i-1234567890abcdef0", "i-9876543210abcdef9"],
                    "InstanceType": "c5.large",
                    "Platform": null
                }
            },
            "Lifecycle": "on-demand",
            "ErrorCode": "InsufficientInstanceCapacity",
            "ErrorMessage": "",
            "InstanceType": "c4.large",
            "Platform": null
        }
    ]
}
```
The following is example output for a fleet of type instant that launched no instances.

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
  "Errors": [
    {
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c4.xlarge",
          "AvailabilityZone": "us-east-1a",
          "Lifecycle": "on-demand",
          "ErrorCode": "InsufficientCapacity",
          "ErrorMessage": "",
          "InstanceType": "c4.xlarge",
          "Platform": null
        },
        "Lifecycle": "on-demand",
        "ErrorCode": "InsufficientCapacity",
        "ErrorMessage": "",
        "InstanceType": "c4.xlarge",
        "Platform": null
      },
      "LaunchTemplateAndOverrides": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
          "Version": "1"
        },
        "Overrides": {
          "InstanceType": "c5.large",
          "AvailabilityZone": "us-east-1a",
          "Lifecycle": "on-demand",
          "ErrorCode": "InsufficientCapacity",
          "ErrorMessage": "",
          "InstanceType": "c5.large",
          "Platform": null
        },
        "Lifecycle": "on-demand",
        "ErrorCode": "InsufficientCapacity",
        "ErrorMessage": "",
        "InstanceType": "c5.large",
        "Platform": null
      },
      "Instances": []
    }
  }
```

**Tag an EC2 Fleet**

To help categorize and manage your EC2 Fleet requests, you can tag them with custom metadata. You can assign a tag to an EC2 Fleet request when you create it, or afterward.
When you tag a fleet request, the instances and volumes that are launched by the fleet are not automatically tagged. You need to explicitly tag the instances and volumes launched by the fleet. You can choose to assign tags to only the fleet request, or to only the instances launched by the fleet, or to only the volumes attached to the instances launched by the fleet, or to all three.

**Note**
For instant fleet types, you can tag volumes that are attached to On-Demand Instances and Spot Instances. For request or maintain fleet types, you can only tag volumes that are attached to On-Demand Instances.

For more information about how tags work, see Tag your Amazon EC2 resources (p. 1450).

**Prerequisite**
Grant the IAM user the permission to tag resources. For more information, see Example: Tag resources (p. 1097).

**To grant an IAM user the permission to tag resources**
Create a IAM policy that includes the following:

- The `ec2:CreateTags` action. This grants the IAM user permission to create tags.
- The `ec2:CreateFleet` action. This grants the IAM user permission to create an EC2 Fleet request.
- For Resource, we recommend that you specify "*". This allows users to tag all resource types.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "TagEC2FleetRequest",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags",
        "ec2:CreateFleet"
      ],
      "Resource": "*"
    }
  ]
}
```

**Important**
We currently do not support resource-level permissions for the `create-fleet` resource. If you specify `create-fleet` as a resource, you will get an unauthorized exception when you try to tag the fleet. The following example illustrates how not to set the policy.

```json
{
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags",
    "ec2:CreateFleet"
  ],
}
```

**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. 696) used to create the fleet. The value for `ResourceType` must be `fleet`. If you specify another value, the fleet request fails.

**To tag instances and volumes launched by an EC2 Fleet**
To tag instances and volumes when they are launched by the fleet, specify the tags in the launch template (p. 399) that is referenced in the EC2 Fleet request.

Note
You can't tag volumes attached to Spot Instances that are launched by a request or maintain fleet type.

To tag an existing EC2 Fleet request, instance, and volume (AWS CLI)

Use the create-tags command to tag existing resources.

```
aws ec2 create-tags --resources fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE i-1234567890abcdef0 vol-1234567890EXAMPLE --tags Key=Key1,Value=Value1
```

Monitor your EC2 Fleet

The EC2 Fleet launches On-Demand Instances when there is available capacity, and launches Spot Instances when your maximum price exceeds the Spot price and capacity is available. The On-Demand Instances run until you terminate them, and the Spot Instances run until they are interrupted or you terminate them.

The returned list of running instances is refreshed periodically and might be out of date.

To monitor your EC2 Fleet (AWS CLI)

Use the describe-fleets command to describe your EC2 Fleets.

```
aws ec2 describe-fleets
```

The following is example output.

```json
{
   "Fleets": [
      {
         "Type": "maintain",
         "FulfilledCapacity": 2.0,
         "LaunchTemplateConfigs": [
            {
               "LaunchTemplateSpecification": {
                  "Version": "2",
                  "LaunchTemplateId": "lt-07b3bc7625cda851"
               }
            }
         ],
         "TerminateInstancesWithExpiration": false,
         "TargetCapacitySpecification": {
            "OnDemandTargetCapacity": 0,
            "SpotTargetCapacity": 2,
            "TotalTargetCapacity": 2,
            "DefaultTargetCapacityType": "spot"
         },
         "FulfilledOnDemandCapacity": 0.0,
         "ActivityStatus": "fulfilled",
         "FleetId": "fleet-76e13e99-01ef-4bd6-ba9b-9208de883e7f",
         "ReplaceUnhealthyInstances": false,
         "SpotOptions": {
            "InstanceInterruptionBehavior": "terminate",
            "InstancePoolsToUseCount": 1,
            "AllocationStrategy": "lowest-price"
         }
      }
   ]
}
```
Work with EC2 Fleets

Use the describe-fleet-instances command to describe the instances for the specified EC2 Fleet.

```
aws ec2 describe-fleet-instances \
   --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

```
{
    "ActiveInstances": [
      {
        "InstanceId": "i-09cd59598cb3765e",
        "InstanceHealth": "healthy",
        "InstanceType": "m4.large",
        "SpotInstanceRequestId": "sir-86k84j6p"
      },
      {
        "InstanceId": "i-09cf95167ca219ff17",
        "InstanceHealth": "healthy",
        "InstanceType": "m4.large",
        "SpotInstanceRequestId": "sir-dvxi7fsm"
      }
    ],
    "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

Use the describe-fleet-history command to describe the history for the specified EC2 Fleet for the specified time.

```
aws ec2 describe-fleet-history --fleet-request-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --start-time 2018-04-10T00:00:00Z
```

```
{
    "HistoryRecords": [],
    "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
    "LastEvaluatedTime": "1970-01-01T00:00:00.000Z",
    "StartTime": "2018-04-09T23:53:20.000Z"
}
```

Modify an EC2 Fleet

You can modify an EC2 Fleet that is in the submitted or active state. When you modify a fleet, it enters the modifying state.

You can only modify an EC2 Fleet that is of type maintain. You cannot modify an EC2 Fleet of type request or instant.

You can modify the following parameters of an EC2 Fleet:

- `target-capacity-specification` – Increase or decrease the target capacity for TotalTargetCapacity, OnDemandTargetCapacity, and SpotTargetCapacity.
- `excess-capacity-termination-policy` – Whether running instances should be terminated if the total target capacity of the EC2 Fleet is decreased below the current size of the fleet. Valid values are no-termination and termination.
When you increase the target capacity, the EC2 Fleet launches the additional instances according to the instance purchasing option specified for DefaultTargetCapacityType, which are either On-Demand Instances or Spot Instances.

If the DefaultTargetCapacityType is spot, the EC2 Fleet launches the additional Spot Instances according to its allocation strategy. If the allocation strategy is lowest-price, the fleet launches the instances from the lowest-priced Spot capacity pool in the request. If the allocation strategy is diversified, the fleet distributes the instances across the pools in the request.

When you decrease the target capacity, the EC2 Fleet deletes any open requests that exceed the new target capacity. You can request that the fleet terminate instances until the size of the fleet reaches the new target capacity. If the allocation strategy is lowest-price, the fleet terminates the instances with the highest price per unit. If the allocation strategy is diversified, the fleet terminates instances across the pools. Alternatively, you can request that EC2 Fleet keep the fleet at its current size, but not replace any Spot Instances that are interrupted or any instances that you terminate manually.

When an EC2 Fleet terminates a Spot Instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

To modify an EC2 Fleet (AWS CLI)

Use the modify-fleet command to update the target capacity of the specified EC2 Fleet.

```bash
aws ec2 modify-fleet \
   --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \n   --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.

```bash
aws ec2 modify-fleet \
   --fleet-id fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \n   --target-capacity-specification TotalTargetCapacity=10 \n   --excess-capacity-termination-policy no-termination
```

Delete 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.

When you delete an EC2 Fleet, you must specify if you want to also 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.

Restrictions

- You can delete up to 25 instant fleets in a single request. If you exceed this number, no instant fleets are deleted and an error is returned. There is no restriction on the number of fleets of type maintain or request that can be deleted in a single request.
- Up to 1000 instances can be terminated in a single request to delete instant fleets.

To delete an EC2 Fleet and terminate its instances (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-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
   --terminate-instances

The following is example output.

```
{
   "UnsuccessfulFleetDeletions": [],
   "SuccessfulFleetDeletions": [
      {
         "CurrentFleetState": "deleted_terminating",
         "PreviousFleetState": "active",
         "FleetId": "fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
      }
   ]
}
```

To delete an EC2 Fleet without terminating the instances (AWS CLI)

You can modify the previous command using the `--no-terminate-instances` parameter to delete the specified EC2 Fleet without terminating the instances.

**Note**

`--no-terminate-instances` is not supported for instant fleets.

aws ec2 delete-fleets \
   --fleet-ids fleet-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
   --no-terminate-instances

The following is example output.

```
{
   "UnsuccessfulFleetDeletions": [],
   "SuccessfulFleetDeletions": [
      {
         "CurrentFleetState": "deleted_running",
         "PreviousFleetState": "active",
         "FleetId": "fleet-4b8aaae8-dfb5-436d-a4c6-3dafa4c6b7dcEXAMPLE"
      }
   ]
}
```

Troubleshoot when a fleet fails to delete

If an EC2 Fleet fails to delete, `UnsuccessfulFleetDeletions` in the output returns the ID of the EC2 Fleet, an error code, and an error message.

The error codes are:

- ExceededInstantFleetNumForDeletion
- fleetIdDoesNotExist
- fleetIdMalformed
- fleetNotInDeletableState
- NoTerminateInstancesNotSupported
- UnauthorizedOperation
- unexpectedError
Troubleshooting ExceededInstantFleetNumForDeletion

If you try to delete more than 25 instant fleets in a single request, the ExceededInstantFleetNumForDeletion error is returned. The following is example output for this error.

```
{
    "UnsuccessfulFleetDeletions": [
        {
            "FleetId": "fleet-5d130460-0c26-bfd9-2c32-0100a098f625",
            "Error": {
                "Message": "Can’t delete more than 25 instant fleets in a single request.",
                "Code": "ExceededInstantFleetNumForDeletion"
            }
        },
        {
            "FleetId": "fleet-9a941b23-0286-5bf4-2430-03a029a07e31",
            "Error": {
                "Message": "Can’t delete more than 25 instant fleets in a single request.",
                "Code": "ExceededInstantFleetNumForDeletion"
            }
        }
    ],
    "SuccessfulFleetDeletions": []
}
```

Troubleshoot NoTerminateInstancesNotSupported

If you specify that the instances in an instant fleet must not be terminated when you delete the fleet, the NoTerminateInstancesNotSupported error is returned. --no-terminate-instances is not supported for instant fleets. The following is example output for this error.

```
{
    "UnsuccessfulFleetDeletions": [
        {
            "FleetId": "fleet-5d130460-0c26-bfd9-2c32-0100a098f625",
            "Error": {
                "Message": "NoTerminateInstances option is not supported for instant fleet",
                "Code": "NoTerminateInstancesNotSupported"
            }
        }
    ],
    "SuccessfulFleetDeletions": []
}
```

Troubleshoot UnauthorizedOperation

If you do not have permission to terminate instances, you get the UnauthorizedOperation error when deleting a fleet that must terminate its instances. The following is the error response.

```
<Response><Errors><Error><Code>UnauthorizedOperation</Code><Message>You are not authorized to perform this operation. Encoded authorization failure message: VvuncIxj75_CPGNYXWqnuFV-YjByeA06G9752ntq-13-qnDLWsrJIFdKnSMMiqS66CqjjiPeDpsnG9z2zYHasFHoaRY3paDVrov0W25azn6KNkUQQLPwhJyujzt2dNCdduJfrqCFYAjlElIRkfdHt7N63SK1weBHTurzXRA560F2nD51MmAbly9UNtqaZJ9SNe5aNhRMQ2aqKtljRbk02Rzv5V2vn9VMk6fm2aMVHbY9JhLvGypLcMUjtj76H9ytg2zR
```
To resolve the error, you must add the `ec2:TerminateInstances` action to the IAM policy, as shown in the following example.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DeleteFleetsAndTerminateInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteFleets",
                "ec2:TerminateInstances"
            ],
            "Resource": "*"
        }
    ]
}
```
Spot Fleet

A Spot Fleet is set of Spot Instances and optionally On-Demand Instances that is launched based on criteria that you specify. The Spot Fleet selects the Spot capacity 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.

Topics
- How Spot Fleet works (p. 711)
- Spot Fleet requests (p. 718)
- CloudWatch metrics for Spot Fleet (p. 737)
- Automatic scaling for Spot Fleet (p. 740)

How Spot Fleet works

A Spot Fleet is a collection, or fleet, of Spot Instances, and optionally On-Demand Instances.

The Spot Fleet attempts to launch the number of Spot Instances and On-Demand Instances to meet the target capacity that you specified in the Spot Fleet request. The request for Spot Instances is fulfilled if there is available capacity and the maximum price you specified in the request exceeds the current Spot price. The Spot Fleet also attempts to maintain its target capacity fleet if your Spot Instances are interrupted.

You can also set a maximum amount per hour that you're willing to pay for your fleet, and Spot Fleet launches instances until it reaches the maximum amount. When the maximum amount you're willing to pay is reached, the fleet stops launching instances even if it hasn't met the target capacity.

A Spot capacity 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 capacity 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. 711)
- Allocation strategy for Spot Instances (p. 712)
- Capacity Rebalancing (p. 714)
- Spot price overrides (p. 716)
- Control spending (p. 716)
- Spot Fleet instance weighting (p. 717)

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.

**Prioritize 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 capacity pools represented by its launch specifications. The following are the allocation strategies that you can specify in your Spot Fleet request:

- **lowestPrice**
  
  The Spot Instances come from the pool with the lowest price. This is the default strategy.

- **diversified**
  
  The Spot Instances are distributed across all pools.

- **capacityOptimized**
  
  The Spot Instances come from the pools with optimal capacity for the number of instances that are launching. You can optionally set a priority for each instance type in your fleet using `capacityOptimizedPrioritized`. Spot Fleet optimizes for capacity first, but honors instance type priorities on a best-effort basis.

  With Spot Instances, pricing changes slowly over time based on long-term trends in supply and demand, but capacity fluctuates in real time. The `capacityOptimized` strategy automatically launches Spot Instances into the most available pools by looking at real-time capacity data and predicting which are the most available. This works well for workloads such as big data and analytics, image and media rendering, machine learning, and high performance computing that may have a higher cost of interruption associated with restarting work and checkpointing. By offering the possibility of fewer interruptions, the `capacityOptimized` strategy can lower the overall cost of your workload.

  Alternatively, you can use the `capacityOptimizedPrioritized` allocation strategy with a priority parameter to order instance types from highest to lowest priority. You can set the same priority for different instance types. Spot Fleet will optimize for capacity first, but will honor instance type priorities on a best-effort basis (for example, if honoring the priorities will not significantly affect Spot Fleet's ability to provision optimal capacity). This is a good option for workloads where the possibility of disruption must be minimized and the preference for certain instance types matters. Using priorities is supported only if your fleet uses a launch template. Note that when you set the priority for `capacityOptimizedPrioritized`, the same priority is also applied to your On-Demand Instances if the `On-Demand AllocationStrategy` is set to `prioritized`.

- **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`. 

---

712
Maintain target capacity

After Spot Instances are terminated due to a change in the Spot price or available capacity of a Spot capacity 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.

Choose an appropriate allocation strategy

You can optimize your Spot Fleets based on your use case.

If your fleet runs workloads that may have a higher cost of interruption associated with restarting work and checkpointing, then use the capacityOptimized strategy. This strategy offers the possibility of fewer interruptions, which can lower the overall cost of your workload. This is the recommended strategy. Use the capacityOptimizedPrioritized strategy for workloads where the possibility of disruption must be minimized and the preference for certain instance types matters.

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 capacity 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 capacity pool becomes temporarily unavailable.

Configure Spot Fleet for cost optimization

To optimize the costs for your use of Spot Instances, specify the lowestPrice allocation strategy so that Spot Fleet automatically deploys the least expensive combination of instance types and Availability Zones based on the current Spot price.

For On-Demand Instance target capacity, Spot Fleet always selects the least expensive instance type based on the public On-Demand price, while continuing to follow the allocation strategy (either lowestPrice, capacityOptimized, or diversified) for Spot Instances.

Configure 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.
For example, if your target capacity is 10 Spot Instances, and you specify 2 Spot capacity pools (for 
InstancePoolsToUseCount), Spot Fleet will draw on the two cheapest pools to fulfill your Spot 
capacity.

Note that Spot Fleet attempts to draw Spot Instances from the number of pools that you specify 
on a best effort basis. If a pool runs out of Spot capacity before fulfilling your target capacity, Spot 
Fleet will continue to fulfill your request by drawing from the next cheapest pool. To ensure that your 
target capacity is met, you might receive Spot Instances from more than the number of pools that you 
specified. Similarly, if most of the pools have no Spot capacity, you might receive your full target capacity 
from fewer than the number of pools that you specified.

**Configure Spot Fleet for capacity optimization**

To launch Spot Instances into the most-available Spot capacity pools, use the capacityOptimized 
allocation strategy. For an example configuration, see Example 9: Launch Spot Instances in a capacity-
optimized fleet (p. 790).

You can also express your pool priorities by using the capacityOptimizedPrioritized allocation 
strategy and then setting the order of instance types to use from highest to lowest priority. Using 
priorities is supported only if your fleet uses a launch template. Note that when you set priorities for 
capacityOptimizedPrioritized, the same priorities are also applied to your On-Demand Instances 
if the OnDemandAllocationStrategy is set to prioritized. For an example configuration, see 
Example 10: Launch Spot Instances in a capacity-optimized fleet with priorities (p. 790).

**Capacity Rebalancing**

You can configure Spot Fleet to launch a replacement Spot Instance when Amazon EC2 emits a rebalance 
recommendation to notify you that a Spot Instance is at an elevated risk of interruption. Capacity 
Rebalancing helps you maintain workload availability by proactively augmenting your fleet with a new 
Spot Instance before a running instance is interrupted by Amazon EC2. For more information, see EC2 
instance rebalance recommendations (p. 314).

To configure Spot Fleet to launch a replacement Spot Instance, you can use the Amazon EC2 console or 
the AWS CLI.

- Amazon EC2 console: You must select the **Capacity rebalance** check box when you create the Spot 
  Fleet. For more information, see step 6.d. in Create a Spot Fleet request using defined parameters 
  (console) (p. 726).

- AWS CLI: Use the request-spot-fleet command and the relevant parameters in the 
  SpotMaintenanceStrategies structure. For more information, see the example launch 
  configuration (p. 789).

**Limitations**

- Only available for fleets of type **maintain**.

- When the fleet is running, you can't modify the Capacity Rebalancing setting. To change the Capacity 
  Rebalancing setting, you must delete the fleet and create a new fleet.

**Considerations**

If you configure a Spot Fleet for Capacity Rebalancing, consider the following:

**Spot Fleet can launch new replacement Spot Instances until fulfilled capacity is double target 
capacity**

When a Spot Fleet is configured for Capacity Rebalancing, the fleet attempts to launch a new 
replacement Spot Instance for every Spot Instance that receives a rebalance recommendation.
After a Spot Instance receives a rebalance recommendation, it is no longer counted as part of the fulfilled capacity, and Spot Fleet does not automatically terminate the instance. This gives you the opportunity to perform rebalancing actions (p. 315) on the instance. Thereafter, you can terminate the instance, or you can leave it running.

If your fleet reaches double its target capacity, it stops launching new replacement instances even if the replacement instances themselves receive a rebalance recommendation.

For example, you create a Spot Fleet with a target capacity of 100 Spot Instances. All the Spot Instances receive a rebalance recommendation, which causes Spot Fleet to launch 100 replacement Spot Instances. This raises the number of fulfilled Spot Instances to 200, which is double the target capacity. Some of the replacement instances receive a rebalance recommendation, but no more replacement instances are launched because the fleet cannot exceed double its target capacity.

Note that you are charged for all of the instances while they are running.

**We recommend that you manually terminate Spot Instances that receive a rebalance recommendation**

If you configure your Spot Fleet for Capacity Rebalancing, we recommend that you monitor the rebalance recommendation signal that is received by the Spot Instances in the fleet. By monitoring the signal, you can quickly perform rebalancing actions (p. 315) on the affected instances before Amazon EC2 interrupts them, and then you can manually terminate them. If you do not terminate the instances, you continue paying for them while they are running. Spot Fleet does not automatically terminate the instances that receive a rebalance recommendation.

You can set up notifications using Amazon EventBridge or instance metadata. For more information, see Monitor rebalance recommendation signals (p. 315).

**Spot Fleet does not count instances that receive a rebalance recommendation when calculating fulfilled capacity during scale in or out**

If your Spot Fleet is configured for Capacity Rebalancing, and you change the target capacity to either scale in or scale out, the fleet does not count the instances that are marked for rebalance as part of the fulfilled capacity, as follows:

- **Scale in** – If you decrease your desired target capacity, the fleet terminates instances that are not marked for rebalance until the desired capacity is reached. The instances that are marked for rebalance are not counted towards the fulfilled capacity.

  For example, you create a Spot Fleet with a target capacity of 100 Spot Instances. 10 instances receive a rebalance recommendation, so the fleet launches 10 new replacement instances, resulting in a fulfilled capacity of 110 instances. You then reduce the target capacity to 50 (scale in), but the fulfilled capacity is actually 60 instances because the 10 instances that are marked for rebalance are not terminated by the fleet. You need to manually terminate these instances, or you can leave them running.

- **Scale out** – If you increase your desired target capacity, the fleet launches new instances until the desired capacity is reached. The instances that are marked for rebalance are not counted towards the fulfilled capacity.

  For example, you create a Spot Fleet with a target capacity of 100 Spot Instances. 10 instances receive a rebalance recommendation, so the fleet launches 10 new replacement instances, resulting in a fulfilled capacity of 110 instances. You then increase the target capacity to 200 (scale out), but the fulfilled capacity is actually 210 instances because the 10 instances that are marked for rebalance are not counted by the fleet as part of the target capacity. You need to manually terminate these instances, or you can leave them running.

**Provide as many Spot capacity pools in the request as possible**

Configure your Spot Fleet to use multiple instance types and Availability Zones. This provides the flexibility to launch Spot Instances in various Spot capacity pools. For more information, see Be flexible about instance types and Availability Zones (p. 286).
Configure your Spot Fleet to use the most optimal Spot capacity pools

Use the capacity-optimized allocation strategy to ensure that replacement Spot Instances are launched in the most optimal Spot capacity pools. For more information, see Use the capacity optimized allocation strategy (p. 286).

Spot price overrides

Each Spot Fleet request can include a global maximum price, or use the default (the On-Demand price). Spot Fleet uses this as the default maximum price for each of its launch specifications.

You can optionally specify a maximum price in one or more launch specifications. This price is specific to the launch specification. If a launch specification includes a specific price, the Spot Fleet uses this maximum price, overriding the global maximum price. Any other launch specifications that do not include a specific maximum price still use the global maximum price.

Control spending

Spot Fleet stops launching instances when it has either reached the target capacity or the maximum amount you're willing to pay. To control the amount you pay per hour for your fleet, you can specify the SpotMaxTotalPrice for Spot Instances and the OnDemandMaxTotalPrice for On-Demand Instances. When the maximum total price is reached, Spot Fleet stops launching instances even if it hasn't met the target capacity.

The following examples show two different scenarios. In the first, Spot Fleet stops launching instances when it has met the target capacity. In the second, Spot Fleet stops launching instances when it has reached the maximum amount you're willing to pay.

Example: Stop launching instances when target capacity is reached

Given a request for m4.large On-Demand Instances, where:

- On-Demand Price: $0.10 per hour
- OnDemandTargetCapacity: 10
- OnDemandMaxTotalPrice: $1.50

Spot Fleet launches 10 On-Demand Instances because the total of $1.00 (10 instances x $0.10) does not exceed the OnDemandMaxTotalPrice of $1.50.

Example: Stop launching instances when maximum total price is reached

Given a request for m4.large On-Demand Instances, where:

- On-Demand Price: $0.10 per hour
- OnDemandTargetCapacity: 10
- OnDemandMaxTotalPrice: $0.80

If Spot Fleet launches the On-Demand target capacity (10 On-Demand Instances), the total cost per hour would be $1.00. This is more than the amount ($0.80) specified for OnDemandMaxTotalPrice. To prevent spending more than you're willing to pay, Spot Fleet launches only 8 On-Demand Instances (below the On-Demand target capacity) because launching more would exceed the OnDemandMaxTotalPrice.
Spot Fleet instance weighting

When you request a fleet of Spot Instances, you can define the capacity units that each instance type would contribute to your application's performance, and adjust your maximum price for each Spot capacity 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.

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Instance weight</th>
<th>Price per instance hour</th>
<th>Price per unit hour</th>
<th>Number of instances launched</th>
</tr>
</thead>
<tbody>
<tr>
<td>r3.xlarge</td>
<td>2</td>
<td>$0.05</td>
<td>.025</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.05 divided by 2)</td>
<td></td>
<td>(10 divided by 2)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Instance weight</th>
<th>Price per instance hour</th>
<th>Price per unit hour</th>
<th>Number of instances launched</th>
</tr>
</thead>
<tbody>
<tr>
<td>r3.8xlarge</td>
<td>8</td>
<td>$0.10</td>
<td>.0125</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(.10 divided by 8)</td>
<td></td>
<td>(10 divided by 8, result rounded up)</td>
</tr>
</tbody>
</table>

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.

**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 Spot Fleet (p. 711).

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. 719)
- Spot Fleet health checks (p. 719)
- Plan a Spot Fleet request (p. 720)
- Spot Fleet permissions (p. 721)
- Create a Spot Fleet request (p. 725)
- Tag a Spot Fleet (p. 729)
- Monitor your Spot Fleet (p. 735)
Spot Fleet request states

A Spot Fleet request can be in one of the following states:

- **submitted** – The Spot Fleet request is being evaluated and Amazon EC2 is preparing to launch the target number of instances.
- **active** – The Spot Fleet has been validated and Amazon EC2 is attempting to maintain the target number of running Spot Instances. The request remains in this state until it is modified or canceled.
- **modifying** – The Spot Fleet request is being modified. The request remains in this state until the modification is fully processed or the Spot Fleet is canceled. A one-time request cannot be modified, and this state does not apply to such Spot requests.
- **cancelled_running** – The Spot Fleet is canceled and does not launch additional Spot Instances. Its existing Spot Instances continue to run until they are interrupted or terminated. The request remains in this state until all instances are interrupted or terminated.
- **cancelled_terminating** – The Spot Fleet is canceled and its Spot Instances are terminating. The request remains in this state until all instances are terminated.
- **cancelled** – The Spot Fleet is canceled and has no running Spot Instances. The Spot Fleet request is deleted two days after its instances were terminated.

The following illustration represents the transitions between the request states. If you exceed your Spot Fleet limits, the request is canceled immediately.

Spot Fleet health checks

Spot Fleet checks the health status of the Spot Instances in the fleet every two minutes. The health status of an instance is either **healthy** or **unhealthy**.

Spot Fleet determines the health status of an instance by using the status checks provided by Amazon EC2. An instance is determined as **unhealthy** when the status of either the instance status check
or the system status check is impaired for three consecutive health checks. For more information, see Status checks for your instances (p. 809).

You can configure your fleet to replace unhealthy Spot Instances. After enabling health check replacement, a Spot Instance is replaced when it is reported as unhealthy. The fleet could go below its target capacity for up to a few minutes while an unhealthy Spot Instance is being replaced.

**Requirements**

- Health check replacement is supported only for Spot Fleets that maintain a target capacity (fleets of type maintain), not for one-time Spot Fleets (fleets of type request).
- Health check replacement is supported only for Spot Instances. This feature is not supported for On-Demand Instances.
- 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.

**Console**

**To configure a Spot Fleet to replace unhealthy Spot Instances using the console**

1. Follow the steps for creating a Spot Fleet. For more information, see Create a Spot Fleet request using defined parameters (console) (p. 726).
2. To configure the fleet to replace unhealthy Spot Instances, for Health check, choose Replace unhealthy instances. To enable this option, you must first choose Maintain target capacity.

**AWS CLI**

**To configure a Spot Fleet to replace unhealthy Spot Instances using the AWS CLI**

1. Follow the steps for creating a Spot Fleet. For more information, see Create a Spot Fleet using the AWS CLI (p. 728).
2. To configure the fleet to replace unhealthy Spot Instances, for ReplaceUnhealthyInstances, enter true.

**Plan 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. 717).
- 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. 780).
Spot Fleet permissions

If your IAM users will create or manage a Spot Fleet, you need to grant them the required permissions.

If you use the Amazon EC2 console to create a Spot Fleet, it creates two service-linked roles named AWSServiceRoleForEC2SpotFleet and AWSServiceRoleForEC2Spot, and a role named aws-ec2-spot-fleet-tagging-role that grant the Spot Fleet the permissions to request, launch, terminate, and tag resources on your behalf. If you use the AWS CLI or an API, you must ensure that these roles exist.

Use the following instructions to grant the required permissions and create the roles.

Permissions and roles
- Grant permission to IAM users for Spot Fleet (p. 721)
- Service-linked role for Spot Fleet (p. 723)
- Service-linked role for Spot Instances (p. 724)
- IAM role for tagging a Spot Fleet (p. 725)

Grant permission to IAM users for Spot Fleet

If your IAM users will create or manage a Spot Fleet, be sure to grant them the required permissions as follows.

To grant an IAM user permissions for Spot Fleet

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose Policies, Create policy.
3. On the Create policy page, choose JSON, and replace the text with the following.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "ec2:RunInstances",
            "ec2:CreateTags",
            "ec2:RequestSpotFleet",
            "ec2:ModifySpotFleetRequest",
            "ec2:CancelSpotFleetRequests",
            "ec2:DescribeSpotFleetRequests",
            "ec2:DescribeSpotFleetInstances",
            "ec2:DescribeSpotFleetRequestHistory"
         ],
         "Resource": "*"
      },
      {
         "Effect": "Allow",
         "Action": "iam:PassRole",
         "Resource": "arn:aws:iam::*:role/aws-ec2-spot-fleet-tagging-role"
      },
      {
         "Effect": "Allow",
         "Action": [
            "iam:CreateServiceLinkedRole",
            "iam:ListRoles",
            "iam:ListInstanceProfiles"
         ],
      }
   ]
}
```

721
The preceding example policy grants an IAM user the permissions required for most Spot Fleet use cases. To limit the user to specific API actions, specify only those API actions instead.

**Required EC2 and IAM APIs**

The following APIs must be included in the policy:

- `ec2:RunInstances` – Required to launch instances in a Spot Fleet
- `ec2:CreateTags` – Required to tag the Spot Fleet request, instances, or volumes
- `iam:PassRole` – Required to specify the Spot Fleet role
- `iam:CreateServiceLinkedRole` – Required to create the service-linked role
- `iam:ListRoles` – Required to enumerate existing IAM roles
- `iam:ListInstanceProfiles` – Required to enumerate existing instance profiles

**Important**

If you specify a role for the IAM instance profile in the launch specification or launch template, you must grant the IAM user the permission to pass the role to the service. To do this, in the IAM policy include "arn:aws:iam::*:role/IamInstanceProfile-role" as a resource for the `iam:PassRole` action. For more information, see Granting a user permissions to pass a role to an AWS service in the IAM User Guide.

**Spot Fleet APIs**

Add the following Spot Fleet API actions to your policy, as needed:

- `ec2:RequestSpotFleet`
- `ec2:ModifySpotFleetRequest`
- `ec2:CancelSpotFleetRequests`
- `ec2:DescribeSpotFleetRequests`
- `ec2:DescribeSpotFleetInstances`
- `ec2:DescribeSpotFleetRequestHistory`

**Optional IAM APIs**

(Optional) To enable an IAM user to create roles or instance profiles using the IAM console, you must add the following actions to the policy:

- `iam:AddRoleToInstanceProfile`
- `iam:AttachRolePolicy`
- `iam:CreateInstanceProfile`
- `iam:CreateRole`
- `iam:GetRole`
- `iam:ListPolicies`

4. Choose **Review policy**.

5. On the **Review policy** page, enter a policy name and description, and choose **Create policy**.

6. In the navigation pane, choose **Users** and select the user.

7. Choose **Permissions, Add permissions**. 722
8. Choose **Attach existing policies directly**. Select the policy that you created earlier and choose **Next: Review**.

9. Choose **Add permissions**.

### Service-linked role for Spot Fleet

Amazon EC2 uses service-linked roles for the permissions that it requires to call other AWS services on your behalf. A service-linked role is a unique type of IAM role that is linked directly to an AWS service. Service-linked roles provide a secure way to delegate permissions to AWS services because only the linked service can assume a service-linked role. For more information, see Using Service-Linked Roles in the [IAM User Guide](https://docs.aws.amazon.com/IAM/latest/UserGuide/gs-service-linked-roles.html).

Amazon EC2 uses the service-linked role named `AWSServiceRoleForEC2SpotFleet` to launch and manage instances on your behalf.

**Important**

If you specify an encrypted AMI (p. 127) or an encrypted Amazon EBS snapshot (p. 1327) in your Spot Fleet, you must grant the `AWSServiceRoleForEC2SpotFleet` role permission to use the CMK so that Amazon EC2 can launch instances on your behalf. For more information, see Grant access to CMKs for use with encrypted AMIs and EBS snapshots (p. 724).

**Permissions granted by AWSServiceRoleForEC2SpotFleet**

Amazon EC2 uses `AWSServiceRoleForEC2SpotFleet` to complete the following actions:

- `ec2:RequestSpotInstances` - Request Spot Instances
- `ec2:RunInstances` - Launch instances
- `ec2:TerminateInstances` - Terminate instances
- `ec2:DescribeImages` - Describe Amazon Machine Images (AMIs) for the instances
- `ec2:DescribeInstanceStatus` - Describe the status of the instances
- `ec2:DescribeSubnets` - Describe the subnets for the instances
- `ec2:CreateTags` - Add tags to the Spot Fleet request, instances, and volumes
- `elasticloadbalancing:RegisterInstancesWithLoadBalancer` - Add the specified instances to the specified load balancer
- `elasticloadbalancing:RegisterTargets` - Register the specified targets with the specified target group

### Create the service-linked role

Under most circumstances, you don't need to manually create a service-linked role. Amazon EC2 creates the `AWSServiceRoleForEC2SpotFleet` service-linked role the first time you create a Spot Fleet using the console.

If you had an active Spot Fleet request before October 2017, when Amazon EC2 began supporting this service-linked role, Amazon EC2 created the `AWSServiceRoleForEC2SpotFleet` role in your AWS account. For more information, see **A new role appeared in my AWS account** in the [IAM User Guide](https://docs.aws.amazon.com/IAM/latest/UserGuide/gs-service-linked-roles.html).

If you use the AWS CLI or an API to create a Spot Fleet, you must first ensure that this role exists.

**To create AWSServiceRoleForEC2SpotFleet using the console**

2. In the navigation pane, choose **Roles**.
3. Choose Create role.
4. For Select type of trusted entity, choose AWS service.
5. Under Choose a use case, Or select a service to view its use cases, choose EC2.
6. Under Select your use case, choose EC2 - Spot Fleet.
7. Choose Next: Permissions.
8. On the next page, choose Next: Tags.

To create AWSServiceRoleForEC2SpotFleet using the AWS CLI

Use the create-service-linked-role command as follows.

```
aws iam create-service-linked-role --aws-service-name spotfleet.amazonaws.com
```

If you no longer need to use Spot Fleet, we recommend that you delete the AWSServiceRoleForEC2SpotFleet role. After this role is deleted from your account, Amazon EC2 will create the role again if you request a Spot Fleet using the console. For more information, see Deleting a Service-Linked Role in the IAM User Guide.

Grant access to CMKs for use with encrypted AMIs and EBS snapshots

If you specify an encrypted AMI (p. 127) or an encrypted Amazon EBS snapshot (p. 1327) in your Spot Fleet request and you use a customer managed customer master key (CMK) for encryption, you must grant the AWSServiceRoleForEC2SpotFleet role permission to use the CMK so that Amazon EC2 can launch instances on your behalf. To do this, you must add a grant to the CMK, as shown in the following procedure.

When providing permissions, grants are an alternative to key policies. For more information, see Using Grants and Using Key Policies in AWS KMS in the AWS Key Management Service Developer Guide.

To grant the AWSServiceRoleForEC2SpotFleet role permissions to use the CMK

- Use the create-grant command to add a grant to the CMK and to specify the principal (the AWSServiceRoleForEC2SpotFleet service-linked role) that is given permission to perform the operations that the grant permits. The CMK is specified by the key-id parameter and the ARN of the CMK. The principal is specified by the grantee-principal parameter and the ARN of the AWSServiceRoleForEC2SpotFleet service-linked role.

```
aws kms create-grant \
--region us-east-1 \
--key-id arn:aws:kms:us-east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
--grantee-principal arn:aws:iam::111122233333:role/AWSServiceRoleForEC2SpotFleet \
--operations "Decrypt" "Encrypt" "GenerateDataKey" "GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom" "ReEncryptTo"
```

Service-linked role for Spot Instances

Amazon EC2 uses the service-linked role named AWSServiceRoleForEC2Spot to launch and manage Spot Instances on your behalf. For more information, see Service-linked role for Spot Instance requests (p. 294).
IAM role for tagging a Spot Fleet

The `aws-ec2-spot-fleet-tagging-role` IAM role grants the Spot Fleet permission to tag the Spot Fleet request, instances, and volumes. For more information, see Tag a Spot Fleet (p. 729).

**Important**
If you choose to tag instances in the fleet and you choose to maintain target capacity (the Spot Fleet request is of type `maintain`), the differences in permissions of the IAM user and the `IamFleetRole` might lead to inconsistent tagging behavior of instances in the fleet. If the `IamFleetRole` does not include the `CreateTags` permission, some of the instances launched by the fleet might not be tagged. While we are working to fix this inconsistency, to ensure that all instances launched by the fleet are tagged, we recommend that you use the `aws-ec2-spot-fleet-tagging-role` role for the `IamFleetRole`. Alternatively, to use an existing role, attach the `AmazonEC2SpotFleetTaggingRole` AWS Managed Policy to the existing role. Otherwise, you need to manually add the `CreateTags` permission to your existing policy.

To create the IAM role for tagging a Spot Fleet

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**.
3. Choose **Create roles**.
4. On the **Select type of trusted entity** page, choose **AWS service**.
5. Under **Choose a use case**, **Or select a service to view its use cases**, choose **EC2**.
6. Under **Select your use case**, choose **EC2 - Spot Fleet Tagging**.
7. Choose **Next: Permissions**.
8. On the next page, choose **Next: Tags**.
9. On the next page, choose **Next: Review**.
10. On the **Review** page, enter a name for the role (for example, `aws-ec2-spot-fleet-tagging-role`) and choose **Create role**.

Create 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. 725). Otherwise, you can modify any of the default settings. For more information, see Create a Spot Fleet request using defined parameters (console) (p. 726) and Create a Spot Fleet using the AWS CLI (p. 728).

Options for creating a Spot Fleet
- Quickly create a Spot Fleet request (console) (p. 725)
- Create a Spot Fleet request using defined parameters (console) (p. 726)
- Create a Spot Fleet using the AWS CLI (p. 728)

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)

2. If you are new to Spot, you see a welcome page; choose **Get started**. Otherwise, choose **Request Spot Instances**.
3. For **Tell us your application or task need**, choose **Load balancing workloads**, **Flexible workloads**, or **Big data 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)**

2. If you are new to Spot, you see a welcome page; choose **Get started**. Otherwise, choose **Request Spot Instances**.
3. For **Tell us your application or task need**, choose **Load balancing workloads**, **Flexible workloads**, or **Big data workloads**.
4. For **Configure your instances**, do the following:
   a. (Optional) For **Launch template**, choose a launch template. The launch template must specify an Amazon Machine Image (AMI), as you cannot override the AMI using Spot Fleet if you specify a launch template.
      
      **Important**
      If you intend to specify **Optional On-Demand portion**, you must choose a launch template.

   b. For **AMI**, choose one of the basic AMIs provided by AWS, or choose **Search for AMI** to use an AMI from our user community, the AWS Marketplace, or one of your own.

c. For **Minimum compute unit**, choose the minimum hardware specifications (vCPUs, memory, and storage) that you need for your application or task, either as **specs** or as an **instance type**.
   - For **as specs**, specify the required number of vCPUs and amount of memory.
   - For **as an instance type**, accept the default instance type, or choose **Change instance type** to choose a different instance type.

d. For **Network**, choose an existing VPC or create a new one.

   [Existing VPC] Choose the VPC.

   [New VPC] Choose **Create new VPC** to go to 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 enable Amazon EBS optimization, for **EBS-optimized**, choose **Launch EBS-optimized instances**.

   b. (Optional) To add temporary block-level storage for your instances, for **Instance store**, choose **Attach at launch**.

   c. (Optional) To add storage, specify additional instance store volumes or Amazon EBS volumes, depending on the instance type.

   d. (Optional) By default, basic monitoring is enabled for your instances. To enable detailed monitoring, for **Monitoring**, choose **Enable CloudWatch detailed monitoring**.

   e. (Optional) To replace unhealthy Spot Instances, for **Health check**, choose **Replace unhealthy instances**. To enable this option, you must first choose **Maintain target capacity**.

   f. (Optional) To run a Dedicated Spot Instance, for **Tenancy**, choose **Dedicated - run a dedicated instance**.

   g. (Optional) For **Security groups**, choose one or more security groups or create a new one.

      [Existing security group] Choose one or more security groups.

      [New security group] Choose **Create new security group** to go the Amazon VPC console. When you are done, return to the wizard and refresh the list.

   h. (Optional) To make your instances reachable from the internet, for **Auto-assign IPv4 Public IP**, choose **Enable**.

   i. (Optional) To launch your Spot Instances with an IAM role, for **IAM instance profile**, choose the role.

   j. (Optional) To run a start-up script, copy it to **User data**.

   k. (Optional) To add a tag, choose **Add new tag** and enter the key and value for the tag. Repeat for each tag.

      For each tag, to tag the instances and the Spot Fleet request with the same tag, ensure that both **Instance** and **Fleet** are selected. To tag only the instances launched by the fleet, clear **Fleet**. To tag only the Spot Fleet request, clear **Instance**.

6. For **Tell us how much capacity you need**, do the following:

   a. For **Total target capacity**, specify the number of units to request for target capacity. You can choose instances or vCPUs. To specify a target capacity of 0 so that you can add capacity later, choose **Maintain target capacity**.

   b. (Optional) For **Optional On-Demand portion**, specify the number of On-Demand units to request. The number must be less than the **Total target capacity**. Amazon EC2 calculates the difference, and allocates the difference to Spot units to request.

      **Important**
      To specify an optional On-Demand portion, you must first choose a launch template.

   c. (Optional) By default, the Spot service terminates Spot Instances when they are interrupted. To maintain the target capacity, select **Maintain target capacity**. You can then specify that the Spot service terminates, stops, or hibernates Spot Instances when they are interrupted. To do so, choose the corresponding option from **Interruption behavior**.

   d. (Optional) To allow Spot Fleet to launch a replacement Spot Instance when an instance rebalance notification is emitted for an existing Spot Instance in the fleet, select **Capacity rebalance**. For more information, see **Capacity Rebalancing (p. 714)**.
Note
When a replacement instance is launched, the instance marked for rebalance is not automatically terminated. You can terminate it, or you can leave it running. You are charged for both instances while they are running. The instance marked for rebalance is at an elevated risk of interruption, and you will receive a two-minute Spot Instance interruption notice before Amazon EC2 interrupts it.

- (Optional) To control the amount you pay per hour for all the Spot Instances in your fleet, select Set maximum cost for Spot Instances and then enter the maximum total amount you're willing to pay per hour. When the maximum total amount is reached, Spot Fleet stops launching Spot Instances even if it hasn't met the target capacity. For more information, see Control spending (p. 716).

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) For Fleet allocation strategy, choose the strategy that meets your needs. For more information, see Allocation strategy for Spot Instances (p. 712).
   c. (Optional) To remove instance types, for Fleet request, select the instance types to remove and then choose Delete. To add instance types, choose Select instance types.

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.

Create a Spot Fleet using the AWS CLI

To create a Spot Fleet request using the AWS CLI

- Use the request-spot-fleet command to create a Spot Fleet request.

```bash
aws ec2 request-spot-fleet --spot-fleet-request-config file:///config.json
```

For example configuration files, see Spot Fleet example configurations (p. 780).

The following is example output:
Tag a Spot Fleet

To help categorize and manage your Spot Fleet requests, you can tag them with custom metadata. You can assign a tag to a Spot Fleet request when you create it, or afterward. You can assign tags using the Amazon EC2 console or a command line tool.

When you tag a Spot Fleet request, the instances and volumes that are launched by the Spot Fleet are not automatically tagged. You need to explicitly tag the instances and volumes launched by the Spot Fleet. You can choose to assign tags to only the Spot Fleet request, or to only the instances launched by the fleet, or to only the volumes attached to the instances launched by the fleet, or to all three.

**Note**
Volume tags are only supported for volumes that are attached to On-Demand Instances. You can't tag volumes that are attached to Spot Instances.

For more information about how tags work, see Tag your Amazon EC2 resources (p. 1450).

Contents

- Prerequisite (p. 729)
- Tag a new Spot Fleet (p. 730)
- Tag a new Spot Fleet and the instances and volumes that it launches (p. 731)
- Tag an existing Spot Fleet (p. 733)
- View Spot Fleet request tags (p. 733)

Prerequisite

Grant the IAM user the permission to tag resources. For more information, see Example: Tag resources (p. 1097).

To grant an IAM user the permission to tag resources

Create a IAM policy that includes the following:

- The `ec2:CreateTags` action. This grants the IAM user permission to create tags.
- The `ec2:RequestSpotFleet` action. This grants the IAM user permission to create a Spot Fleet request.
- For `Resource`, you must specify "/". This allows users to tag all resource types.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TagSpotFleetRequest",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags",
                "ec2:RequestSpotFleet"
            ],
            "Resource": "*"
        }
    ]
}
```
Tag a new Spot Fleet

To tag a new Spot Fleet request using the console

1. Follow the Create a Spot Fleet request using defined parameters (console) (p. 726) procedure.
2. To add a tag, expand Additional configurations, choose Add new tag, and enter the key and value for the tag. Repeat for each tag.

   For each tag, you can tag the Spot Fleet request and the instances with the same tag. To tag both, ensure that both Instance tags and Fleet tags are selected. To tag only the Spot Fleet request, clear Instance tags. To tag only the instances launched by the fleet, clear Fleet tags.
3. Complete the required fields to create a Spot Fleet request, and then choose Launch. For more information, see Create a Spot Fleet request using defined parameters (console) (p. 726).

To tag a new Spot Fleet request using the AWS CLI

To tag a Spot Fleet request when you create it, configure the Spot Fleet request configuration as follows:

- Specify the tags for the Spot Fleet request in SpotFleetRequestConfig.
- For ResourceType, specify spot-fleet-request. If you specify another value, the fleet request will fail.
- For Tags, specify the key-value pair. You can specify more than one key-value pair.

In the following example, the Spot Fleet request is tagged with two tags: Key=Environment and Value=Production, and Key=Cost-Center and Value=123.

```json
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-role",
        "LaunchSpecifications": [
            {
                "ImageId": "ami-0123456789EXAMPLE",
                "InstanceType": "c4.large"
            }
        ],
        "SpotPrice": "5",
        "TargetCapacity": 2,
        "TerminateInstancesWithExpiration": true,
        "Type": "maintain",
        "ReplaceUnhealthyInstances": true,
        "InstanceInterruptionBehavior": "terminate",
    }
}
```
Tag a new Spot Fleet and the instances and volumes that it launches

To tag a new Spot Fleet request and the instances and volumes that it launches using the AWS CLI

To tag a Spot Fleet request when you create it, and to tag the instances and volumes when they are launched by the fleet, configure the Spot Fleet request configuration as follows:

**Spot Fleet request tags:**

- Specify the tags for the Spot Fleet request in SpotFleetRequestConfig.
- For ResourceType, specify spot-fleet-request. If you specify another value, the fleet request will fail.
- For Tags, specify the key-value pair. You can specify more than one key-value pair.

**Instance tags:**

- Specify the tags for the instances in LaunchSpecifications.
- For ResourceType, specify instance. If you specify another value, the fleet request will fail.
- For Tags, specify the key-value pair. You can specify more than one key-value pair.

Alternatively, you can specify the tags for the instance in the launch template (p. 399) that is referenced in the Spot Fleet request.

**Volume tags:**

- Specify the tags for the volumes in the launch template (p. 399) that is referenced in the Spot Fleet request. Volume tagging in LaunchSpecifications is not supported.

In the following example, the Spot Fleet request is tagged with two tags: Key=Environment and Value=Production, and Key=Cost-Center and Value=123. The instances that are launched by the fleet are tagged with one tag (which is the same as one of the tags for the Spot Fleet request): Key=Cost-Center and Value=123.

```json
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-role",
        "InstancePoolsToUseCount": 1,
        "TagSpecifications": [
            {
                "ResourceType": "spot-fleet-request",
                "Tags": [
                    {
                        "Key": "Environment",
                        "Value": "Production"
                    },
                    {
                        "Key": "Cost-Center",
                        "Value": "123"
                    }
                ]
            }
        ]
    }
}
```
"LaunchSpecifications": [ 
  { 
    "ImageId": "ami-0123456789EXAMPLE",
    "InstanceType": "c4.large",
    "TagSpecifications": [ 
      { 
        "ResourceType": "instance",
        "Tags": [ 
          { 
            "Key": "Cost-Center",
            "Value": "123"
          }
        ]
      }
    ],
    "SpotPrice": "5",
    "TargetCapacity": 2,
    "TerminateInstancesWithExpiration": true,
    "Type": "maintain",
    "ReplaceUnhealthyInstances": true,
    "InstanceInterruptionBehavior": "terminate",
    "InstancePoolsToUseCount": 1,
    "TagSpecifications": [ 
      { 
        "ResourceType": "spot-fleet-request",
        "Tags": [ 
          { 
            "Key": "Environment",
            "Value": "Production"
          },
          { 
            "Key": "Cost-Center",
            "Value": "123"
          }
        ]
      }
    ]
  },
]
}

To tag instances launched by a Spot Fleet using the AWS CLI

To tag instances when they are launched by the fleet, you can either specify the tags in the launch template (p. 399) that is referenced in the Spot Fleet request, or you can specify the tags in the Spot Fleet request configuration as follows:

- Specify the tags for the instances in LaunchSpecifications.
- For ResourceType, specify instance. If you specify another value, the fleet request will fail.
- For Tags, specify the key-value pair. You can specify more than one key-value pair.

In the following example, the instances that are launched by the fleet are tagged with one tag: Key=Cost-Center and Value=123.
To tag volumes attached to On-Demand Instances launched by a Spot Fleet using the AWS CLI

To tag volumes when they are created by the fleet, you must specify the tags in the launch template (p. 399) that is referenced in the Spot Fleet request.

**Note**
- Volume tags are only supported for volumes that are attached to On-Demand Instances. You can’t tag volumes that are attached to Spot Instances.
- Volume tagging in LaunchSpecifications is not supported.

### Tag an existing Spot Fleet

#### To tag an existing Spot Fleet request using the console

After you have created a Spot Fleet request, you can add tags to the fleet request using the console.

2. Select your Spot Fleet request.
3. Choose the **Tags** tab and choose **Create Tag**.

#### To tag an existing Spot Fleet request using the AWS CLI

You can use the `create-tags` command to tag existing resources. In the following example, the existing Spot Fleet request is tagged with Key=purpose and Value=test.

```
aws ec2 create-tags
  --resources sfr-11112222-3333-4444-5555-66666EXAMPLE
  --tags Key=purpose,Value=test
```

### View Spot Fleet request tags

#### To view Spot Fleet request tags using the console

2. Select your Spot Fleet request and choose the **Tags** tab.
To describe Spot Fleet request tags

Use the `describe-tags` command to view the tags for the specified resource. In the following example, you describe the tags for the specified Spot Fleet request.

```bash
aws ec2 describe-tags \
  --filters "Name=resource-id,Values=sfr-11112222-3333-4444-5555-66666EXAMPLE"
```

```
{
  "Tags": [
    {
      "Key": "Environment",
      "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
      "ResourceType": "spot-fleet-request",
      "Value": "Production"
    },
    {
      "Key": "Another key",
      "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
      "ResourceType": "spot-fleet-request",
      "Value": "Another value"
    }
  ]
}
```

You can also view the tags of a Spot Fleet request by describing the Spot Fleet request.

Use the `describe-spot-fleet-requests` command to view the configuration of the specified Spot Fleet request, which includes any tags that were specified for the fleet request.

```bash
aws ec2 describe-spot-fleet-requests \
  --spot-fleet-request-ids sfr-11112222-3333-4444-5555-66666EXAMPLE
```

```
{
  "SpotFleetRequestConfigs": [
    {
      "ActivityStatus": "fulfilled",
      "CreateTime": "2020-02-13T02:49:19.709Z",
      "SpotFleetRequestConfig": {
        "AllocationStrategy": "capacityOptimized",
        "OnDemandAllocationStrategy": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "Default",
        "FulfilledCapacity": 2.0,
        "OnDemandFulfilledCapacity": 0.0,
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-role",
        "LaunchSpecifications": [
          {
            "ImageId": "ami-0123456789EXAMPLE",
            "InstanceType": "c4.large"
          }
        ],
        "TargetCapacity": 2,
        "OnDemandTargetCapacity": 0,
        "Type": "maintain",
        "ReplaceUnhealthyInstances": false,
        "InstanceInterruptionBehavior": "terminate"
      },
      "SpotFleetRequestId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
      "SpotFleetRequestState": "active",
      "Tags": [}
```
Monitor your Spot Fleet

The Spot Fleet launches Spot Instances when your maximum price exceeds the Spot price and capacity is available. The Spot Instances run until they are interrupted or you terminate them.

To monitor your Spot Fleet (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Spot Requests.
3. Select your Spot Fleet request. To see the configuration details, choose Description.
4. To list the Spot Instances for the Spot Fleet, choose Instances.
5. To view the history for the Spot Fleet, choose History.

To monitor your Spot Fleet (AWS CLI)

Use the describe-spot-fleet-requests command to describe your Spot Fleet requests.

```bash
aws ec2 describe-spot-fleet-requests
```

Use the describe-spot-fleet-instances command to describe the Spot Instances for the specified Spot Fleet.

```bash
aws ec2 describe-spot-fleet-instances \
--spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Use the describe-spot-fleet-request-history command to describe the history for the specified Spot Fleet request.

```bash
aws ec2 describe-spot-fleet-request-history \
--spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
--start-time 2015-05-18T00:00:00Z
```

Modify a Spot Fleet request

You can modify an active Spot Fleet request to complete the following tasks:

- Increase the target capacity and On-Demand portion
- Decrease the target capacity and On-Demand portion

**Note**

You can't modify a one-time Spot Fleet request. You can only modify a Spot Fleet request if you selected Maintain target capacity when you created the Spot Fleet request.
When you increase the target capacity, the Spot Fleet launches additional Spot Instances. When you increase the On-Demand portion, the Spot Fleet launches additional On-Demand Instances.

When you increase the target capacity, the Spot Fleet launches the additional Spot Instances according to the allocation strategy for its Spot Fleet request. If the allocation strategy is lowestPrice, the Spot Fleet launches the instances from the lowest-priced Spot capacity 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)**

2. Select your Spot Fleet request.
3. Choose Actions, Modify target capacity.
4. In Modify target capacity, do the following:
   a. Enter the new target capacity and On-Demand portion.
   b. (Optional) If you are decreasing the target capacity but want to keep the fleet at its current size, clear Terminate instances.
   c. Choose Submit.

**To modify a Spot Fleet request using the AWS CLI**

Use the modify-spot-fleet-request command to update the target capacity of the specified Spot Fleet request.

```
aws ec2 modify-spot-fleet-request \
  --spot-fleet-request-id sfr-73fbd2ce-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-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
  --target-capacity 10  \
  --excess-capacity-termination-policy NoTermination
```

**Cancel 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)

2. Select your Spot Fleet request.
3. Choose Actions, Cancel spot request.
4. In Cancel spot request, verify that you want to cancel the Spot Fleet. To keep the fleet at its current size, clear Terminate instances. When you are ready, choose Confirm.

To cancel a Spot Fleet request using the AWS CLI

Use the cancel-spot-fleet-requests command to cancel the specified Spot Fleet request and terminate the instances.

```
aws ec2 cancel-spot-fleet-requests \
   --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
   --terminate-instances
```

The following is example output:

```
{
   "SuccessfulFleetRequests": [
      {
         "SpotFleetRequestId": "sfr-73fbd2ce-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-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
   --no-terminate-instances
```

The following is example output:

```
{
   "SuccessfulFleetRequests": [
      {
         "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
         "CurrentSpotFleetRequestState": "cancelled_running",
         "PreviousSpotFleetRequestState": "active"
      }
   ],
   "UnsuccessfulFleetRequests": []
}
```

CloudWatch metrics for Spot Fleet

Amazon EC2 provides Amazon CloudWatch metrics that you can use to monitor your Spot Fleet.
Important
To ensure accuracy, we recommend that you enable detailed monitoring when using these metrics. For more information, see Enable or turn off detailed monitoring for your instances (p. 840).

For more information about CloudWatch metrics provided by Amazon EC2, see Monitor your instances using CloudWatch (p. 839).

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. 842).

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AvailableInstancePoolsCount</td>
<td>The Spot capacity pools specified in the Spot Fleet request.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td>BidsSubmittedForCapacity</td>
<td>The capacity for which Amazon EC2 has submitted Spot Fleet requests.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td>EligibleInstancePoolCount</td>
<td>The Spot capacity pools specified in the Spot Fleet request where Amazon EC2 can fulfill requests. Amazon EC2 does not fulfill requests in pools where the maximum price you're willing to pay for Spot Instances is less than the Spot price or the Spot price is greater than the price for On-Demand Instances.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td>FulfilledCapacity</td>
<td>The capacity that Amazon EC2 has fulfilled.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td>MaxPercentCapacityAllocation</td>
<td>The maximum value of PercentCapacityAllocation across all Spot Fleet pools specified in the Spot Fleet request.</td>
</tr>
<tr>
<td></td>
<td>Units: Percent</td>
</tr>
<tr>
<td>PendingCapacity</td>
<td>The difference between TargetCapacity and FulfilledCapacity.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td>PercentCapacityAllocation</td>
<td>The capacity allocated for the Spot capacity pool for the specified dimensions. To get the maximum value recorded across all Spot capacity pools, use MaxPercentCapacityAllocation.</td>
</tr>
<tr>
<td></td>
<td>Units: Percent</td>
</tr>
<tr>
<td>TargetCapacity</td>
<td>The target capacity of the Spot Fleet request.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td>TerminatingCapacity</td>
<td>The capacity that is being terminated because the provisioned capacity is greater than the target capacity.</td>
</tr>
</tbody>
</table>
CloudWatch metrics for Spot Fleet

Metric | Description | 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.

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AvailabilityZone</td>
<td>Filter the data by Availability Zone.</td>
</tr>
<tr>
<td>FleetRequestId</td>
<td>Filter the data by Spot Fleet request.</td>
</tr>
<tr>
<td>InstanceType</td>
<td>Filter the data by instance type.</td>
</tr>
</tbody>
</table>

**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**

2. In the navigation pane, choose **Metrics**.
3. Choose the **EC2 Spot** namespace.
   
   **Note**
   If the **EC2 Spot** namespace is not displayed, there are two reasons for this. Either you’ve not yet used Spot Fleet—only the AWS services that you’re using send metrics to Amazon CloudWatch. Or, if you’ve not used Spot Fleet for the past two weeks, the namespace does not appear.

4. (Optional) To filter the metrics by dimension, select one of the following:
   - **Fleet Request Metrics** – Group by Spot Fleet request
   - **By Availability Zone** – Group by Spot Fleet request and Availability Zone
   - **By Instance Type** – Group by Spot Fleet request and instance type
   - **By Availability Zone/Instance Type** – Group by Spot Fleet request, Availability Zone, and instance type
5. To view the data for a metric, select the check box next to the metric.
Automatic scaling for Spot Fleet

*Automatic scaling* is the ability to increase or decrease the target capacity of your Spot Fleet automatically based on demand. A Spot Fleet can either launch instances (scale out) or terminate instances (scale in), within the range that you choose, in response to one or more scaling policies.

Spot Fleet supports the following types of automatic scaling:

- **Target tracking scaling** *(p. 742)* – 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. 743)* – 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. 744)* – Increase or decrease the current capacity of the fleet based on the date and time.

If you are using *instance weighting* *(p. 717)*, keep in mind that Spot Fleet can exceed the target capacity as needed. Fulfilled capacity can be a floating-point number but target capacity must be an integer, so Spot Fleet rounds up to the next integer. You must take these behaviors into account when you look at the outcome of a scaling policy when an alarm is triggered. For example, suppose that the target capacity is 30, the fulfilled capacity is 30.1, and the scaling policy subtracts 1. When the alarm is triggered, the automatic scaling process subtracts 1 from 30.1 to get 29.1 and then rounds it up to 30, so no scaling action is taken. As another example, suppose that you selected instance weights of 2, 4, and 8, and a target capacity of 10, but no weight 2 instances were available so Spot Fleet provisioned instances of weights 4 and 8 for a fulfilled capacity of 12. If the scaling policy decreases target capacity by 20% and an alarm is triggered, the automatic scaling process subtracts 12*0.2 from 12 to get 9.6 and then rounds it up to 10, so no scaling action is taken.

The scaling policies that you create for Spot Fleet support a cooldown period. This is the number of seconds after a scaling activity completes where previous trigger-related scaling activities can influence future scaling events. For scale-out policies, while the cooldown period is in effect, the capacity that has been added by the previous scale-out event that initiated the cooldown is calculated as part of the desired capacity for the next scale out. The intention is to continuously (but not excessively) scale out. For scale in policies, the cooldown period is used to block subsequent scale in requests until it has expired. The intention is to scale in conservatively to protect your application’s availability. However, if another alarm triggers a scale-out policy during the cooldown period after a scale-in, automatic scaling scales out your scalable target immediately.
We recommend that you scale based on instance metrics with a 1-minute frequency because that ensures a faster response to utilization changes. Scaling on metrics with a 5-minute frequency can result in slower response time and scaling on stale metric data. To send metric data for your instances to CloudWatch in 1-minute periods, you must specifically enable detailed monitoring. For more information, see Enable or turn off detailed monitoring for your instances (p. 840) and Create a Spot Fleet request using defined parameters (console) (p. 726).

For more information about configuring scaling for Spot Fleet, see the following resources:

- application-autoscaling section of the AWS CLI Command Reference
- Application Auto Scaling API Reference
- Application Auto Scaling User Guide

IAM permissions required for Spot Fleet automatic scaling

Automatic scaling for Spot Fleet is made possible by a combination of the Amazon EC2, Amazon CloudWatch, and Application Auto Scaling APIs. Spot Fleet requests are created with Amazon EC2, alarms are created with CloudWatch, and scaling policies are created with Application Auto Scaling.

In addition to the IAM permissions for Spot Fleet (p. 721) and Amazon EC2, the IAM user that accesses fleet scaling settings must have the appropriate permissions for the services that support dynamic scaling. IAM users must have permissions to use the actions shown in the following example policy.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "application-autoscaling:*",
            "ec2:DescribeSpotFleetRequests",
            "ec2:ModifySpotFleetRequest",
            "cloudwatch:DeleteAlarms",
            "cloudwatch:DescribeAlarmHistory",
            "cloudwatch:DescribeAlarms",
            "cloudwatch:DescribeAlarmsForMetric",
            "cloudwatch:GetMetricStatistics",
            "cloudwatch:ListMetrics",
            "cloudwatch:PutMetricAlarm",
            "cloudwatch:DisableAlarmActions",
            "cloudwatch:EnableAlarmActions",
            "iam:CreateServiceLinkedRole",
            "sns:CreateTopic",
            "sns:Subscribe",
            "sns:Get*",
            "sns:List*"
         ],
         "Resource": "*"
      }
   ]
}
```

You can also create your own IAM policies that allow more fine-grained permissions for calls to the Application Auto Scaling API. For more information, see Authentication and Access Control in the Application Auto Scaling User Guide.

The Application Auto Scaling service also needs permission to describe your Spot Fleet and CloudWatch alarms, and permissions to modify your Spot Fleet target capacity on your behalf. If you enable automatic scaling for your Spot Fleet, it creates a service-linked role named
Scale Spot Fleet using a target tracking policy

With target tracking scaling policies, you select a metric and set a target value. Spot Fleet creates and manages the CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value. The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value. In addition to keeping the metric close to the target value, a target tracking scaling policy also adjusts to the fluctuations in the metric due to a fluctuating load pattern and minimizes rapid fluctuations in the capacity of the fleet.

You can create multiple target tracking scaling policies for a Spot Fleet, provided that each of them uses a different metric. The fleet scales based on the policy that provides the largest fleet capacity. This enables you to cover multiple scenarios and ensure that there is always enough capacity to process your application workloads.

To ensure application availability, the fleet scales out proportionally to the metric as fast as it can, but scales in more gradually.

When a Spot Fleet terminates an instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

Do not edit or delete the CloudWatch alarms that Spot Fleet manages for a target tracking scaling policy. Spot Fleet deletes the alarms automatically when you delete the target tracking scaling policy.

Limitation

The Spot Fleet request must have a request type of maintain. Automatic scaling is not supported for requests of type request, or Spot blocks.

To configure a target tracking policy (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Spot Requests.
3. Select your Spot Fleet request and choose Auto Scaling.
4. If automatic scaling is not configured, choose Configure.
5. Use Scale capacity between to set the minimum and maximum capacity for your fleet. Automatic scaling does not scale your fleet below the minimum capacity or above the maximum capacity.
6. For Policy name, enter a name for the policy.
7. Choose a Target metric.
8. Enter a Target value for the metric.
9. (Optional) Set Cooldown period to modify the default cooldown period.
10. (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.
11. Choose Save.

To configure a target tracking policy 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.
Scale Spot Fleet using step scaling policies

With step scaling policies, you specify CloudWatch alarms to trigger the scaling process. For example, if you want to scale out when CPU utilization reaches a certain level, create an alarm using the CPUUtilization metric provided by Amazon EC2.

When you create a step scaling policy, you must specify one of the following scaling adjustment types:

- **Add** – Increase the target capacity of the fleet by a specified number of capacity units or a specified percentage of the current capacity.
- **Remove** – Decrease the target capacity of the fleet by a specified number of capacity units or a specified percentage of the current capacity.
- **Set to** – Set the target capacity of the fleet to the specified number of capacity units.

When an alarm is triggered, the automatic scaling process calculates the new target capacity using the fulfilled capacity and the scaling policy, and then updates the target capacity accordingly. For example, suppose that the target capacity and fulfilled capacity are 10 and the scaling policy adds 1. When the alarm is triggered, the automatic scaling process adds 1 to 10 to get 11, so Spot Fleet launches 1 instance.

When a Spot Fleet terminates an instance because the target capacity was decreased, the instance receives a Spot Instance interruption notice.

**Limitation**

The Spot Fleet request must have a request type of maintain. Automatic scaling is not supported for requests of type request, or Spot blocks.

**Prerequisites**

- Consider which CloudWatch metrics are important to your application. You can create CloudWatch alarms based on metrics provided by AWS or your own custom metrics.
- For the AWS metrics that you will use in your scaling policies, enable CloudWatch metrics collection if the service that provides the metrics does not enable it by default.

**To create a CloudWatch alarm**

2. In the navigation pane, choose Alarms.
3. Choose Create alarm.
4. On the Specify metric and conditions page, choose Select metric.
5. Choose EC2 Spot, Fleet Request Metrics, select a metric (for example, CPUUtilization), and then choose Select metric.

   The Specify metric and conditions page appears, showing a graph and other information about the metric you selected.
6. For Period, choose the evaluation period for the alarm, for example, 1 minute. When evaluating the alarm, each period is aggregated into one data point.
   
   **Note**
   
   A shorter period creates a more sensitive alarm.
7. For Conditions, define the alarm by defining the threshold condition. For example, you can define a threshold to trigger the alarm whenever the value of the metric is greater than or equal to 80 percent.
8. Under **Additional configuration**, for **Datapoints to alarm**, specify how many datapoints (evaluation periods) must be in the ALARM state to trigger the alarm, for example, 1 evaluation period or 2 out of 3 evaluation periods. This creates an alarm that goes to ALARM state if that many consecutive periods are breaching. For more information, see **Evaluating an Alarm** in the Amazon CloudWatch User Guide.

9. For **Missing data treatment**, choose one of the options (or leave the default of **Treat missing data as missing**). For more information, see Configuring How CloudWatch Alarms Treat Missing Data in the Amazon CloudWatch User Guide.

10. Choose **Next**.

11. (Optional) To receive notification of a scaling event, for **Notification**, you can choose or create the Amazon SNS topic you want to use to receive notifications. Otherwise, you can delete the notification now and add one later as needed.

12. Choose **Next**.

13. Under **Add a description**, enter a name and description for the alarm and choose **Next**.

14. Choose **Create alarm**.

**To configure a step scaling policy for your Spot Fleet (console)**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Spot Requests**.
3. Select your Spot Fleet request and choose **Auto Scaling**.
4. If automatic scaling is not configured, choose **Configure**.
5. Use **Scale capacity between** to set the minimum and maximum capacity for your fleet. Automatic scaling does not scale your fleet below the minimum capacity or above the maximum capacity.
6. Initially, **Scaling policies** contains policies named ScaleUp and ScaleDown. You can complete these policies, or choose **Remove policy** to delete them. You can also choose **Add policy**.

7. To define a policy, do the following:
   a. For **Policy name**, enter 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 an existing 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.

You can only create a scheduled action for Spot Fleets that already exist. You can't create a scheduled
action at the same time that you create a Spot Fleet.

Limitation

The Spot Fleet request must have a request type of maintain. Automatic scaling is not supported for
requests of type request, or Spot blocks.

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 the Scheduled Scaling tab near the bottom of the screen.
5. For Name, specify a name for the scheduled action.
6. Enter 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 the Scheduled Scaling tab near the bottom of the screen.
4. For Recurrence, choose one of the predefined schedules (for example, Every day), or choose
   Custom and enter 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 the Scheduled Scaling tab near the bottom of the screen.
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 the Scheduled Scaling tab near the bottom of the screen.
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

Monitor fleet events using Amazon EventBridge

When the state of an EC2 Fleet or Spot Fleet changes, the fleet emits a notification. The notification is made available as an event that is sent to Amazon EventBridge (formerly known as Amazon CloudWatch Events). Events are emitted on a best effort basis.

With Amazon EventBridge, you can create rules that trigger programmatic actions in response to an event. For example, you can create two EventBridge rules, one that's triggered when a fleet state changes, and one that's triggered when an instance in the fleet is terminated. You can configure the first rule so that, if the fleet state changes, the rule invokes an SNS topic to send an email notification to you. You can configure the second rule so that, if an instance is terminated, the rule invokes a Lambda function to launch a new instance.

Topics

- EC2 Fleet event types (p. 746)
- Spot Fleet event types (p. 751)
- Create Amazon EventBridge rules (p. 755)

EC2 Fleet event types

Note

Only fleets of type maintain and request emit events. Fleets of type instant do not emit events because they submit synchronous one-time requests, and the state of the fleet is known immediately in the response.

There are five EC2 Fleet event types. For each event type, there are several sub-types.

The events are sent to EventBridge in JSON format. The following fields in the event form the event pattern that is defined in the rule, and which trigger an action:

"source": "aws.ec2fleet"

Identifies that the event is from EC2 Fleet.

"detail-type": "$EC2 Fleet State Change"

Identifies the event type.

"detail": { "sub-type": "submitted" }

Identifies the event sub-type.
EC2 Fleet State Change

EC2 Fleet sends an EC2 Fleet State Change event to Amazon EventBridge when an EC2 Fleet changes state.

The following is example data for this event.

```json
{
  "version": "0",
  "id": "715ed6b3-b8fc-27fe-fad6-528c7b8bf8a2",
  "detail-type": "EC2 Fleet State Change",
  "source": "aws.ec2fleet",
  "account": "123456789012",
  "time": "2020-11-09T09:00:20Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-598fb973-87b7-422d-be4d-6b0809bfff0a"
  ],
  "detail": {
    "sub-type": "active"
  }
}
```

The possible values for `sub-type` are:

- **submitted**
  - The EC2 Fleet request is being evaluated and Amazon EC2 is preparing to launch the target number of instances.

- **active**
  - The EC2 Fleet request has been validated and Amazon EC2 is attempting to maintain the target number of running Spot Instances.

- **progress**
  - The EC2 Fleet request is in the process of being fulfilled.

- **cancelled_terminating**
  - The EC2 Fleet request is deleted and its instances are terminating. The request remains in this state until all instances are terminated.

- **cancelled_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.

- **cancelled**
  - The EC2 Fleet request is deleted and has no running instances. The EC2 Fleet will be deleted two days after its instances are terminated.

- **modify_in_progress**
  - The EC2 Fleet request is being modified. The request remains in this state until the modification is fully processed or the EC2 Fleet request is deleted.
modify_succeeded

The EC2 Fleet request was modified. This state does not apply to instant fleets because instant fleets cannot be modified.

expired

The EC2 Fleet request has expired. If the request was created with TerminateInstancesWithExpiration set, a subsequent event indicates that the instances are terminated.

**EC2 Fleet Spot Instance Request Change**

EC2 Fleet sends an **EC2 Fleet Spot Instance Request Change** event to Amazon EventBridge when a Spot Instance request in the fleet changes state.

The following is example data for this event.

```
{
  "version": "0",
  "id": "19331f74-bf4b-a3dd-0f1b-ddb1422032b9",
  "detail-type": "EC2 Fleet Spot Instance Request Change",
  "source": "aws.ec2fleet",
  "account": "123456789012",
  "time": "2020-11-09T09:00:05Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-83fd4e48-552a-40ef-9532-82a3acca5f10"
  ],
  "detail": {
    "spot-instance-request-id": "sir-rmqske6h",
    "description": "SpotInstanceRequestId sir-rmqske6h, PreviousState: cancelled_running",
    "sub-type": "cancelled"
  }
}
```

The possible values for **sub-type** are:

- **submitted**
  
  The request is submitted.

- **disabled**
  
  You stopped the Spot Instance.

- **active**
  
  The request is fulfilled and has an associated Spot Instance.

- **cancelled**
  
  You cancelled the request, or the request expired.

**EC2 Fleet Instance Change**

EC2 Fleet sends an **EC2 Fleet Instance Change** event to Amazon EventBridge when an instance in the fleet changes state.
The following is example data for this event.

```
{
  "version": "0",
  "id": "542ce428-c8f1-0608-c015-e8ed6522c5bc",
  "detail-type": "EC2 Fleet Instance Change",
  "source": "aws.ec2fleet",
  "account": "123456789012",
  "time": "2020-11-09T09:00:23Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-598fb973-87b7-422d-be4d-6b0809b0ff0a"
  ],
  "detail": {
    "instance-id": "i-0c594155dd5ff1829",
    "description": "{"instanceType":"c5.large","image":"ami-6057e21a","productDescription":"Linux/UNIX","availabilityZone":"us-east-1"},
    "sub-type": "launched"
  }
}
```

The possible values for `sub-type` are:

- **launched**
  
  A new instance was launched.

- **terminated**
  
  The instance was terminated.

- **termination_notified**
  
  An instance termination notification was sent.

### EC2 Fleet Information

EC2 Fleet sends an EC2 Fleet Information event to Amazon EventBridge when there is an error during fulfillment. The information event does not block the fleet from attempting to fulfill its target capacity.

The following is example data for this event.

```
{
  "version": "0",
  "id": "76529817-d605-4571-7224-d36cc1b2c0c4",
  "detail-type": "EC2 Fleet Information",
  "source": "aws.ec2fleet",
  "account": "123456789012",
  "time": "2020-11-09T08:17:07Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-8becf5fe-bb9e-415d-8f54-3fa5a8628b91"
  ],
  "detail": {
    "description": "r3.8xlarge, ami-032930428bf1abbff, Linux/UNIX, us-east-1a, Spot bid price is less than Spot market price $0.5291",
    "sub-type": "launchSpecUnusable"
  }
}
```
The possible values for sub-type are:

launchSpecUnusable

The price in a launch specification is not valid because it is below the Spot price or the Spot price is above the On-Demand price.

classProgressHalted

The price in every launch specification is not valid. A launch specification might become valid if the Spot price changes.

registerWithLoadBalancersFailed

An attempt to register instances with load balancers failed. For more information, see the description of the event.

launchSpecTemporarilyBlacklisted

The configuration is not valid and several attempts to launch instances have failed. For more information, see the description of the event.

EC2 Fleet Error

EC2 Fleet sends an EC2 Fleet Error event to Amazon EventBridge when there is an error during fulfillment. The error event blocks the fleet from attempting to fulfill its target capacity.

The following is example data for this event.

```
{
    "version": "0",
    "id": "69849a22-6d0f-d4ce-602b-b47c1c98240e",
    "detail-type": "EC2 Fleet Error",
    "source": "aws.ec2fleet",
    "account": "123456789012",
    "time": "2020-10-07T01:44:24Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:fleet/fleet-9bb19bc6-60d3-4fd2-ae47-d33e68eafa08"
    ],
    "detail": {
        "description": "m3.large, ami-00068cd7555f543d5, Linux/UNIX: IPv6 is not supported for the instance type 'm3.large'. ",
        "sub-type": "spotFleetRequestConfigurationInvalid"
    }
}
```

The possible values for sub-type are:

allLaunchSpecsTemporarilyBlacklisted

None of the configurations are valid, and several attempts to launch instances have failed. For more information, see the description of the event.

spotFleetRequestConfigurationInvalid

The configuration is not valid. For more information, see the description of the event.

spotInstanceCountLimitExceeded

You've reached the limit on the number of Spot Instances that you can launch.
Spot Fleet event types

There are five Spot Fleet event types. For each event type, there are several sub-types.

The events are sent to EventBridge in JSON format. The following fields in the event form the event pattern that is defined in the rule, and which trigger an action:

"source": "aws.ec2spotfleet"

Identifies that the event is from Spot Fleet.

"detail-type": "EC2 Spot Fleet State Change"

Identifies the event type.

"detail": { "sub-type": "submitted" }

Identifies the event sub-type.

**Event types**

- EC2 Spot Fleet State Change (p. 751)
- EC2 Spot Fleet Spot Instance Request Change (p. 752)
- EC2 Spot Fleet Instance Change (p. 753)
- EC2 Spot Fleet Information (p. 754)
- EC2 Spot Fleet Error (p. 754)

**EC2 Spot Fleet State Change**

Spot Fleet sends an EC2 Spot Fleet State Change event to Amazon EventBridge when a Spot Fleet changes state.

The following is example data for this event.

```
{
    "version": "0",
    "id": "d1af1091-6cc3-2e24-203a-3b870e455d5b",
    "detail-type": "EC2 Spot Fleet State Change",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-09T08:57:06Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-4b6d274d-0cea-4b2c-b3be-9dc627ad1f55"
    ],
    "detail": {
        "sub-type": "submitted"
    }
}
```

The possible values for `sub-type` are:

- `submitted`

The Spot Fleet request is being evaluated and Amazon EC2 is preparing to launch the target number of instances.
active

The Spot Fleet request has been validated and Amazon EC2 is attempting to maintain the target number of running Spot Instances.

progress

The Spot Fleet request is in the process of being fulfilled.

cancelled_terminating

The Spot Fleet request is deleted and its instances are terminating. The request remains in this state until all instances are terminated.

cancelled_running

The Spot 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.

cancelled

The Spot Fleet request is deleted and has no running instances. The Spot Fleet will be deleted two days after its instances are terminated.

modify_in_progress

The Spot Fleet request is being modified. The request remains in this state until the modification is fully processed or the Spot Fleet request is deleted.

modify_succeeded

The Spot Fleet request was modified.

expired

The Spot Fleet request has expired. If the request was created with TerminateInstancesWithExpiration set, a subsequent event indicates that the instances are terminated.

EC2 Spot Fleet Spot Instance Request Change

Spot Fleet sends an EC2 Spot Fleet Spot Instance Request Change event to Amazon EventBridge when a Spot Instance request in the fleet changes state.

The following is example data for this event.

```
{
  "version": "0",
  "id": "cd141ef0-14af-d670-a71d-fe46e9971bd2",
  "detail-type": "EC2 Spot Fleet Spot Instance Request Change",
  "source": "aws.ec2spotfleet",
  "account": "123456789012",
  "time": "2020-11-09T08:53:21Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-a98d2133-941a-47dc-8b03-0f94c6852ad1"
  ],
  "detail": {
    "spot-instance-request-id": "sir-a2w9gc5h",
    "description": "SpotInstanceRequestId sir-a2w9gc5h, PreviousState: cancelled_running",
  }
}
```
The possible values for `sub-type` are:

- **submitted**
  - The request is submitted.

- **disabled**
  - You stopped the Spot Instance.

- **active**
  - The request is fulfilled and has an associated Spot Instance.

- **cancelled**
  - You cancelled the request, or the request expired.

### EC2 Spot Fleet Instance Change

Spot Fleet sends an EC2 Spot Fleet Instance Change event to Amazon EventBridge when an instance in the fleet changes state.

The following is example data for this event.

```json
{
   "version": "0",
   "id": "11591686-5bd7-bbbaa-eb40-d46529c2710f",
   "detail-type": "EC2 Spot Fleet Instance Change",
   "source": "aws.ec2spotfleet",
   "account": "123456789012",
   "time": "2020-11-09T07:25:02Z",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-c8a764a4-bedc-4b62-af9c-0095e6e3ba61"
   ],
   "detail": {
      "instance-id": "i-08b90df1e09c30c9b",
      "description": "{"\"instanceType\":\"r4.2xlarge\",\"image\":\"ami-032930428bf1abf99f775c02f349b92\",\"productDescription\":\"Linux/UNIX\",\"availabilityZone\":\"us-east-1a\"}",
      "sub-type": "launched"
   }
}
```

The possible values for `sub-type` are:

- **launched**
  - A new instance was launched.

- **terminated**
  - The instance was terminated.

- **termination_notified**
  - An instance termination notification was sent.
EC2 Spot Fleet Information

Spot Fleet sends an EC2 Spot Fleet Information event to Amazon EventBridge when there is an error during fulfillment. The information event does not block the fleet from attempting to fulfil its target capacity.

The following is example data for this event.

```
{
    "version": "0",
    "id": "73a60f37-3409-a66c-635c-7f66c5f5b669",
    "detail-type": "EC2 Spot Fleet Information",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-08T20:56:12Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-2531ea06-af18-4647-8757-7d69c94971b1"
    ],
    "detail": {
        "description": "r3.8xlarge, ami-032930428bf1abbff, Linux/UNIX, us-east-1a, Spot bid price is less than Spot market price $0.5291",
        "sub-type": "launchSpecUnusable"
    }
}
```

The possible values for sub-type are:

- **launchSpecUnusable**
  The price in a launch specification is not valid because it is below the Spot price or the Spot price is above the On-Demand price.

- **fleetProgressHalted**
  The price in every launch specification is not valid. A launch specification might become valid if the Spot price changes.

- **registerWithLoadBalancersFailed**
  An attempt to register instances with load balancers failed. For more information, see the description of the event.

- **launchSpecTemporarilyBlacklisted**
  The configuration is not valid and several attempts to launch instances have failed. For more information, see the description of the event.

EC2 Spot Fleet Error

Spot Fleet sends an EC2 Spot Fleet Error event to Amazon EventBridge when there is an error during fulfillment. The error event blocks the fleet from attempting to fulfil its target capacity.

The following is example data for this event.

```
{
    "version": "0",
    "id": "10adc4e7-675c-643e-125c-5bfa1b1ba5d2",
    "detail-type": "EC2 Spot Fleet Error",
    "source": "aws.ec2spotfleet",
    "account": "123456789012",
    "time": "2020-11-08T20:56:12Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:spot-fleet-request/sfr-2531ea06-af18-4647-8757-7d69c94971b1"
    ],
    "detail": {
        "description": "r3.8xlarge, ami-032930428bf1abbff, Linux/UNIX, us-east-1a, Spot bid price is less than Spot market price $0.5291",
        "sub-type": "launchSpecUnusable"
    }
}
```
Create Amazon EventBridge rules

When a notification of a state change is emitted for an EC2 Fleet or Spot Fleet, the event for the notification is sent to Amazon EventBridge in the form of a JSON file. You can write an EventBridge rule to automate what actions to take when the event pattern matches the rule.

You can write an EventBridge rule and automate what actions to take when the event pattern matches the rule.

Topics

- Create Amazon EventBridge rules to monitor EC2 Fleet events (p. 755)
- Create Amazon EventBridge rules to monitor Spot Fleet events (p. 758)

Create Amazon EventBridge rules to monitor EC2 Fleet events

When a notification of a state change is emitted for an EC2 Fleet, the event for the notification is sent to Amazon EventBridge in the form of a JSON file. You can write an EventBridge rule to automate what actions to take when an event pattern matches the rule. If EventBridge detects an event pattern that matches a pattern defined in a rule, EventBridge invokes the target (or targets) specified in the rule.

The following fields form the event pattern that is defined in the rule:

"source": "aws.ec2fleet"

Identifies that the event is from EC2 Fleet.

"detail-type": "EC2 Fleet State Change"

Identifies the event type.
"detail": { "sub-type": "submitted" }

Identifies the event sub-type.

For the list of EC2 Fleet events and example event data, see the section called "EC2 Fleet event types" (p. 746).

Examples
- Create an EventBridge rule to send a notification (p. 756)
- Create an EventBridge rule to trigger a Lambda function (p. 757)

Create an EventBridge rule to send a notification

The following example creates an EventBridge rule to send an email, text message, or mobile push notification every time that Amazon EC2 emits an EC2 Fleet state change notification. The signal in this example is emitted as an EC2 Fleet State Change event, which triggers the action defined by the rule. Before creating the EventBridge rule, you must create the Amazon SNS topic for the email, text message, or mobile push notification.

To create an EventBridge rule to send a notification when an EC2 Fleet state changes
1. Open the Amazon EventBridge console at https://console.aws.amazon.com/events/.
2. Choose Create rule.
3. Enter a Name for the rule, and, optionally, a description.
   A rule can't have the same name as another rule in the same Region and on the same event bus.
4. For Define pattern, choose Event pattern.
5. Under Event matching pattern, you can choose Pre-defined pattern by service or Custom pattern. The Custom pattern allows you to create a more detailed rule.
   a. If you choose Pre-defined pattern by service, do the following:
      i. For Service provider, choose AWS.
      ii. For Service name, choose EC2 Fleet.
      iii. For Event type, select the required event type. For this example, choose EC2 Fleet Instance Change.
   b. If you choose Custom pattern, do the following:
      • In the Event pattern box, add the following pattern to match the EC2 Fleet Instance Change event for this example, and then choose Save.

```json
{
    "source": ["aws.ec2fleet"],
    "detail-type": ["EC2 Fleet Instance Change"]
}
```
6. For Select event bus, choose AWS default event bus. When an AWS service in your account emits an event, it always goes to your account's default event bus.
7. Confirm that Enable the rule on the selected event bus is toggled on.
8. For Target, choose SNS topic to send an email, text message, or mobile push notification when the event occurs.
9. For Topic, choose an existing topic. You first need to create an Amazon SNS topic using the Amazon SNS console. For more information, see Using Amazon SNS for application-to-person (A2P) messaging in the Amazon Simple Notification Service Developer Guide.
10. For **Configure input**, choose the input for the email, text message, or mobile push notification.
11. Choose **Create**.

For more information, see Amazon EventBridge rules and Amazon EventBridge event patterns in the Amazon EventBridge User Guide

**Create an EventBridge rule to trigger a Lambda function**

The following example creates an EventBridge rule to trigger a Lambda function every time that Amazon EC2 emits an EC2 Fleet instance change notification for when an instance is launched. The signal in this example is emitted as an EC2 Fleet Instance Change event, sub-type launched, which triggers the action defined by the rule. Before creating the EventBridge rule, you must create the Lambda function.

**To create an EventBridge rule to trigger a Lambda function when an instance in an EC2 Fleet changes state**

1. Open the AWS Lambda console at https://console.aws.amazon.com/lambda/.
2. Choose **Create function**.
3. Enter a name for your function, configure the code, and then choose **Create function**.
   For more information about using Lambda, see Create a Lambda function with the console in the AWS Lambda Developer Guide.
4. Open the Amazon EventBridge console at https://console.aws.amazon.com/events/.
5. Choose **Create rule**.
6. Enter a **Name** for the rule, and, optionally, a description.
   A rule can’t have the same name as another rule in the same Region and on the same event bus.
7. For **Define pattern**, choose **Event pattern**.
8. Under **Event matching pattern**, you can choose Pre-defined pattern by service or Custom pattern. The Custom pattern allows you to create a more detailed rule.
   a. If you choose Pre-defined pattern by service, do the following:
      i. For Service provider, choose AWS.
      ii. For Service name, choose EC2 Fleet.
      iii. For Event type, select the required event type. For this example, choose EC2 Fleet Instance Change.
   b. If you choose Custom pattern, do the following:
      • In the **Event pattern** box, add the following pattern to match the EC2 Fleet Instance Change event and launched sub-type for this example, and then choose **Save**.

        ```json
        {
            "source": ["aws.ec2fleet"],
            "detail-type": ["EC2 Fleet Instance Change"],
            "detail": {
                "sub-type": ["launched"]
            }
        }
        ```

9. For **Target**, choose **Lambda function**, and for **Function**, choose the function that you created to respond when the event occurs.
10. Choose **Create**.
   In this example, the Lambda function will be triggered when the EC2 Fleet Instance Change event with the sub-type launched occurs.
Create Amazon EventBridge rules to monitor Spot Fleet events

When a notification of a state change is emitted for a Spot Fleet, the event for the notification is sent to Amazon EventBridge in the form of a JSON file. You can write an EventBridge rule to automate what actions to take when an event pattern matches the rule. If EventBridge detects an event pattern that matches a pattern defined in a rule, EventBridge invokes the target (or targets) specified in the rule.

The following fields form the event pattern that is defined in the rule:

- "source": "aws.ec2spotfleet"
  - Identifies that the event is from Spot Fleet.
- "detail-type": "EC2 Spot Fleet State Change"
  - Identifies the event type.
- "detail": { "sub-type": "submitted" }
  - Identifies the event sub-type.

For the list of Spot Fleet events and example event data, see the section called “Spot Fleet event types” (p. 751).

Examples

- Create an EventBridge rule to send a notification (p. 756)
- Create an EventBridge rule to trigger a Lambda function (p. 757)

Create an EventBridge rule to send a notification

The following example creates an EventBridge rule to send an email, text message, or mobile push notification every time that Amazon EC2 emits a Spot Fleet state change notification. The signal in this example is emitted as an EC2 Spot Fleet State Change event, which triggers the action defined by the rule. Before creating the EventBridge rule, you must create the Amazon SNS topic for the email, text message, or mobile push notification.

To create an EventBridge rule to send a notification when a Spot Fleet state changes

1. Open the Amazon EventBridge console at https://console.aws.amazon.com/events/.
2. Choose Create rule.
3. Enter a Name for the rule, and, optionally, a description.
   - A rule can't have the same name as another rule in the same Region and on the same event bus.
4. For Define pattern, choose Event pattern.
5. Under Event matching pattern, you can choose Pre-defined pattern by service or Custom pattern.
   - The Custom pattern allows you to create a more detailed rule.
   a. If you choose Pre-defined pattern by service, do the following:
      i. For Service provider, choose AWS.
      ii. For Service name, choose EC2 Spot Fleet.
      iii. For Event type, select the required event type. For this example, choose EC2 Spot Fleet Instance Change.
b. If you choose **Custom pattern**, do the following:

- In the **Event pattern** box, add the following pattern to match the **EC2 Spot Fleet Instance Change event** for this example, and then choose **Save**.

```json
{
    "source": ["aws.ec2spotfleet"],
    "detail-type": ["EC2 Spot Fleet Instance Change"]
}
```

6. For **Select event bus**, choose **AWS default event bus**. When an AWS service in your account emits an event, it always goes to your account's default event bus.

7. Confirm that **Enable the rule on the selected event bus** is toggled on.

8. For **Target**, choose **SNS topic** to send an email, text message, or mobile push notification when the event occurs.

9. For **Topic**, choose an existing topic. You first need to create an Amazon SNS topic using the Amazon SNS console. For more information, see **Using Amazon SNS for application-to-person (A2P) messaging** in the **Amazon Simple Notification Service Developer Guide**.

10. For **Configure input**, choose the input for the email, text message, or mobile push notification.

11. Choose **Create**.

For more information, see **Amazon EventBridge rules** and **Amazon EventBridge event patterns** in the **Amazon EventBridge User Guide**.

### Create an EventBridge rule to trigger a Lambda function

The following example creates an EventBridge rule to trigger a Lambda function every time that Amazon EC2 emits a Spot Fleet instance change notification for when an instance is launched. The signal in this example is emitted as an **EC2 Spot Fleet Instance Change event**, sub-type **launched**, which triggers the action defined by the rule. Before creating the EventBridge rule, you must create the Lambda function.

**To create an EventBridge rule to trigger a Lambda function when an instance in a Spot Fleet changes state**

2. Choose **Create function**.
3. Enter a name for your function, configure the code, and then choose **Create function**.

   For more information about using Lambda, see **Create a Lambda function with the console** in the **AWS Lambda Developer Guide**.

4. Open the Amazon EventBridge console at [https://console.aws.amazon.com/events/](https://console.aws.amazon.com/events/).
5. Choose **Create rule**.
6. Enter a **Name** for the rule, and, optionally, a description.

   A rule can't have the same name as another rule in the same Region and on the same event bus.

7. For **Define pattern**, choose **Event pattern**.
8. Under **Event matching pattern**, you can choose **Pre-defined pattern by service** or **Custom pattern**. The **Custom pattern** allows you to create a more detailed rule.

   a. If you choose **Pre-defined pattern by service**, do the following:

   i. For **Service provider**, choose **AWS**.

   ii. For **Service name**, choose **EC2 Spot Fleet**.
iii. For Event type, select the required event type. For this example, choose EC2 Spot Fleet Instance Change.

b. If you choose Custom pattern, do the following:

- In the Event pattern box, add the following pattern to match the EC2 Spot Fleet Instance Change event and launched sub-type for this example, and then choose Save.

```
{
    "source": ["aws.ec2spotfleet"],
    "detail-type": ["EC2 Spot Fleet Instance Change"],
    "detail": {
        "sub-type": ["launched"]
    }
}
```

9. For Target, choose Lambda function, and for Function, choose the function that you created to respond when the event occurs.

10. Choose Create.

In this example, the Lambda function will be triggered when the EC2 Fleet Instance Change event with the sub-type launched occurs.

For a tutorial on how to create a Lambda function and an EventBridge rule that runs the Lambda function, see Tutorial: Log the State of an Amazon EC2 Instance Using EventBridge in the AWS Lambda Developer Guide.

Tutorials for EC2 Fleet and Spot Fleet

The following tutorials take you through the common processes for creating EC2 Fleets and Spot Fleets.

Tutorials
- Tutorial: Use EC2 Fleet with instance weighting (p. 760)
- Tutorial: Use EC2 Fleet with On-Demand as the primary capacity (p. 763)
- Tutorial: Use Spot Fleet with instance weighting (p. 764)

Tutorial: Use EC2 Fleet with instance weighting

This tutorial uses a fictitious company called Example Corp to illustrate the process of requesting an EC2 Fleet using instance weighting.

Objective

Example Corp, a pharmaceutical company, wants to use the computational power of Amazon EC2 for screening chemical compounds that might be used to fight cancer.

Planning

Example Corp first reviews Spot Best Practices. Next, Example Corp determines the requirements for their EC2 Fleet.

Instance types

Example Corp has a compute- and memory-intensive application that performs best with at least 60 GB of memory and eight virtual CPUs (vCPUs). They want to maximize these resources for the application at
the lowest possible price. Example Corp decides that any of the following EC2 instance types would meet their needs:

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Memory (GiB)</th>
<th>vCPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>r3.2xlarge</td>
<td>61</td>
<td>8</td>
</tr>
<tr>
<td>r3.4xlarge</td>
<td>122</td>
<td>16</td>
</tr>
<tr>
<td>r3.8xlarge</td>
<td>244</td>
<td>32</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Instance type</th>
<th>On-Demand price</th>
<th>Instance weight</th>
<th>Price per unit hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>r3.2xLarge</td>
<td>$0.7</td>
<td>1</td>
<td>$0.7</td>
</tr>
<tr>
<td>r3.4xLarge</td>
<td>$1.4</td>
<td>2</td>
<td>$0.7</td>
</tr>
<tr>
<td>r3.8xLarge</td>
<td>$2.8</td>
<td>4</td>
<td>$0.7</td>
</tr>
</tbody>
</table>

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.

**Verify 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. 693).
Create a launch template

Next, Example Corp creates a launch template. The launch template ID is used in the following step. For more information, see Create a launch template (p. 399).

Create the EC2 Fleet

Example Corp creates a file, config.json, with the following configuration for its EC2 Fleet. In the following example, replace the resource identifiers with your own resource identifiers.

```json
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-07b3bc7625cdab851",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "r3.2xlarge",
                    "SubnetId": "subnet-482e4972",
                    "WeightedCapacity": 1
                },
                {
                    "InstanceType": "r3.4xlarge",
                    "SubnetId": "subnet-482e4972",
                    "WeightedCapacity": 2
                },
                {
                    "InstanceType": "r3.8xlarge",
                    "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 Create an EC2 Fleet (p. 701).

Fulfillment

The allocation strategy determines which Spot capacity pools your Spot Instances come from.

With the lowest-price strategy (which is the default strategy), the Spot Instances come from the pool with the lowest price per unit at the time of fulfillment. To provide 20 units of capacity, the EC2 Fleet launches either 20 r3.2xlarge instances (20 divided by 1), 10 r3.4xlarge instances (20 divided by 2), or 5 r3.8xlarge instances (20 divided by 4).
Tutorial: Use EC2 Fleet with On-Demand as the primary capacity

This tutorial uses a fictitious company called ABC Online to illustrate the process of requesting an EC2 Fleet with On-Demand as the primary capacity, and Spot capacity if available.

Objective

ABC Online, a restaurant delivery company, wants to be able to provision Amazon EC2 capacity across EC2 instance types and purchasing options to achieve their desired scale, performance, and cost.

Plan

ABC Online requires a fixed capacity to operate during peak periods, but would like to benefit from increased capacity at a lower price. ABC Online determines the following requirements for their EC2 Fleet:

- On-Demand Instance capacity – ABC Online requires 15 On-Demand Instances to ensure that they can accommodate traffic at peak periods.
- Spot Instance capacity – ABC Online would like to improve performance, but at a lower price, by provisioning 5 Spot Instances.

Verify 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. 693).

Create a launch template

Next, ABC Online creates a launch template. The launch template ID is used in the following step. For more information, see Create a launch template (p. 399).

Create the EC2 Fleet

ABC Online creates a file, config.json, with the following configuration for its EC2 Fleet. In the following example, replace the resource identifiers with your own resource identifiers.

```json
{
    "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 Create an EC2 Fleet (p. 701).

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.

Tutorial: Use Spot Fleet with instance weighting

This tutorial 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:

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Memory (GiB)</th>
<th>vCPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>r3.2xlarge</td>
<td>61</td>
<td>8</td>
</tr>
<tr>
<td>r3.4xlarge</td>
<td>122</td>
<td>16</td>
</tr>
<tr>
<td>r3.8xlarge</td>
<td>244</td>
<td>32</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Instance type</th>
<th>On-Demand price</th>
<th>Instance weight</th>
<th>Price per unit hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>r3.2xLarge</td>
<td>$0.7</td>
<td>1</td>
<td>$0.7</td>
</tr>
<tr>
<td>r3.4xLarge</td>
<td>$1.4</td>
<td>2</td>
<td>$0.7</td>
</tr>
<tr>
<td>r3.8xLarge</td>
<td>$2.8</td>
<td>4</td>
<td>$0.7</td>
</tr>
</tbody>
</table>

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.

Verify permissions

Before creating a Spot Fleet request, Example Corp verifies that it has an IAM role with the required permissions. For more information, see Spot Fleet permissions (p. 721).

Create the request

Example Corp creates a file, config.json, with the following configuration for its Spot Fleet request:

```json
{
    "SpotPrice": "0.70",
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [ 
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "SubnetId": "subnet-482e4972",
            "WeightedCapacity": 1
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.4xlarge",
            "SubnetId": "subnet-482e4972",
            "WeightedCapacity": 2
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.8xlarge",
            "SubnetId": "subnet-482e4972",
            "SpotPrice": "0.90",
            "WeightedCapacity": 4
        }
    ]
}
```
Example Corp creates the Spot Fleet request using the `request-spot-fleet` command.

```
aws ec2 request-spot-fleet --spot-fleet-request-config file:///config.json
```

For more information, see Spot Fleet requests (p. 718).

**Fulfillment**

The allocation strategy determines which Spot capacity 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.

**Example configurations for EC2 Fleet and Spot Fleet**

The following examples show launch configurations that you can use to create EC2 Fleets and Spot Fleets.

**Topics**

- EC2 Fleet example configurations (p. 766)
- Spot Fleet example configurations (p. 780)

**EC2 Fleet example configurations**

The following examples show launch configurations that you can use with the `create-fleet` command to create an EC2 Fleet. For more information about the `create-fleet` parameters, see the EC2 Fleet JSON configuration file reference (p. 698).

**Examples**

- Example 1: Launch Spot Instances as the default purchasing option (p. 767)
- Example 2: Launch On-Demand Instances as the default purchasing option (p. 767)
- Example 3: Launch On-Demand Instances as the primary capacity (p. 768)
- Example 4: Launch Spot Instances using the lowest-price allocation strategy (p. 768)
- Example 5: Launch On-Demand Instances using Capacity Reservations and the prioritized allocation strategy (p. 769)
- Example 6: Launch On-Demand Instances using Capacity Reservations and the prioritized allocation strategy when the total target capacity is more than the number of unused Capacity Reservations (p. 771)
Example 7: Launch On-Demand Instances using Capacity Reservations and the lowest-price allocation strategy (p. 773)

Example 8: Launch On-Demand Instances using Capacity Reservations and the lowest-price allocation strategy when the total target capacity is more than the number of unused Capacity Reservations (p. 775)

Example 9: Configure Capacity Rebalancing to launch replacement Spot Instances (p. 778)

Example 10: Launch Spot Instances in a capacity-optimized fleet (p. 779)

Example 11: Launch Spot Instances in a capacity-optimized fleet with priorities (p. 779)

Example 1: Launch Spot Instances as the default purchasing option

The following example specifies the minimum parameters required in an EC2 Fleet: a launch template, target capacity, and default purchasing option. The launch template is identified by its launch template ID and version number. The target capacity for the fleet is 2 instances, and the default purchasing option is spot, which results in the fleet launching 2 Spot Instances.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "spot"
    }
}
```

Example 2: Launch On-Demand Instances as the default purchasing option

The following example specifies the minimum parameters required in an EC2 Fleet: a launch template, target capacity, and default purchasing option. The launch template is identified by its launch template ID and version number. The target capacity for the fleet is 2 instances, and the default purchasing option is on-demand, which results in the fleet launching 2 On-Demand Instances.

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "on-demand"
    }
}
```
Example 3: Launch On-Demand Instances as the primary capacity

The following example specifies the total target capacity of 2 instances for the fleet, and a target capacity of 1 On-Demand Instance. The default purchasing option is spot. The fleet launches 1 On-Demand Instance as specified, but needs to launch one more instance to fulfill the total target capacity. The purchasing option for the difference is calculated as TotalTargetCapacity - OnDemandTargetCapacity = DefaultTargetCapacityType, which results in the fleet launching 1 Spot Instance.

```
{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      }
    },
    "TargetCapacitySpecification": {
      "TotalTargetCapacity": 2,
      "OnDemandTargetCapacity": 1,
      "DefaultTargetCapacityType": "spot"
    }
  }
}
```

Example 4: Launch Spot Instances using the lowest-price allocation strategy

If the allocation strategy for Spot Instances is not specified, the default allocation strategy, which is lowest-price, is used. The following example uses the lowest-price allocation strategy. The three launch specifications, which override the launch template, have different instance types but the same weighted capacity and subnet. The total target capacity is 2 instances and the default purchasing option is spot. The EC2 Fleet launches 2 Spot Instances using the instance type of the launch specification with the lowest price.

```
{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      }
    },
    "Overrides": [
      {
        "InstanceType": "c4.large",
        "WeightedCapacity": 1,
        "SubnetId": "subnet-a4f6e5c6d3"
      },
      {
        "InstanceType": "c3.large",
        "WeightedCapacity": 1,
        "SubnetId": "subnet-a4f6e5c6d3"
      },
      {
        "InstanceType": "c5.large",
        "WeightedCapacity": 1,
        "SubnetId": "subnet-a4f6e5c6d3"
      }
    ]
  }
}
```
Example 5: Launch On-Demand Instances using Capacity Reservations and the prioritized allocation strategy

You can configure a fleet to use On-Demand Capacity Reservations first when launching On-Demand Instances by setting the usage strategy for Capacity Reservations to use-capacity-reservations-first. And if multiple instance pools have unused Capacity Reservations, the chosen On-Demand allocation strategy is applied. In this example, the On-Demand allocation strategy is prioritized.

In this example, there are 15 available unused Capacity Reservations. This is more than the fleet's target On-Demand capacity of 12 On-Demand Instances.

The account has the following 15 unused Capacity Reservations in 3 different pools. The number of Capacity Reservations in each pool is indicated by AvailableInstanceCount.

```
{
   "CapacityReservationId": "cr-111",
   "InstanceType": "c4.large",
   "InstancePlatform": "Linux/UNIX",
   "AvailabilityZone": "us-east-1a",
   "AvailableInstanceCount": 5,
   "InstanceMatchCriteria": "open",
   "State": "active"
}

{
   "CapacityReservationId": "cr-222",
   "InstanceType": "c3.large",
   "InstancePlatform": "Linux/UNIX",
   "AvailabilityZone": "us-east-1a",
   "AvailableInstanceCount": 5,
   "InstanceMatchCriteria": "open",
   "State": "active"
}

{
   "CapacityReservationId": "cr-333",
   "InstanceType": "c5.large",
   "InstancePlatform": "Linux/UNIX",
   "AvailabilityZone": "us-east-1a",
   "AvailableInstanceCount": 5,
   "InstanceMatchCriteria": "open",
   "State": "active"
}
```

The following fleet configuration shows only the pertinent configurations for this example. The On-Demand allocation strategy is prioritized, and the usage strategy for Capacity Reservations is use-capacity-reservations-first. The total target capacity is 12, and the default target capacity type is on-demand.
Note
The fleet type must be instant. use-capacity-reservations-first is not supported for other fleet types.

```json
{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
            "LaunchTemplateId": "lt-1234567890abcdefg",
            "Version": "1"
        },
        "Overrides": [
            {
                "InstanceType": "c4.large",
                "AvailabilityZone": "us-east-1a",
                "WeightedCapacity": 1,
                "Priority": 1.0
            },
            {
                "InstanceType": "c3.large",
                "AvailabilityZone": "us-east-1a",
                "WeightedCapacity": 1,
                "Priority": 2.0
            },
            {
                "InstanceType": "c5.large",
                "AvailabilityZone": "us-east-1a",
                "WeightedCapacity": 1,
                "Priority": 3.0
            }
        ]
    },
    
    
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 12,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "prioritized"
    },
    "CapacityReservationOptions": {
        "UsageStrategy": "use-capacity-reservations-first"
    },
    "Type": "instant"
}
```

After you create the instant fleet using the preceding configuration, the following 12 instances are launched to meet the target capacity:

- 5 c4.large On-Demand Instances in us-east-1a – c4.large in us-east-1a is prioritized first, and there are 5 available unused c4.large Capacity Reservations
- 5 c3.large On-Demand Instances in us-east-1a – c3.large in us-east-1a is prioritized second, and there are 5 available unused c3.large Capacity Reservations
- 2 c5.large On-Demand Instances in us-east-1a – c5.large in us-east-1a is prioritized third, and there are 5 available unused c5.large Capacity Reservations of which only 2 are needed to meet the target capacity

After the fleet is launched, you can run `describe-capacity-reservations` to see how many unused Capacity Reservations are remaining. In this example, you should see the following response, which shows that all of the c4.large and c3.large Capacity Reservations were used, with 3 c5.large Capacity Reservations remaining unused.
Example 6: Launch On-Demand Instances using Capacity Reservations and the prioritized allocation strategy when the total target capacity is more than the number of unused Capacity Reservations

You can configure a fleet to use On-Demand Capacity Reservations first when launching On-Demand Instances by setting the usage strategy for Capacity Reservations to `use-capacity-reservations-first`. And if the number of unused Capacity Reservations is less than the On-Demand target capacity, the remaining On-Demand target capacity is launched according to the chosen On-Demand allocation strategy. In this example, the On-Demand allocation strategy is `prioritized`.

In this example, there are 15 available unused Capacity Reservations. This is less than the fleet's On-Demand target capacity of 16 On-Demand Instances.

The account has the following 15 unused Capacity Reservations in 3 different pools. The number of Capacity Reservations in each pool is indicated by `AvailableInstanceCount`.

```json
{
  "CapacityReservationId": "cr-111",
  "InstanceType": "c4.large",
  "InstancePlatform": "Linux/UNIX",
  "AvailabilityZone": "us-east-1a",
  "AvailableInstanceCount": 5,
  "InstanceMatchCriteria": "open",
  "State": "active"
}
{
  "CapacityReservationId": "cr-111",
  "InstanceType": "c3.large",
  "InstancePlatform": "Linux/UNIX",
  "AvailabilityZone": "us-east-1a",
  "AvailableInstanceCount": 5,
  "InstanceMatchCriteria": "open",
  "State": "active"
}
{
  "CapacityReservationId": "cr-111",
  "InstanceType": "c5.large",
  "InstancePlatform": "Linux/UNIX",
  "AvailableInstanceCount": 3
}
```
The following fleet configuration shows only the pertinent configurations for this example. The On-Demand allocation strategy is prioritized, and the usage strategy for Capacity Reservations is use-capacity-reservations-first. The total target capacity is 16, and the default target capacity type is on-demand.

**Note**
The fleet type must be instant. use-capacity-reservations-first is not supported for other fleet types.

```json
{
   "LaunchTemplateConfigs": [
   {
      "LaunchTemplateSpecification": {
         "LaunchTemplateId": "lt-0e8c754449b27161c",
         "Version": "1"
      }
   },
   "Overrides": [
      {
         "InstanceType": "c4.large",
         "AvailabilityZone": "us-east-1a",
         "WeightedCapacity": 1,
         "Priority": 1.0
      },
      {
         "InstanceType": "c3.large",
         "AvailabilityZone": "us-east-1a",
         "WeightedCapacity": 1,
         "Priority": 2.0
      },
      {
         "InstanceType": "c5.large",
         "AvailabilityZone": "us-east-1a",
         "WeightedCapacity": 1,
         "Priority": 3.0
      }
   ],
   "TargetCapacitySpecification": {
      "TotalTargetCapacity": 16,
      "DefaultTargetCapacityType": "on-demand"
   },
   "OnDemandOptions": {
      "AllocationStrategy": "prioritized"
   },
   "Type": "instant",
}
```

After you create the **instant** fleet using the preceding configuration, the following 16 instances are launched to meet the target capacity:

- 6 `c4.large` On-Demand Instances in `us-east-1a` - `c4.large` in `us-east-1a` is prioritized first, and there are 5 available unused `c4.large` Capacity Reservations. The Capacity Reservations are used first to launch
5 On-Demand Instances plus an additional On-Demand Instance is launched according to the On-Demand allocation strategy, which is prioritized in this example.

- 5 c3.large On-Demand Instances in us-east-1a – c3.large in us-east-1a is prioritized second, and there are 5 available unused c3.large Capacity Reservations
- 5 c5.large On-Demand Instances in us-east-1a – c5.large in us-east-1a is prioritized third, and there are 5 available unused c5.large Capacity Reservations

After the fleet is launched, you can run `describe-capacity-reservations` to see how many unused Capacity Reservations are remaining. In this example, you should see the following response, which shows that all of the Capacity Reservations in all of the pools were used.

```json
{
  "CapacityReservationId": "cr-111",
  "InstanceType": "c4.large",
  "AvailableInstanceCount": 0
}
{
  "CapacityReservationId": "cr-222",
  "InstanceType": "c3.large",
  "AvailableInstanceCount": 0
}
{
  "CapacityReservationId": "cr-333",
  "InstanceType": "c5.large",
  "AvailableInstanceCount": 0
}
```

**Example 7: Launch On-Demand Instances using Capacity Reservations and the lowest-price allocation strategy**

You can configure a fleet to use On-Demand Capacity Reservations first when launching On-Demand Instances by setting the usage strategy for Capacity Reservations to `use-capacity-reservations-first`. And if multiple instance pools have unused Capacity Reservations, the chosen On-Demand allocation strategy is applied. In this example, the On-Demand allocation strategy is `lowest-price`.

In this example, there are 15 available unused Capacity Reservations. This is more than the fleet's target On-Demand capacity of 12 On-Demand Instances.

The account has the following 15 unused Capacity Reservations in 3 different pools. The number of Capacity Reservations in each pool is indicated by `AvailableInstanceCount`.

```json
{
  "CapacityReservationId": "cr-111",
  "InstanceType": "m5.large",
  "InstancePlatform": "Linux/UNIX",
  "AvailabilityZone": "us-east-1a",
  "AvailableInstanceCount": 5,
  "InstanceMatchCriteria": "open",
  "State": "active"
}
{
  "CapacityReservationId": "cr-222",
  "InstanceType": "m4.xlarge",
  "InstancePlatform": "Linux/UNIX",
  "AvailabilityZone": "us-east-1a",
  "AvailableInstanceCount": 0
}
```
The following fleet configuration shows only the pertinent configurations for this example. The On-Demand allocation strategy is lowest-price, and the usage strategy for Capacity Reservations is use-capacity-reservations-first. The total target capacity is 12, and the default target capacity type is on-demand.

In this example, the On-Demand Instance price is:

- m5.large – $0.096 per hour
- m4.xlarge – $0.20 per hour
- m4.2xlarge – $0.40 per hour

**Note**
The fleet type must be instant. use-capacity-reservations-first is not supported for other fleet types.
After you create the instant fleet using the preceding configuration, the following 12 instances are launched to meet the target capacity:

- 5 m5.large On-Demand Instances in us-east-1a – m5.large in us-east-1a is the lowest price, and there are 5 available unused m5.large Capacity Reservations
- 5 m4.xlarge On-Demand Instances in us-east-1a – m4.xlarge in us-east-1a is the next lowest price, and there are 5 available unused m4.xlarge Capacity Reservations
- 2 m4.2xlarge On-Demand Instances in us-east-1a – m4.2xlarge in us-east-1a is the third lowest price, and there are 5 available unused m4.2xlarge Capacity Reservations of which only 2 are needed to meet the target capacity

After the fleet is launched, you can run describe-capacity-reservations to see how many unused Capacity Reservations are remaining. In this example, you should see the following response, which shows that all of the m5.large and m4.xlarge Capacity Reservations were used, with 3 m4.2xlarge Capacity Reservations remaining unused.

```
{
  "CapacityReservationId": "cr-111",
  "InstanceType": "m5.large",
  "AvailableInstanceCount": 0
}
{
  "CapacityReservationId": "cr-222",
  "InstanceType": "m4.xlarge",
  "AvailableInstanceCount": 0
}
{
  "CapacityReservationId": "cr-333",
  "InstanceType": "m4.2xlarge",
  "AvailableInstanceCount": 3
}
```

**Example 8: Launch On-Demand Instances using Capacity Reservations and the lowest-price allocation strategy when the total target capacity is more than the number of unused Capacity Reservations**

You can configure a fleet to use On-Demand Capacity Reservations first when launching On-Demand Instances by setting the usage strategy for Capacity Reservations to use-capacity-reservations-first. And if the number of unused Capacity Reservations is less than the On-Demand target capacity, the remaining On-Demand target capacity is launched according to the chosen On-Demand allocation strategy. In this example, the On-Demand allocation strategy is lowest-price.

In this example, there are 15 available unused Capacity Reservations. This is less than the fleet's On-Demand target capacity of 16 On-Demand Instances.
The account has the following 15 unused Capacity Reservations in 3 different pools. The number of Capacity Reservations in each pool is indicated by `AvailableInstanceCount`.

```json
{  "CapacityReservationId": "cr-111",  "InstanceType": "m5.large",  "InstancePlatform": "Linux/UNIX",  "AvailabilityZone": "us-east-1a",  "AvailableInstanceCount": 5,  "InstanceMatchCriteria": "open",  "State": "active" }

{  "CapacityReservationId": "cr-222",  "InstanceType": "m4.xlarge",  "InstancePlatform": "Linux/UNIX",  "AvailabilityZone": "us-east-1a",  "AvailableInstanceCount": 5,  "InstanceMatchCriteria": "open",  "State": "active" }

{  "CapacityReservationId": "cr-333",  "InstanceType": "m4.2xlarge",  "InstancePlatform": "Linux/UNIX",  "AvailabilityZone": "us-east-1a",  "AvailableInstanceCount": 5,  "InstanceMatchCriteria": "open",  "State": "active" }
```

The following fleet configuration shows only the pertinent configurations for this example. The On-Demand allocation strategy is `lowest-price`, and the usage strategy for Capacity Reservations is `use-capacity-reservations-first`. The total target capacity is 16, and the default target capacity type is `on-demand`.

In this example, the On-Demand Instance price is:

- m5.large – $0.096 per hour
- m4.xlarge – $0.20 per hour
- m4.2xlarge – $0.40 per hour

**Note**

The fleet type must be `instant`. `use-capacity-reservations-first` is not supported for other fleet types.

```json
{  "LaunchTemplateConfigs": [    {      "LaunchTemplateSpecification": {        "LaunchTemplateId": "lt-0e8c754449b27161c",        "Version": "1"      }    },    "Overrides": [      {        "InstanceType": "m5.large",        "AvailabilityZone": "us-east-1a",        "WeightedCapacity": 1
```
After you create the *instant* fleet using the preceding configuration, the following 16 instances are launched to meet the target capacity:

- 6 m5.large On-Demand Instances in us-east-1a – m5.large in us-east-1a is the lowest price, and there are 5 available unused m5.large Capacity Reservations. The Capacity Reservations are used first to launch 5 On-Demand Instances plus an additional On-Demand Instance is launched according to the On-Demand allocation strategy, which is *lowest-price* in this example.
- 5 m4.xlarge On-Demand Instances in us-east-1a – m4.xlarge in us-east-1a is the next lowest price, and there are 5 available unused m4.xlarge Capacity Reservations
- 5 m4.2xlarge On-Demand Instances in us-east-1a – m4.2xlarge in us-east-1a is the third lowest price, and there are 5 available unused m4.2xlarge Capacity Reservations

After the fleet is launched, you can run `describe-capacity-reservations` to see how many unused Capacity Reservations are remaining. In this example, you should see the following response, which shows that all of the Capacity Reservations in all of the pools were used.

```json
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}
{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "AvailableInstanceCount": 0
}
{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "AvailableInstanceCount": 0
}
```
Example 9: Configure Capacity Rebalancing to launch replacement Spot Instances

The following example configures the EC2 Fleet to launch a replacement Spot Instance when Amazon EC2 emits a rebalance recommendation for a Spot Instance in the fleet. To configure the automatic replacement of Spot Instances, for ReplacementStrategy, specify launch.

Note
When a replacement instance is launched, the instance marked for rebalance is not automatically terminated. You can terminate it, or you can leave it running. You are charged for both instances while they are running.

The effectiveness of the Capacity Rebalancing strategy depends on the number of Spot capacity pools specified in the EC2 Fleet request. We recommend that you configure the fleet with a diversified set of instance types and Availability Zones, and for AllocationStrategy, specify capacity-optimized. For more information about what you should consider when configuring an EC2 Fleet for Capacity Rebalancing, see Capacity Rebalancing (p. 688).

```json
{
    "ExcessCapacityTerminationPolicy": "termination",
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "LaunchTemplate",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "c3.large",
                    "WeightedCapacity": 1,
                    "Placement": {
                        "AvailabilityZone": "us-east-1a"
                    }
                },
                {
                    "InstanceType": "c4.large",
                    "WeightedCapacity": 1,
                    "Placement": {
                        "AvailabilityZone": "us-east-1a"
                    }
                },
                {
                    "InstanceType": "c5.large",
                    "WeightedCapacity": 1,
                    "Placement": {
                        "AvailabilityZone": "us-east-1a"
                    }
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 5,
        "DefaultTargetCapacityType": "spot"
    },
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
        "MaintenanceStrategies": {
            "CapacityRebalance": {
                "ReplacementStrategy": "launch"
            }
        }
    }
}
```
Example 10: Launch Spot Instances in a capacity-optimized fleet

The following example demonstrates how to configure an EC2 Fleet with a Spot allocation strategy that optimizes for capacity. To optimize for capacity, you must set `AllocationStrategy` to `capacity-optimized`.

In the following example, the three launch specifications specify three Spot capacity pools. The target capacity is 50 Spot Instances. The EC2 Fleet attempts to launch 50 Spot Instances into the Spot capacity pool with optimal capacity for the number of instances that are launching.

```json
{
    "SpotOptions": {
        "AllocationStrategy": "capacity-optimized",
    },
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateName": "my-launch-template",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "r4.2xlarge",
                    "Placement": {
                        "AvailabilityZone": "us-west-2a"
                    }
                },
                {
                    "InstanceType": "m4.2xlarge",
                    "Placement": {
                        "AvailabilityZone": "us-west-2b"
                    }
                },
                {
                    "InstanceType": "c5.2xlarge",
                    "Placement": {
                        "AvailabilityZone": "us-west-2b"
                    }
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 50,
        "DefaultTargetCapacityType": "spot"
    }
}
```

Example 11: Launch Spot Instances in a capacity-optimized fleet with priorities

The following example demonstrates how to configure an EC2 Fleet with a Spot allocation strategy that optimizes for capacity while using priority on a best-effort basis.

When using the `capacity-optimized-prioritized` allocation strategy, you can use the `Priority` parameter to specify the priorities of the Spot capacity pools, where the lower the number the higher priority. You can also set the same priority for several Spot capacity pools if you favor them equally. If you do not set a priority for a pool, the pool will be considered last in terms of priority.
To prioritize Spot capacity pools, you must set `AllocationStrategy` to `capacity-optimized-prioritized`. EC2 Fleet will optimize for capacity first, but will honor the priorities on a best-effort basis (for example, if honoring the priorities will not significantly affect EC2 Fleet's ability to provision optimal capacity). This is a good option for workloads where the possibility of disruption must be minimized and the preference for certain instance types matters.

In the following example, the three launch specifications specify three Spot capacity pools. Each pool is prioritized, where the lower the number the higher priority. The target capacity is 50 Spot Instances. The EC2 Fleet attempts to launch 50 Spot Instances into the Spot capacity pool with the highest priority on a best-effort basis, but optimizes for capacity first.

```json
{
  "SpotOptions": {
    "AllocationStrategy": "capacity-optimized-prioritized"
  },
  "LaunchTemplateConfigs": [
  {
    "LaunchTemplateSpecification": {
      "LaunchTemplateName": "my-launch-template",
      "Version": "1"
    },
    "Overrides": [
    {
      "InstanceType": "r4.2xlarge",
      "Priority": 1
      "Placement": {
        "AvailabilityZone": "us-west-2a"
      },
    },
    {
      "InstanceType": "m4.2xlarge",
      "Priority": 2
      "Placement": {
        "AvailabilityZone": "us-west-2b"
      },
    },
    {
      "InstanceType": "c5.2xlarge",
      "Priority": 3
      "Placement": {
        "AvailabilityZone": "us-west-2b"
      }
    }
    ]
  },
  "TargetCapacitySpecification": {
    "TotalTargetCapacity": 50,
    "DefaultTargetCapacityType": "spot"
  }
}
```

**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 Create a Spot Fleet request (p. 725).

**Note**

For Spot Fleet, you can't specify an network interface ID in a launch specification. Make sure you omit the `NetworkInterfaceID` parameter in your launch specification.

**Examples**
Example 1: Launch Spot Instances using the lowest-priced Availability Zone or subnet in the Region (p. 781)

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",
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
      }
    },
    {
      "ImageId": "ami-567890123",
      "KeyName": "my-other-key-pair",
      "SecurityGroups": [
        {
          "GroupId": "sg-567890123"
        }
      ],
      "InstanceType": "m3.medium",
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-other-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.

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.
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**

```json
{
    "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**

```json
{
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
```
Example 4. Override the price for the request

We recommended 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

```json
{
    "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. 712).

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": "c3.8xlarge",
      "Placement": {
        "AvailabilityZone": "us-west-2b"
      }
    }
  ]
}
"InstanceType": "r3.2xlarge",
"Placement": {
    "AvailabilityZone": "us-west-2b"
}
],
}

Subnet

{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "SubnetId": "subnet-1a2b3c4d"
        }
    ]
}

A best practice to increase the chance that a spot request can be fulfilled by EC2 capacity in the event of an outage in one of the Availability Zones is to diversify across zones. For this scenario, include each Availability Zone available to you in the launch specification. And, instead of using the same subnet each time, use three unique subnets (each mapping to a different zone).

Availability Zone

{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2a"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",

    786
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 \texttt{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 \texttt{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 \textbf{Spot Fleet instance weighting (p. 717)}.
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. 711).

```json
{
  "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
}
```
Example 8: Configure Capacity Rebalancing to launch replacement Spot Instances

The following example configures the Spot Fleet to launch a replacement Spot Instance when Amazon EC2 emits a rebalance recommendation for a Spot Instance in the fleet. To configure the automatic replacement of Spot Instances, for ReplacementStrategy, specify launch.

Note
When a replacement instance is launched, the instance marked for rebalance is not automatically terminated. You can terminate it, or you can leave it running. You are charged for both instances while they are running.

The effectiveness of the Capacity Rebalancing strategy depends on the number of Spot capacity pools specified in the Spot Fleet request. We recommend that you configure the fleet with a diversified set of instance types and Availability Zones, and for AllocationStrategy, specify capacityOptimized. For more information about what you should consider when configuring a Spot Fleet for Capacity Rebalancing, see Capacity Rebalancing (p. 714).
Example 9: Launch Spot Instances in a capacity-optimized fleet

The following example demonstrates how to configure a Spot Fleet with a Spot allocation strategy that optimizes for capacity. To optimize for capacity, you must set AllocationStrategy to capacityOptimized.

In the following example, the three launch specifications specify three Spot capacity pools. The target capacity is 50 Spot Instances. The Spot Fleet attempts to launch 50 Spot Instances into the Spot capacity pool with optimal capacity for the number of instances that are launching.

```
{
  "TargetCapacity": "50",
  "SpotFleetRequestConfig": {
    "AllocationStrategy": "capacityOptimized",
  },
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "1"
      },
      "Overrides": [
        {
          "InstanceType": "r4.2xlarge",
          "AvailabilityZone": "us-west-2a"
        },
        {
          "InstanceType": "m4.2xlarge",
          "AvailabilityZone": "us-west-2b"
        },
        {
          "InstanceType": "c5.2xlarge",
          "AvailabilityZone": "us-west-2b"
        }
      ]
    }
  ]
}
```

Example 10: Launch Spot Instances in a capacity-optimized fleet with priorities

The following example demonstrates how to configure a Spot Fleet with a Spot allocation strategy that optimizes for capacity while using priority on a best-effort basis.

```
{
  "TargetCapacity": "50",
  "SpotFleetRequestConfig": {
    "AllocationStrategy": "capacityOptimized",
  },
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "1"
      },
      "Overrides": [
        {
          "InstanceType": "r4.2xlarge",
          "AvailabilityZone": "us-west-2a"
        },
        {
          "InstanceType": "m4.2xlarge",
          "AvailabilityZone": "us-west-2b"
        },
        {
          "InstanceType": "c5.2xlarge",
          "AvailabilityZone": "us-west-2b"
        }
      ]
    }
  ]
}
```
When using the `capacityOptimizedPrioritized` allocation strategy, you can use the `Priority` parameter to specify the priorities of the Spot capacity pools, where the lower the number the higher priority. You can also set the same priority for several Spot capacity pools if you favor them equally. If you do not set a priority for a pool, the pool will be considered last in terms of priority.

To prioritize Spot capacity pools, you must set `AllocationStrategy` to `capacityOptimizedPrioritized`. Spot Fleet will optimize for capacity first, but will honor the priorities on a best-effort basis (for example, if honoring the priorities will not significantly affect Spot Fleet’s ability to provision optimal capacity). This is a good option for workloads where the possibility of disruption must be minimized and the preference for certain instance types matters.

In the following example, the three launch specifications specify three Spot capacity pools. Each pool is prioritized, where the lower the number the higher priority. The target capacity is 50 Spot Instances. The Spot Fleet attempts to launch 50 Spot Instances into the Spot capacity pool with the highest priority on a best-effort basis, but optimizes for capacity first.

```json
{
  "TargetCapacity": "50",
  "SpotFleetRequestConfig": {
    "AllocationStrategy": "capacityOptimizedPrioritized"
  },
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "1"
      },
      "Overrides": [
        {
          "InstanceType": "r4.2xlarge",
          "Priority": 1,
          "AvailabilityZone": "us-west-2a"
        },
        {
          "InstanceType": "m4.2xlarge",
          "Priority": 2,
          "AvailabilityZone": "us-west-2b"
        },
        {
          "InstanceType": "c5.2xlarge",
          "Priority": 3,
          "AvailabilityZone": "us-west-2b"
        }
      ]
    }
  ]
}
```

Fleet quotas

The usual Amazon EC2 quotas apply to instances launched by an EC2 Fleet or a Spot Fleet, such as Spot Instance limits (p. 327) and volume limits (p. 1407). In addition, the following limits apply:

- The number of active EC2 Fleets and Spot Fleets per Region: 1,000* †
- The number of Spot capacity pools (unique combination of instance type and subnet): 300* ‡
- The size of the user data in a launch specification: 16 KB †
- The target capacity per EC2 Fleet or Spot Fleet: 10,000
- The target capacity across all EC2 Fleets and Spot Fleets in a Region: 100,000*
• An EC2 Fleet request or a Spot Fleet request can't span Regions.
• An EC2 Fleet request or a Spot Fleet request can't span different subnets from the same Availability Zone.

If you need more than the default limits for target capacity, complete the AWS Support Center Create case form to request a limit increase. For Limit type, choose EC2 Fleet, choose a Region, and then choose Target Fleet Capacity per Fleet (in units) or Target Fleet Capacity per Region (in units), or both.

* These limits apply to both your EC2 Fleets and your Spot Fleets.
† These are hard limits. You cannot request a limit increase for these limits.
‡ This limit only applies to fleets of type request or maintain. This limit does not apply to instant fleets.
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. 216).

Contents
• Elastic Graphics basics (p. 793)
• Pricing for Elastic Graphics (p. 795)
• Elastic Graphics limitations (p. 795)
• Work with Elastic Graphics (p. 796)
• Use CloudWatch metrics to monitor Elastic Graphics (p. 801)
• Troubleshoot (p. 803)

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.

Note
Bare metal instances are not supported.

Elastic Graphics accelerators are available in the following AWS Regions: us-east-1, us-east-2, us-west-2, ap-northeast-1, ap-southeast-1, ap-southeast-2, eu-central-1, and eu-west-1.

The following instance types support Elastic Graphics accelerators:
• C3 | C4 | C5 | C5a | C5ad | C5d | C5n
• D2 | D3 | D3en
• H1
• I3 | I3en
• M3 | M4 | M5 | M5d | M5dn | M5n
• P2 | P3 | P3dn
• R3 | R4 | R5 | R5d | R5dn | R5n
• t2.medium or larger | t3.medium or larger
• X1 | X1e
• z1d
The following Elastic Graphics accelerators are available. You can attach any Elastic Graphics accelerator to any supported instance type.

<table>
<thead>
<tr>
<th>Elastic Graphics accelerator</th>
<th>Graphics memory (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>eg1.medium</td>
<td>1</td>
</tr>
<tr>
<td>eg1.large</td>
<td>2</td>
</tr>
<tr>
<td>eg1.xlarge</td>
<td>4</td>
</tr>
<tr>
<td>eg1.2xlarge</td>
<td>8</td>
</tr>
</tbody>
</table>

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 or restart 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 Configure your security groups (p. 796).

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 Instance or Spot Instance, however, the On-Demand price for the accelerator applies.

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 with Microsoft Windows Server 2012 R2 or later. Linux instances are currently not supported.
- You can attach one accelerator to an instance at a time.
- You can attach an accelerator only during instance launch. You cannot attach an accelerator to an existing instance.
- You can't hibernate an instance with an attached accelerator.
- 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.
- You can't attach accelerators to instances that are configured to use Instance Metadata Service v2 (IMDSv2).
Work 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. For limitations, see Elastic Graphics limitations (p. 795).

Tasks
- Configure your security groups (p. 796)
- Launch an instance with an Elastic Graphics accelerator (p. 797)
- Install the required software for Elastic Graphics (p. 798)
- Verify Elastic Graphics functionality on your instance (p. 798)
- View Elastic Graphics information (p. 800)
- Submit feedback (p. 801)

Configure your security groups

Elastic Graphics requires a self-referencing security group that allows inbound and outbound traffic to and from the security group itself. The security group must include the following inbound and outbound rules:

<table>
<thead>
<tr>
<th>Inbound rule</th>
<th>Type</th>
<th>Protocol</th>
<th>Port</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Elastic Graphics</td>
<td>TCP</td>
<td>2007</td>
<td>The security group ID (its own resource ID)</td>
</tr>
</tbody>
</table>

If you use the Amazon EC2 console to launch your instance with an Elastic Graphics accelerator, you can either allow the launch instance wizard to automatically create the required security group rules, or you can select a security that you created previously.

If you are launching your instance using the AWS CLI or an SDK, you must specify a security group that you created previously.

To create 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 and then choose Create security group.
3. In the Create security group window, do the following:
   a. For Security group name, enter a descriptive name for the security group, such as Elastic Graphics security group.
   b. (Optional) For Description, enter a brief description of the security group.
   c. For VPC, select the VPC into which you intend to use Elastic Graphics.
Launch 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. 795).

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. 793).
4. On the Configure Instance Details page, select a VPC and subnet in which to launch your instance.
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 Configure your security groups (p. 796). 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:
Amazon Elastic Compute Cloud
User Guide for Windows Instances
Install the required software for Elastic Graphics

```
--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 Configure your security groups (p. 796).

To associate an Elastic Graphics accelerator during instance launch (Tools for Windows PowerShell)

Use the `New-EC2Instance` Tools for Windows PowerShell command.

Install 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.

Verify 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 Install the required software for Elastic Graphics (p. 798).

Contents

- Use the Elastic Graphics status monitor (p. 798)
- Use the Elastic Graphics command line tool (p. 799)

Use 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.
Use 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.

View 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 Details 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:

```bash
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.

```bash
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:
     ```bash
     ```
   - From your web browser, paste the following URL into the address field:
Submit 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

1. In the notification area of the taskbar in your Windows instance, open the Elastic Graphics Status Monitor.
2. In the lower left corner, choose Feedback.
3. Enter your feedback and choose Submit.

Use 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.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPUConnectivityCheckFailed</td>
<td>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</td>
</tr>
<tr>
<td>GPUHealthCheckFailed</td>
<td>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</td>
</tr>
<tr>
<td>GPUMemoryUtilization</td>
<td>The GPU memory used.</td>
</tr>
<tr>
<td></td>
<td>Units: MiB</td>
</tr>
</tbody>
</table>
Elastic Graphics dimensions

You can filter the metrics data for your Elastic Graphics accelerators using the following dimensions.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGPUId</td>
<td>Filters the data by the Elastic Graphics accelerator.</td>
</tr>
<tr>
<td>InstanceId</td>
<td>Filters the data by the instance to which the Elastic Graphics accelerator is attached.</td>
</tr>
</tbody>
</table>

View 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**

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.

**To view Elastic Graphics metrics (AWS CLI)**

Use the following list-metrics command:

```
aws cloudwatch list-metrics --namespace "AWS/ElasticGPUs"
```

Create 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**

2. In the navigation pane, choose Alarms, Create Alarm.
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**.

---

**Troubleshoot**

The following are common errors and troubleshooting steps.

**Contents**

- Investigate application performance issues (p. 803)
  - OpenGL rendering performance issues (p. 803)
  - Remote access performance issues (p. 804)
- Resolve unhealthy status issues (p. 805)
  - Stop and start the instance (p. 805)
  - Verify the installed components (p. 805)
  - Check the Elastic Graphics logs (p. 805)

**Investigate 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\ElasticGPUs\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:

```
```
  - 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 instance’s 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.
Resolve 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 Stop and start your instances (p. 427).

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 Install the required software for Elastic Graphics (p. 798).

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
Monitor 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 collect. 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:

<table>
<thead>
<tr>
<th>Item to monitor</th>
<th>Amazon EC2 metric</th>
<th>Monitoring agent/CloudWatch Logs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU utilization</td>
<td>CPUUtilization (p. 842)</td>
<td></td>
</tr>
<tr>
<td>Network utilization</td>
<td>NetworkIn (p. 842)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NetworkOut (p. 842)</td>
<td></td>
</tr>
<tr>
<td>Disk performance</td>
<td>DiskReadOps (p. 842)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DiskWriteOps (p. 842)</td>
<td></td>
</tr>
<tr>
<td>Disk Reads/Writes</td>
<td>DiskReadBytes (p. 842)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DiskWriteBytes (p. 842)</td>
<td></td>
</tr>
<tr>
<td>Memory utilization, disk swap utilization, disk space utilization, page file utilization, log collection</td>
<td>[Linux and Windows Server instances] Collect Metrics and Logs from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent</td>
<td></td>
</tr>
</tbody>
</table>
Automated and manual monitoring

AWS provides various tools that you can use to monitor Amazon EC2. You can configure some of these tools to do the monitoring for you, while some of the tools require manual intervention.

Monitoring tools

- Automated monitoring tools (p. 807)
- Manual monitoring tools (p. 808)

## Automated monitoring tools

You can use the following automated monitoring tools to watch Amazon EC2 and report back to you when something is wrong:

- **System status checks** – monitor the AWS systems required to use your instance to ensure that they are working properly. These checks detect problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue or you can resolve it yourself (for example, by stopping and restarting or terminating and replacing an instance). Examples of problems that cause system status checks to fail include:
  - Loss of network connectivity
  - Loss of system power
  - Software issues on the physical host
  - Hardware issues on the physical host that impact network reachability

For more information, see Status checks for your instances (p. 809).

- **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. 809).

- **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 Monitor your instances using CloudWatch (p. 839).

- **Amazon EventBridge** – automate your AWS services and respond automatically to system events. Events from AWS services are delivered to EventBridge 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 EventBridge?.

- **Amazon CloudWatch Logs** – monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, or other sources. For more information, see the Amazon CloudWatch Logs User Guide.

- **CloudWatch agent** – collect logs and system-level metrics from both hosts and guests on your EC2 instances and on-premises servers. For more information, see Collecting Metrics and Logs from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent in the Amazon CloudWatch User Guide.

- **AWS Management Pack for Microsoft System Center Operations Manager** – links Amazon EC2 instances and the Windows or Linux operating systems running inside them. The AWS Management Pack is an extension to Microsoft System Center Operations Manager. It uses a designated computer in your datacenter (called a watcher node) and the Amazon Web Services APIs to remotely discover and collect information about your AWS resources. For more information, see AWS Management Pack for Microsoft System Center (p. 1551).

**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.
Monitor the status of your instances

You can monitor the status of your instances by viewing status checks and scheduled events for your instances.

A status check gives you the information that results from automated checks performed by Amazon EC2. These automated checks detect whether specific issues are affecting your instances. The status check information, together with the data provided by Amazon CloudWatch, gives you detailed operational visibility into each of your instances.

You can also see status of specific events that are scheduled for your instances. The status of events provides information about upcoming activities that are planned for your instances, such as rebooting or retirement. They also provide the scheduled start and end time of each event.

Contents

• Status checks for your instances (p. 809)
• Scheduled events for your instances (p. 816)

Status checks for your instances

With instance status monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues. You can view the results of these status checks to identify specific and detectable problems. The event status data augments the information that Amazon EC2 already provides about the state of each instance (such as pending, running, stopping) and the utilization metrics that Amazon CloudWatch monitors (CPU utilization, network traffic, and disk activity).

Status checks are performed every minute, returning a pass or a fail status. If all checks pass, the overall status of the instance is OK. If one or more checks fail, the overall status is impaired. Status checks are built into Amazon EC2, so they cannot be disabled or deleted.

When a status check fails, the corresponding CloudWatch metric for status checks is incremented. For more information, see Status check metrics (p. 848). You can use these metrics to create CloudWatch alarms that are triggered based on the result of the status checks. For example, you can create an alarm to warn you if status checks fail on a specific instance. For more information, see Create and edit status check alarms (p. 813).
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. 447).

Contents

- Types of status checks (p. 810)
- View status checks (p. 811)
- Report instance status (p. 813)
- Create and edit status check alarms (p. 813)

Types of status checks

There are two types of status checks: system status checks and instance status checks.

System status checks

System status checks monitor the AWS systems on which your instance runs. These checks detect underlying problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue, or you can resolve it yourself. For instances backed by Amazon EBS, you can stop and start the instance yourself, which in most cases results in the instance being migrated to a new host. For Linux instances backed by instance store, you can terminate and replace the instance. For Windows instances, the root volume must be an Amazon EBS volume; instance store is not supported for the root volume. Note that instance store volumes are ephemeral and all data is lost when the instance is stopped.

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

Note

If you perform a restart from the operating system on a bare metal instance, the system status check might temporarily return a fail status. When the instance becomes available, the system status check should return a pass status.

Instance status checks

Instance status checks monitor the software and network configuration of your individual instance. Amazon EC2 checks the health of the instance by sending an address resolution protocol (ARP) request to the network interface (NIC). These checks detect problems that require your involvement to repair. When an instance status check fails, you typically must address the problem yourself (for example, by rebooting the instance or by making instance configuration changes).

The following are examples of problems that can cause instance status checks to fail:

- Failed system status checks
- Incorrect networking or startup configuration
- Exhausted memory
- Corrupted file system
- During instance reboot or while a Windows instance store-backed instance is being bundled, an instance status check reports a failure until the instance becomes available again.
Note
If you perform a restart from the operating system on a bare metal instance, the instance status check might temporarily return a fail status. When the instance becomes available, the instance status check should return a pass status.

View status checks
Amazon EC2 provides you with several ways to view and work with status checks.

View status using the console
You can view status checks by using the AWS Management Console.

New console

To view status checks (console)
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. On the Instances page, the Status check column lists the operational status of each instance.
4. To view the status of a specific instance, select the instance, and then choose the Status checks tab.

If your instance has a failed status check, you typically must address the problem yourself (for example, by rebooting the instance or by making instance configuration changes). However, if your instance has a failed status check and has been unreachable for over 20 minutes, choose Open support case to submit a request for assistance.

5. To review the CloudWatch metrics for status checks, select the instance, and then choose the Monitoring tab. Scroll until you see the graphs for the following metrics:
   - Status check failed (any)
   - Status check failed (instance)
Old console

To view status checks (console)
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. On the Instances page, the Status Checks column lists the operational status of each instance.
4. To view the status of a specific instance, select the instance, and then choose the Status Checks tab.

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.
5. To review the CloudWatch metrics for status checks, select the instance, and then choose the Monitoring tab. Scroll until you see the graphs for the following metrics:
   - Status Check Failed (Any)
   - Status Check Failed (Instance)
   - Status Check Failed (System)

View status using the command line

You can view status checks for running instances by using the describe-instance-status (AWS CLI) command.

To view the status of all instances, use the following command.

```bash
aws ec2 describe-instance-status
```

To get the status of all instances with an instance status of impaired, use the following command.

```bash
aws ec2 describe-instance-status \   --filters Name=instance-status.status,Values=impaired
```

To get the status of a single instance, use the following command.

```bash
aws ec2 describe-instance-status \   --instance-ids i-1234567890abcdef0
```

Alternatively, use the following commands:
Report instance status

You can provide feedback if you are having problems with an instance whose status is not shown as impaired, or if you want to send AWS additional details about the problems you are experiencing with an impaired instance.

We use reported feedback to identify issues impacting multiple customers, but do not respond to individual account issues. Providing feedback does not change the status check results that you currently see for the instance.

Report status feedback using the console

New console

To report instance status (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance, choose the Status Checks tab, choose Actions (the second Actions menu in the bottom half of the page), and then choose Report instance status.
4. Complete the Report instance status form, and then choose Submit.

Old console

To report instance status (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance, choose the Status Checks tab, and choose Submit feedback.
4. Complete the Report Instance Status form, and then choose Submit.

Report status feedback using the command line

Use the report-instance-status (AWS CLI) command to send feedback about the status of an impaired instance.

```
aws ec2 report-instance-status \
    --instances i-1234567890abcdef0 \
    --status impaired \
    --reason-codes code
```

Alternatively, use the following commands:

- Send-EC2InstanceStatus (AWS Tools for Windows PowerShell)
- ReportInstanceStatus (Amazon EC2 Query API)

Create and edit status check alarms

You can use the status check metrics (p. 848) to create CloudWatch alarms to notify you when an instance has a failed status check.
Create a status check alarm using the console

Use the following procedure to configure an alarm that sends you a notification by email, or stops, terminates, or recovers an instance when it fails a status check.

New console

To create a status check alarm (console)
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance, choose the Status Checks tab, and choose Actions, Create status check alarm.
4. On the Manage CloudWatch alarms page, under Add or edit alarm, choose Create an alarm.
5. For Alarm notification, turn the toggle on to configure Amazon Simple Notification Service (Amazon SNS) notifications. Select an existing Amazon SNS topic or enter a name to create a new topic.
   
   If you add 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.
6. For Alarm action, turn the toggle on to specify an action to take when the alarm is triggered. Select the action.
7. For Alarm thresholds, specify the metric and criteria for the alarm.
   
   You can leave the default settings for Group samples by (Average) and Type of data to sample (Status check failed:either), or you can change them to suit your needs.
   
   For Consecutive period, set the number of periods to evaluate and, in Period, enter the evaluation period duration before triggering the alarm and sending an email.
8. (Optional) For Sample metric data, choose Add to dashboard.
9. Choose Create.

Old console

To create a status check alarm (console)
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance, choose the Status Checks tab, and choose Create Status Check Alarm.
4. Select Send a notification to. Choose an existing SNS topic, or choose create topic to create a new one. If creating a new topic, in With these recipients, enter your email address and the addresses of any additional recipients, separated by commas.
5. (Optional) Select Take the action, and then select the action that you’d like to take.
6. In Whenever, select the status check that you want to be notified about.
   
   If you selected Recover this instance in the previous step, select Status Check Failed (System).
7. In For at least, set the number of periods you want to evaluate and in consecutive periods, select the evaluation period duration before triggering the alarm and sending an email.
8. (Optional) In Name of alarm, replace the default name with another name for the alarm.
9. Choose Create Alarm.
Important
If you added an email address to the list of recipients or created a new topic, Amazon
SNS sends a subscription confirmation email message to each new address. Each
recipient must confirm the subscription by choosing the link contained in that message.
Alert notifications are sent only to confirmed addresses.

If you need to make changes to an instance status alarm, you can edit it.

New console

To edit a status check alarm using the console
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance and choose Actions, Monitoring, Manage CloudWatch alarms.
4. On the Manage CloudWatch alarms page, under Add or edit alarm, choose Edit an alarm.
5. For Search for alarm, choose the alarm.
6. When you are finished making changes, choose Update.

Old console

To edit a status check alarm using the console
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance and choose Actions, CloudWatch Monitoring, Add/Edit Alarms.
4. In the Alarm Details dialog box, choose the name of the alarm.
5. In the Edit Alarm dialog box, make the desired changes, and then choose Save.

Create a status check alarm using the AWS CLI

In the following example, the alarm publishes a notification to an SNS topic, arn:aws:sns:us-
west-2:111122223333:my-sns-topic, when the instance fails either the instance check or system
status check for at least two consecutive periods. The CloudWatch metric used is StatusCheckFailed.

To create a status check alarm using the AWS CLI
1. Select an existing SNS topic or create a new one. For more information, see Using the AWS CLI with
Amazon SNS in the AWS Command Line Interface User Guide.
2. Use the following list-metrics command to view the available Amazon CloudWatch metrics for
Amazon EC2.
3. Use the following put-metric-alarm command to create the alarm.

```bash
aws cloudwatch list-metrics --namespace AWS/EC2

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
```
Scheduled events for your instances

AWS can schedule events for your instances, such as a reboot, stop/start, or retirement. These events do not occur frequently. If one of your instances will be affected by a scheduled event, AWS sends an email to the email address that's associated with your AWS account prior to the scheduled event. The email provides details about the event, including the start and end date. Depending on the event, you might be able to take action to control the timing of the event. AWS also sends an AWS Health event, which you can monitor and manage by using Amazon CloudWatch Events. For more information about monitoring AWS Health events with CloudWatch, see Monitoring AWS Health events with CloudWatch Events.

Scheduled events are managed by AWS; you cannot schedule events for your instances. You can view the events scheduled by AWS, customize scheduled event notifications to include or remove tags from the email notification, perform actions when an instance is scheduled to reboot, retire, or stop.

To update the contact information for your account so that you can be sure to be notified about scheduled events, go to the Account Settings page.

Contents
- Types of scheduled events (p. 816)
- View scheduled events (p. 816)
- Customize scheduled event notifications (p. 821)
- Work with instances scheduled to stop or retire (p. 823)
- Work with instances scheduled for reboot (p. 823)
- Work with instances scheduled for maintenance (p. 825)
- Reschedule a scheduled event (p. 825)
- Define event windows for scheduled events (p. 827)

Types of scheduled events

Amazon EC2 can create the following types of events for your instances, where the event occurs at a scheduled time:

- **Instance stop**: At the scheduled time, the instance is stopped. When you start it again, it's migrated to a new host. Applies only to instances backed by Amazon EBS.
- **Instance retirement**: At the scheduled time, the instance is stopped if it is backed by Amazon EBS, or terminated if it is backed by instance store.
- **Instance reboot**: At the scheduled time, the instance is rebooted.
- **System reboot**: At the scheduled time, the host for the instance is rebooted.
- **System maintenance**: At the scheduled time, the instance might be temporarily affected by network maintenance or power maintenance.

View scheduled events

In addition to receiving notification of scheduled events in email, you can check for scheduled events by using one of the following methods.
New console

**To view scheduled events for your instances using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. You can view scheduled events in the following screens:
   - In the navigation pane, choose Events. Any resources with an associated event are displayed. You can filter by Resource ID, Resource type, Availability zone, Event status, or Event type.

   ![Events Screen]

   - Alternatively, in the navigation pane, choose EC2 Dashboard. Any resources with an associated event are displayed under Scheduled events.

Old console

**To view scheduled events for your instances using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. You can view scheduled events in the following screens:
   - In the navigation pane, choose Events. Any resources with an associated event are displayed. You can filter by resource type, or by specific event types. You can select the resource to view details.
Scheduled events

- Alternatively, in the navigation pane, choose **EC2 Dashboard.** Any resources with an associated event are displayed under **Scheduled Events.**

- Some events are also shown for affected resources. For example, in the navigation pane, choose **Instances** and select an instance. If the instance has an associated instance stop or instance retirement event, it is displayed in the lower pane.

**AWS CLI**

To view scheduled events for your instances using the AWS CLI

Use the **describe-instance-status** command.

```
aws ec2 describe-instance-status \
   --instance-id i-1234567890abcdef0 \ 
   --query "InstanceStatuses[].Events"
```

The following example output shows a reboot event.

```json
[
   "Events": [
      {
         "InstanceEventId": "instance-event-0d59937288b749b32",
         "Code": "system-reboot",
         "Description": "The instance is scheduled for a reboot",
         "NotAfter": "2019-03-15T22:00:00.000Z",
         "NotBefore": "2019-03-14T20:00:00.000Z",
         "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"
      }
   ]
```
The following example output shows an instance retirement event.

```

"Events": [
  {
    "InstanceEventId": "instance-event-0e439355b779n26",
    "Code": "instance-stop",
    "Description": "The instance is running on degraded hardware",
    "NotBefore": "2015-05-23T00:00:00.000Z"
  }
]
```

PowerShell

**To view scheduled events for your instances using the AWS Tools for Windows PowerShell**

Use the following `Get-EC2InstanceStatus` command.

```
PS C:\> (Get-EC2InstanceStatus -InstanceId i-1234567890abcdef0).Events
```

The following example output shows an instance retirement event.

```
Code         : instance-stop
Description  : The instance is running on degraded hardware
NotBefore    : 5/23/2015 12:00:00 AM
```

Instance metadata

**To view scheduled events for your instances using instance metadata**

You can retrieve information about active maintenance events for your instances from the instance metadata (p. 579) by using Instance Metadata Service Version 2 or Instance Metadata Service Version 1.

**IMDSv2**

```
```

**IMDSv1**

```
```

The following is example output with information about a scheduled system reboot event, in JSON format.

```
[
```

819
To view event history about completed or canceled events for your instances using instance metadata

You can retrieve information about completed or canceled events for your instances from instance metadata (p. 579) by using Instance Metadata Service Version 2 or Instance Metadata Service Version 1.

**IMDSv2**

```
```

**IMDSv1**

```
```

The following is example output with information about a system reboot event that was canceled, and a system reboot event that was completed, in JSON format.

```
[
  {
    "NotBefore": "21 Jan 2019 09:00:43 GMT",
    "Code": "system-reboot",
    "Description": "[Canceled] scheduled reboot",
    "EventId": "instance-event-0d59937288b749b32",
    "NotAfter": "21 Jan 2019 09:17:23 GMT",
    "State": "canceled"
  },
  {
    "NotBefore": "29 Jan 2019 09:00:43 GMT",
    "Code": "system-reboot",
    "Description": "[Completed] scheduled reboot",
    "EventId": "instance-event-0d59937288b749b32",
    "NotAfter": "29 Jan 2019 09:17:23 GMT",
    "State": "completed"
  }
]
```

**AWS Health**

You can use the AWS Personal Health Dashboard to learn about events that can affect your instance. The AWS Personal Health Dashboard organizes issues in three groups: open issues, scheduled changes, and other notifications. The scheduled changes group contains items that are ongoing or upcoming.

For more information, see Getting started with the AWS Personal Health Dashboard in the AWS Health User Guide.
Customize scheduled event notifications

You can customize scheduled event notifications to include tags in the email notification. This makes it easier to identify the affected resource (instances or Dedicated Hosts) and to prioritize actions for the upcoming event.

When you customize event notifications to include tags, you can choose to include:

- All of the tags that are associated with the affected resource
- Only specific tags that are associated with the affected resource

For example, suppose that you assign application, costcenter, project, and owner tags to all of your instances. You can choose to include all of the tags in event notifications. Alternatively, if you’d like to see only the owner and project tags in event notifications, then you can choose to include only those tags.

After you select the tags to include, the event notifications will include the resource ID (instance ID or Dedicated Host ID) and the tag key and value pairs that are associated with the affected resource.

Topics

- Include tags in event notifications (p. 821)
- Remove tags from event notifications (p. 822)
- View the tags to be included in event notifications (p. 822)

Include tags in event notifications

The tags that you choose to include apply to all resources (instances and Dedicated Hosts) in the selected Region. To customize event notifications in other Regions, first select the required Region and then perform the following steps.

You can include tags in event notifications by using one of the following methods.

New console

To include tags in event notifications

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Events.
3. Choose Actions, Manage event notifications.
4. Select Include resource tags in event notifications.
5. Do one of the following, depending on the tags that you want to include in event notifications:
   - To include all of the tags associated with the affected instance or Dedicated Host, select Include all resource tags.
   - To manually select the tags to include, select Choose the tags to include, and then for Choose the tags to include, enter the tag key and press Enter.
6. Choose Save.

AWS CLI

To include all tags in event notifications

Use the register-instance-event-notification-attributes AWS CLI command and set the IncludeAllTagsOfInstance parameter to true.
To include specific tags in event notifications

Use the `register-instance-event-notification-attributes` AWS CLI command and specify the tags to include by using the `InstanceTagKeys` parameter.

```
aws ec2 register-instance-event-notification-attributes --instance-tag-attribute
  'InstanceTagKeys=["tag_key_1", "tag_key_2", "tag_key_3"]'
```

Remove tags from event notifications

You can remove tags from event notifications by using one of the following methods.

New console

**To remove tags from event notifications**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Events.
3. Choose Actions, Manage event notifications.
4. Do one of the following, depending on the tag that you want to remove from event notifications.
   - To remove all tags from event notifications, clear Include resource tags in event notifications.
   - To remove specific tags from event notifications, choose Remove (X) for the tags listed below the Choose the tags to include field.
5. Choose Save.

AWS CLI

**To remove all tags from event notifications**

Use the `deregister-instance-event-notification-attributes` AWS CLI command and set the `IncludeAllTagsOfInstance` parameter to false.

```
aws ec2 deregister-instance-event-notification-attributes --instance-tag-attribute
  "IncludeAllTagsOfInstance=false"
```

**To remove specific tags from event notifications**

Use the `deregister-instance-event-notification-attributes` AWS CLI command and specify the tags to remove by using the `InstanceTagKeys` parameter.

```
aws ec2 deregister-instance-event-notification-attributes --instance-tag-attribute
  'InstanceTagKeys=["tag_key_1", "tag_key_2", "tag_key_3"]'
```

View the tags to be included in event notifications

You can view the tags that are to be included in event notifications by using one of the following methods.
New console

To view the tags that are to be included in event notifications

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Events.
3. Choose Actions, Manage event notifications.

AWS CLI

To view the tags that are to be included in event notifications

Use the describe-instance-event-notification-attributes AWS CLI command.

aws ec2 describe-instance-event-notification-attributes

Work with instances scheduled to stop or retire

When AWS detects irreparable failure of the underlying host for your instance, it schedules the instance to stop or terminate, depending on the type of root device for the instance. If the root device is an EBS volume, the instance is scheduled to stop. If the root device is an instance store volume, the instance is scheduled to terminate. For more information, see Instance retirement (p. 439).

Important

Any data stored on instance store volumes is lost when an instance is stopped, hibernated, or terminated. This includes instance store volumes that are attached to an instance that has an EBS volume as the root device. Be sure to save data from your instance store volumes that you might need later before the instance is stopped, hibernated, or terminated.

Actions for Instances Backed by Amazon EBS

You can wait for the instance to stop as scheduled. Alternatively, you can stop and start the instance yourself, which migrates it to a new host. For more information about stopping your instance, in addition to information about the changes to your instance configuration when it's stopped, see Stop and start your instance (p. 425).

You can automate an immediate stop and start in response to a scheduled instance stop event. For more information, see Automating Actions for EC2 Instances in the AWS Health User Guide.

Actions for Instances Backed by Instance Store

We recommend that you launch a replacement instance from your most recent AMI and migrate all necessary data to the replacement instance before the instance is scheduled to terminate. Then, you can terminate the original instance, or wait for it to terminate as scheduled.

Work with instances scheduled for reboot

When AWS must perform tasks such as installing updates or maintaining the underlying host, it can schedule the instance or the underlying host for a reboot. You can reschedule most reboot events (p. 825) so that your instance is rebooted at a specific date and time that suits you.

If you stop your linked EC2-Classic instance (p. 1033), it is automatically unlinked from the VPC and the VPC security groups are no longer associated with the instance. You can link your instance to the VPC again after you've restarted it.
View the reboot event type

You can view whether a reboot event is an instance reboot or a system reboot by using one of the following methods.

New console

To view the type of scheduled reboot event using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Events.
3. Choose Resource type: instance from the filter list.
4. For each instance, view the value in the Event type column. The value is either system-reboot or instance-reboot.

Old console

To view the type of scheduled reboot event using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Events.
3. Choose Instance resources from the filter list.
4. For each instance, view the value in the Event Type column. The value is either system-reboot or instance-reboot.

AWS CLI

To view the type of scheduled reboot event using the AWS CLI

Use the describe-instance-status command.

```
aws ec2 describe-instance-status --instance-id i-1234567890abcdef0
```

For scheduled reboot events, the value for Code is either system-reboot or instance-reboot. The following example output shows a system-reboot event.

```
[  
  {  
    "InstanceEventId": "instance-event-0d59937288b749b32",  
    "Code": "system-reboot",  
    "Description": "The instance is scheduled for a reboot",  
    "NotAfter": "2019-03-14T22:00:00.000Z",  
    "NotBefore": "2019-03-14T20:00:00.000Z",  
    "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"
  }
]
```

Actions for instance reboot

You can wait for the instance reboot to occur within its scheduled maintenance window, reschedule (p. 825) the instance reboot to a date and time that suits you, or reboot (p. 438) the instance yourself at a time that is convenient for you.
After your instance is rebooted, the scheduled event is cleared and the event’s description is updated. The pending maintenance to the underlying host is completed, and you can begin using your instance again after it has fully booted.

**Actions for system reboot**

It is not possible for you to reboot the system yourself. You can wait for the system reboot to occur during its scheduled maintenance window, or you can reschedule (p. 825) the system reboot to a date and time that suits you. A system reboot typically completes in a matter of minutes. After the system reboot has occurred, the instance retains its IP address and DNS name, and any data on local instance store volumes is preserved. After the system reboot is complete, the scheduled event for the instance is cleared, and you can verify that the software on your instance is operating as expected.

Alternatively, if it is necessary to maintain the instance at a different time and you can't reschedule the system reboot, then you can stop and start an Amazon EBS-backed instance, which migrates it to a new host. However, the data on the local instance store volumes is not preserved. You can also automate an immediate instance stop and start in response to a scheduled system reboot event. For more information, see Automating Actions for EC2 Instances in the AWS Health User Guide. For an instance store-backed instance, if you can't reschedule the system reboot, then you can launch a replacement instance from your most recent AMI, migrate all necessary data to the replacement instance before the scheduled maintenance window, and then terminate the original instance.

**Work with instances scheduled for maintenance**

When AWS must maintain the underlying host for an instance, it schedules the instance for maintenance. There are two types of maintenance events: network maintenance and power maintenance.

During network maintenance, scheduled instances lose network connectivity for a brief period of time. Normal network connectivity to your instance is restored after maintenance is complete.

During power maintenance, scheduled instances are taken offline for a brief period, and then rebooted. When a reboot is performed, all of your instance's configuration settings are retained.

After your instance has rebooted (this normally takes a few minutes), verify that your application is working as expected. At this point, your instance should no longer have a scheduled event associated with it, or if it does, the description of the scheduled event begins with [Completed]. It sometimes takes up to 1 hour for the instance status description to refresh. Completed maintenance events are displayed on the Amazon EC2 console dashboard for up to a week.

**Actions for Instances Backed by Amazon EBS**

You can wait for the maintenance to occur as scheduled. Alternatively, you can stop and start the instance, which migrates it to a new host. For more information about stopping your instance, in addition to information about the changes to your instance configuration when it's stopped, see Stop and start your instance (p. 425).

You can automate an immediate stop and start in response to a scheduled maintenance event. For more information, see Automating Actions for EC2 Instances in the AWS Health User Guide.

**Actions for instances backed by instance store**

You can wait for the maintenance to occur as scheduled. Alternatively, if you want to maintain normal operation during a scheduled maintenance window, you can launch a replacement instance from your most recent AMI, migrate all necessary data to the replacement instance before the scheduled maintenance window, and then terminate the original instance.

**Reschedule a scheduled event**

You can reschedule an event so that it occurs at a specific date and time that suits you. Only events that have a deadline date can be rescheduled. There are other limitations for rescheduling an event (p. 827).
You can reschedule an event by using one of the following methods.

New console

To reschedule an event using the console
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Events.
3. Choose Resource type: instance from the filter list.
4. Select one or more instances, and then choose Actions, Schedule event.
   Only events that have an event deadline date, indicated by a value for Deadline, can be rescheduled. If one of the selected events does not have a deadline date, Actions, Schedule event is disabled.
5. For New start time, enter a new date and time for the event. The new date and time must occur before the Event deadline.
6. Choose Save.
   It might take a minute or 2 for the updated event start time to be reflected in the console.

Old console

To reschedule an event using the console
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Events.
3. Choose Instance resources from the filter list.
4. Select one or more instances, and then choose Actions, Schedule Event.
   Only events that have an event deadline date, indicated by a value for Event Deadline, can be rescheduled.
5. For Event start time, enter a new date and time for the event. The new date and time must occur before the Event Deadline.
   It might take a minute or 2 for the updated event start time to be reflected in the console.

AWS CLI

To reschedule an event using the AWS CLI
1. Only events that have an event deadline date, indicated by a value for NotBeforeDeadline, can be rescheduled. Use the describe-instance-status command to view the NotBeforeDeadline parameter value.

   ```
   aws ec2 describe-instance-status --instance-id i-1234567890abcdef0
   ```

   The following example output shows a system-reboot event that can be rescheduled because NotBeforeDeadline contains a value.

   ```
   [ "Events": [
   { "InstanceId": "instance-event-0d59937288b749b32",
   ```
Scheduled events

"Code": "system-reboot",
"Description": "The instance is scheduled for a reboot",
"NotAfter": "2019-03-14T22:00:00.000Z",
"NotBefore": "2019-03-14T20:00:00.000Z",
"NotBeforeDeadline": "2019-04-05T11:00:00.000Z"
]
]

2. To reschedule the event, use the `modify-instance-event-start-time` command. Specify the new event start time by using the `not-before` parameter. The new event start time must fall before the `NotBeforeDeadline`.

```
aws ec2 modify-instance-event-start-time --instance-id i-1234567890abcdef0
   --instance-event-id instance-event-0d59937288b749b32 --not-before 2019-03-25T10:00:00.000
```

It might take a minute or 2 before the `describe-instance-status` command returns the updated `not-before` parameter value.

**Limitations**

- Only events with an event deadline date can be rescheduled. The event can be rescheduled up to the event deadline date. The `Deadline` column in the console and the `NotBeforeDeadline` field in the AWS CLI indicate if the event has a deadline date.
- Only events that have not yet started can be rescheduled. The `Start time` column in the console and the `NotBefore` field in the AWS CLI indicate the event start time. Events that are scheduled to start in the next 5 minutes cannot be rescheduled.
- The new event start time must be at least 60 minutes from the current time.
- If you reschedule multiple events using the console, the event deadline date is determined by the event with the earliest event deadline date.

**Define event windows for scheduled events**

You can define custom event windows that recur weekly for scheduled events that reboot, stop, or terminate your Amazon EC2 instances. You can associate one or more instances with an event window. If a scheduled event for those instances is planned, AWS will schedule the events within the associated event window.

You can use event windows to maximize workload availability by specifying event windows that occur during off-peak periods for your workload. You can also align the event windows with your internal maintenance schedules.

You define an event window by specifying a set of time ranges. The minimum time range is 2 hours. The combined time ranges must total at least 4 hours.

You can associate one or more instances with an event window by using either instance IDs or instance tags. You can also associate Dedicated Hosts with an event window by using the host ID.

**Warning**

Event windows are applicable only for scheduled events that stop, reboot, or terminate instances.

Event windows are not applicable for:

- Expedited scheduled events and network maintenance events.
- Unscheduled maintenance such as AutoRecovery and unplanned reboots.
Work with event windows

- Considerations (p. 828)
- View event windows (p. 828)
- Create event windows (p. 830)
- Modify event windows (p. 834)
- Delete event windows (p. 838)
- Tag event windows (p. 839)

Considerations

- All event window times are in UTC.
- The minimum weekly event window duration is 4 hours.
- The time ranges within an event window must each be at least 2 hours.
- Only one target type (instance ID, Dedicated Host ID, or instance tag) can be associated with an event window.
- A target (instance ID, Dedicated Host ID, or instance tag) can only be associated with one event window.
- A maximum of 100 instance IDs, or 50 Dedicated Host IDs, or 50 instance tags can be associated with an event window. The instance tags can be associated with any number of instances.
- A maximum of 200 event windows can be created per AWS Region.
- Multiple instances that are associated with event windows can potentially have scheduled events occur at the same time.
- If AWS has already scheduled an event, modifying an event window won't change the time of the scheduled event. If the event has a deadline date, you can reschedule the event (p. 825).
- You can stop and start an instance prior to the scheduled event, which migrates the instance to a new host, and the scheduled event will no longer take place.

View event windows

You can view event windows by using one of the following methods.

Console

To view event windows using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Events.
3. Choose Actions, Manage event windows.
4. Select an event window to view its details.

AWS CLI

To describe all event windows using the AWS CLI

Use the describe-instance-event-windows command.

```
aws ec2 describe-instance-event-windows \
   --region us-east-1
```

Expected output
Scheduled events

```json
{
    "InstanceEventWindows": [
        {
            "InstanceEventWindowId": "iew-0abcdef1234567890",
            "Name": "myEventWindowName",
            "CronExpression": "* 21-23 * * 2,3",
            "AssociationTarget": {
                "InstanceId": [
                    "i-1234567890abcdef0",
                    "i-0598c7d356eba48d7"
                ],
                "Tags": [],
                "DedicatedHostIds": []
            },
            "State": "active",
            "Tags": []
        }
    ],
    "NextToken": "9d624e0c-388b-4862-a31e-a85c64fc1d4a"
}
```

To describe a specific event window using the AWS CLI

Use the `describe-instance-event-windows` command with the `--instance-event-window-id` parameter to describe a specific event window.

```
aws ec2 describe-instance-event-windows \
   --region us-east-1 \
   --instance-event-window-id iew-0abcdef1234567890
```

To describe event windows that match one or more filters using the AWS CLI

Use the `describe-instance-event-windows` command with the `--filters` parameter. In the following example, the `instance-id` filter is used to describe all of the event windows that are associated with the specified instance.

When a filter is used, it performs a direct match. However, the `instance-id` filter is different. If there is no direct match to the instance ID, then it falls back to indirect associations with the event window, such as the instance's tags or Dedicated Host ID (if the instance is on a Dedicated Host).

For the list of supported filters, see `describe-instance-event-windows` in the [AWS CLI Reference](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/describe-instance-event-windows.html).

```
aws ec2 describe-instance-event-windows \
   --region us-east-1 \
   --filters Name=instance-id,Values=i-1234567890abcdef0 \
   --max-results 100 \
   --next-token <next-token-value>
```

Expected output

In the following example, the instance is on a Dedicated Host, which is associated with the event window.

```json
{
    "InstanceEventWindows": [
        {
            "InstanceEventWindowId": "iew-0db0c0adb66f335982",
            "TimeRanges": [
```
Create event windows

You can create one or more event windows. For each event window, you specify one or more blocks of time. For example, you can create an event window with blocks of time that occur every day at 4 AM for 2 hours. Or you can create an event window with blocks of time that occur on Sundays from 2 AM to 4 AM and on Wednesdays from 3 AM to 5 AM.

For the event window constraints, see Considerations (p. 828) earlier in this topic.

Event windows recur weekly until you delete them.

Use one of the following methods to create an event window.

Console

To create an event window using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Events.
3. Choose Create instance event window.
4. For Event window name, enter a descriptive name for the event window.
5. For Event window schedule, choose to specify the blocks of time in the event window by using the cron schedule builder or by specifying time ranges.
   - If you choose Cron schedule builder, specify the following:
     1. For Days (UTC), specify the days of the week on which the event window occurs.
     2. For Start time (UTC), specify the time when the event window begins.
     3. For Duration, specify the duration of the blocks of time in the event window. The minimum duration per block of time is 2 hours. The minimum duration of the event window must equal or exceed 4 hours in total. All times are in UTC.
   - If you choose Time ranges, choose Add new time range and specify the start day and time and the end day and time. Repeat for each time range. The minimum duration per time range is 2 hours. The minimum duration for all time ranges combined must equal or exceed 4 hours in total.
6. (Optional) For Target details, associate one or more instances with the event window so that if the instances are scheduled for maintenance, the scheduled event will occur during the
associated event window. You can associate one or more instances with an event window by using instance IDs or instance tags. You can associate Dedicated Hosts with an event window by using the host ID.

Note that you can create the event window without associating a target with the window. Later, you can modify the window to associate one or more targets.

7. (Optional) For **Event window tags**, choose **Add tag**, and enter the key and value for the tag. Repeat for each tag.

8. Choose **Create event window**.

**AWS CLI**

To create an event window using the AWS CLI, you first create the event window, and then you associate one or more targets with the event window.

**Create an event window**

You can define either a set of time ranges or a cron expression when creating the event window, but not both.

**To create an event window with a time range using the AWS CLI**

Use the `create-instance-event-window` command and specify the `--time-range` parameter. You can’t also specify the `--cron-expression` parameter.

```
aws ec2 create-instance-event-window \
  --region us-east-1 \
  --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8 \
  --tag-specifications "ResourceType=instance-event-window,Tags=[{Key=K1,Value=V1}]" \
  --name myEventWindowName
```

**Expected output**

```json
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "TimeRanges": [
      {
        "StartWeekDay": "monday",
        "StartHour": 2,
        "EndWeekDay": "wednesday",
        "EndHour": 8
      }
    ],
    "Name": "myEventWindowName",
    "State": "creating",
    "Tags": [
      {
        "Key": "K1",
        "Value": "V1"
      }
    ]
  }
}
```

**To create an event window with a cron expression using the AWS CLI**

Use the `create-instance-event-window` command and specify the `--cron-expression` parameter. You can’t also specify the `--time-range` parameter.
aws ec2 create-instance-event-window \
  --region us-east-1 \
  --cron-expression "* 21-23 * * 2,3" \
  --tag-specifications "ResourceType=instance-event-window,Tags=[{Key=K1,Value=V1}]" \
  --name myEventWindowName

Expected output

{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "State": "creating",
    "Tags": [
      {
        "Key": "K1",
        "Value": "V1"
      }
    ]
  }
}

Associate a target with an event window

You can associate only one type of target (instance IDs, Dedicated Host IDs, or instance tags) with an event window.

To associate instance tags with an event window using the AWS CLI

Use the `associate-instance-event-window` command and specify the `instance-event-window-id` parameter to specify the event window. To associate instance tags, specify the `--association-target` parameter, and for the parameter values, specify one or more tags.

aws ec2 associate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"

Expected output

{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [
        {
          "Key": "k2",
          "Value": "v2"
        },
        {
          "Key": "k1",
          "Value": "v1"
        }
      ],
      "DedicatedHostIds": []
    }
  }
}
To associate one or more instances with an event window using the AWS CLI

Use the `associate-instance-event-window` command and specify the `instance-event-window-id` parameter to specify the event window. To associate instances, specify the `--association-target` parameter, and for the parameter values, specify one or more instance IDs.

```bash
aws ec2 associate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eb048d7"
```

Expected output

```json
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [
        "i-1234567890abcdef0",
        "i-0598c7d356eb048d7"
      ],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}
```

To associate a Dedicated Host with an event window using the AWS CLI

Use the `associate-instance-event-window` command and specify the `instance-event-window-id` parameter to specify the event window. To associate a Dedicated Host, specify the `--association-target` parameter, and for the parameter values, specify one or more Dedicated Host IDs.

```bash
aws ec2 associate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "DedicatedHostIds=h-029fa35a02b99801d"
```

Expected output

```json
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": [
        "h-029fa35a02b99801d"
      ]
    }
  }
}
```
Modify event windows

You can modify all of the fields of an event window except its ID. For example, when daylight savings begin, you might want to modify the event window schedule. For existing event windows, you might want to add or remove targets.

Use one of the following methods to modify an event window.

Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose ♦ Events.
3. Choose ♦ Actions, Manage event windows.
4. Select the event window to modify, and then choose Actions, Modify instance event window.
5. Modify the fields in the event window, and then choose Modify event window.

AWS CLI

To modify an event window using the AWS CLI, you can modify the time range or cron expression, and associate or disassociate one or more targets with the event window.

Modify the event window time

You can modify either a time range or a cron expression when modifying the event window, but not both.

To modify the time range of an event window using the AWS CLI

Use the modify-instance-event-window command and specify the event window to modify. Specify the --time-range parameter to modify the time range. You can’t also specify the --cron-expression parameter.

```bash
aws ec2 modify-instance-event-window \
  --region us-east-1 \
  --instance-event-window-idiew-0abcdef1234567890 \
  --time-range StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8
```

Expected output

```json
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "TimeRanges": [
      {
        "StartWeekDay": "monday",
        "StartHour": 2,
        "EndWeekDay": "wednesday",
        "EndHour": 8
      }
    ]
  }
}
```
To modify a set of time ranges for an event window using the AWS CLI

Use the `modify-instance-event-window` command and specify the event window to modify. Specify the `--time-range` parameter to modify the time range. You can't also specify the `--cron-expression` parameter in the same call.

```
aws ec2 modify-instance-event-window
  --region us-east-1
  --instance-event-window-id iev-0abcdef1234567890
  --time-range '[["StartWeekDay": "monday", "StartHour": 2, "EndWeekDay": "wednesday", "EndHour": 8],
  {"StartWeekDay": "thursday", "StartHour": 2, "EndWeekDay": "friday", "EndHour": 8}]'
```

**Expected output**

```
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "TimeRanges": [
      {
        "StartWeekDay": "monday",
        "StartHour": 2,
        "EndWeekDay": "wednesday",
        "EndHour": 8
      },
      {
        "StartWeekDay": "thursday",
        "StartHour": 2,
        "EndWeekDay": "friday",
        "EndHour": 8
      }
    ],
    "Name": "myEventWindowName",
    "AssociationTarget": {
      "InstanceIds": [
        "i-0abcdef1234567890",
        "i-0be35f9ac08ba01f0"
      ],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating",
    "Tags": [
      {
        "Key": "K1",
        "Value": "V1"
      }
    ]
  }
}
```
To modify the cron expression of an event window using the AWS CLI

Use the `modify-instance-event-window` command and specify the event window to modify. Specify the `--cron-expression` parameter to modify the cron expression. You can’t also specify the `--time-range` parameter.

```bash
aws ec2 modify-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --cron-expression "* 21-23 * * 2,3"
```

Expected output

```json
{

  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceId": [
        "i-0abcdef1234567890",
        "i-0be35f9ac8ba01f0"
      ],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating",
    "Tags": [
      {
        "Key": "K1",
        "Value": "V1"
      }
    ]
  }
}
```

Modify the targets associated with an event window

You can associate additional targets with an event window. You can also disassociate existing targets from an event window. However, only one type of target (instance IDs, Dedicated Host IDs, or instance tags) can be associated with an event window.

To associate additional targets with an event window

For the instructions on how to associate targets with an event window, see Associate a target with an event window.

To disassociate instance tags from an event window using the AWS CLI

Use the `disassociate-instance-event-window` command and specify the `instance-event-window-id` parameter to specify the event window. To disassociate instance tags, specify the `--association-target` parameter, and for the parameter values, specify one or more tags.
aws ec2 disassociate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "InstanceTags=[{Key=k2,Value=v2},{Key=k1,Value=v1}]"

Expected output

```
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}
```

To disassociate one or more instances from an event window using the AWS CLI

Use the `disassociate-instance-event-window` command and specify the `instance-event-window-id` parameter to specify the event window. To disassociate instances, specify the `--association-target` parameter, and for the parameter values, specify one or more instance IDs.

aws ec2 disassociate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"

Expected output

```
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}
```

To disassociate a Dedicated Host from an event window using the AWS CLI

Use the `disassociate-instance-event-window` command and specify the `instance-event-window-id` parameter to specify the event window. To disassociate a Dedicated Host, specify the `--association-target` parameter, and for the parameter values, specify one or more Dedicated Host IDs.

aws ec2 disassociate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target DedicatedHostIds=h-029fa350a02b99801d
Scheduled events

Expected output

```json
{
   "InstanceEventWindow": {
      "InstanceEventWindowId": "iew-0abcdef1234567890",
      "Name": "myEventWindowName",
      "CronExpression": "* 21-23 * * 2,3",
      "AssociationTarget": {
         "InstanceIds": [],
         "Tags": [],
         "DedicatedHostIds": []
      },
      "State": "creating"
   }
}
```

Delete event windows

You can delete one event window at a time by using one of the following methods.

Console

To delete an event window using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Events.
3. Choose Actions, Manage event windows.
4. Select the event window to delete, and then choose Actions, Delete instance event window.
5. When prompted, enter `delete`, and then choose Delete.

AWS CLI

To delete an event window using the AWS CLI

Use the `delete-instance-event-window` command and specify the event window to delete.

```
aws ec2 delete-instance-event-window
   --region us-east-1
   --instance-event-window-id iew-0abcdef1234567890
```

To force delete an event window using the AWS CLI

Use the `--force-delete` parameter if the event window is currently associated with targets.

```
aws ec2 delete-instance-event-window
   --region us-east-1
   --instance-event-window-id iew-0abcdef1234567890
   --force-delete
```

Expected output

```json
{
   "InstanceEventWindowState": {
      "InstanceEventWindowId": "iew-0abcdef1234567890",
      "State": "deleting"
   }
}
```
Tag event windows

You can tag an event window when you create it, or afterwards.

To tag an event window when you create it, see Create event windows (p. 830).

Use one of the following methods to tag an event window.

Console

To tag an existing event window using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Events.
3. Choose Actions, Manage event windows.
4. Select the event window to tag, and then choose Actions, Manage instance event window tags.
5. Choose Add tag to add a tag. Repeat for each tag.
6. Choose Save.

AWS CLI

To tag an existing event window using the AWS CLI

Use the create-tags command to tag existing resources. In the following example, the existing event window is tagged with Key=purpose and Value=test.

```
aws ec2 create-tags \\n  --resources iew-0abcdef1234567890 \\
  --tags Key=purpose,Value=test
```

Monitor your instances using CloudWatch

You can monitor your instances using Amazon CloudWatch, which collects and processes raw data from Amazon EC2 into readable, near real-time metrics. These statistics are recorded for a period of 15 months, so that you can access historical information and gain a better perspective on how your web application or service is performing.

By default, Amazon EC2 sends metric data to CloudWatch in 5-minute periods. To send metric data for your instance to CloudWatch in 1-minute periods, you can enable detailed monitoring on the instance. For more information, see Enable or turn off detailed monitoring for your instances (p. 840).

The Amazon EC2 console displays a series of graphs based on the raw data from Amazon CloudWatch. Depending on your needs, you might prefer to get data for your instances from Amazon CloudWatch instead of the graphs in the console.

For more information about Amazon CloudWatch, see the Amazon CloudWatch User Guide.

Contents

- Enable or turn off detailed monitoring for your instances (p. 840)
- List the available CloudWatch metrics for your instances (p. 842)
Amazon Elastic Compute Cloud
User Guide for Windows Instances
Enable detailed monitoring

- Get statistics for metrics for your instances (p. 853)
- Graph metrics for your instances (p. 861)
- Create a CloudWatch alarm for an instance (p. 861)
- Create alarms that stop, terminate, reboot, or recover an instance (p. 863)

Enable or turn off detailed monitoring for your instances

By default, your instance is enabled for basic monitoring. You can optionally enable detailed monitoring. After you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period for the instance.

The following describes the data interval and charge for basic and detailed monitoring for instances.

<table>
<thead>
<tr>
<th>Monitoring type</th>
<th>Description</th>
<th>Charges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic monitoring</td>
<td>Data is available automatically in 5-minute periods.</td>
<td>No charge</td>
</tr>
<tr>
<td>Detailed monitoring</td>
<td>Data is available in 1-minute periods. To get this level of data, you must</td>
<td>You are charged per metric that is sent to CloudWatch. You are not</td>
</tr>
<tr>
<td></td>
<td>specifically enable it for the instance. For the instances where you’ve</td>
<td>charged for data storage. For more information, see Paid tier and</td>
</tr>
<tr>
<td></td>
<td>enabled detailed monitoring, you can also get aggregated data across groups</td>
<td>Example 1 - EC2 Detailed Monitoring on the Amazon CloudWatch pricing</td>
</tr>
<tr>
<td></td>
<td>of similar instances.</td>
<td>page.</td>
</tr>
</tbody>
</table>

Topics
- Required IAM permissions (p. 840)
- Enable detailed monitoring (p. 840)
- Turn off detailed monitoring (p. 841)

Required IAM permissions

To enable detailed monitoring for an instance, your IAM user must have permission to use the MonitorInstances API action. To turn off detailed monitoring for an instance, your IAM user must have permission to use the UnmonitorInstances API action.

Enable detailed monitoring

You can enable detailed monitoring on an instance as you launch it or after the instance is running or stopped. Enabling detailed monitoring on an instance does not affect the monitoring of the EBS volumes attached to the instance. For more information, see Amazon CloudWatch metrics for Amazon EBS (p. 1375).

New console

To enable detailed monitoring for 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 the instance and choose **Actions, Monitoring, Manage detailed monitoring**.
4. On the **Detailed monitoring** detail page, for **Detailed monitoring**, select the **Enable** check box.
5. Choose **Save**.

**To enable detailed monitoring when launching an instance**

When launching an instance using the AWS Management Console, select the **Monitoring** check box on the **Configure Instance Details** page.

**Old console**

**To enable detailed monitoring for an existing instance**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](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.

**AWS CLI**

**To enable detailed monitoring for an existing instance**

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**

Use the `run-instances` command with the `--monitoring` flag to enable detailed monitoring.

```
aws ec2 run-instances --image-id ami-09092360 --monitoring Enabled=true...
```

**Turn off detailed monitoring**

You can turn off detailed monitoring on an instance as you launch it or after the instance is running or stopped.

**New console**

**To turn off detailed monitoring**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions, Monitoring, Manage detailed monitoring**.
4. On the **Detailed monitoring** detail page, for **Detailed monitoring**, clear the **Enable** check box.

5. Choose **Save**.

**Old console**

**To turn off detailed monitoring**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](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**.

**AWS CLI**

**To turn off detailed monitoring**

Use the following `unmonitor-instances` command to turn off detailed monitoring for the specified instances.

```bash
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

**List the available CloudWatch metrics for your instances**

Amazon EC2 sends metrics to Amazon CloudWatch. You can use the AWS Management Console, the AWS CLI, or an API to list the metrics that Amazon EC2 sends to CloudWatch. By default, each data point covers the 5 minutes that follow the start time of activity for the instance. If you've enabled detailed monitoring, each data point covers the next minute of activity from the start time.

For information about getting the statistics for these metrics, see [Get statistics for metrics for your instances](p. 853).

**Contents**

- Instance metrics (p. 842)
- CPU credit metrics (p. 845)
- Amazon EBS metrics for Nitro-based instances (p. 846)
- Status check metrics (p. 848)
- Traffic mirroring metrics (p. 849)
- Amazon EC2 metric dimensions (p. 849)
- Amazon EC2 usage metrics (p. 849)
- List metrics using the console (p. 850)
- List metrics using the AWS CLI (p. 852)

**Instance metrics**

The **AWS/EC2** namespace includes the following instance metrics.
<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPUUtilization</td>
<td>The percentage of allocated EC2 compute units that are currently in use on the instance. This metric identifies the processing power required to run an application on a selected instance.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Units: Percent</td>
</tr>
<tr>
<td>DiskReadOps</td>
<td>Completed read operations from all instance store volumes available to the instance in a specified period of time.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>If there are no instance store volumes, either the value is 0 or the metric is not reported.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td>DiskWriteOps</td>
<td>Completed write operations to all instance store volumes available to the instance in a specified period of time.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>If there are no instance store volumes, either the value is 0 or the metric is not reported.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td>DiskReadBytes</td>
<td>Bytes read from all instance store volumes available to the instance.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>If there are no instance store volumes, either the value is 0 or the metric is not reported.</td>
</tr>
<tr>
<td></td>
<td>Units: Bytes</td>
</tr>
<tr>
<td>DiskWriteBytes</td>
<td>Bytes written to all instance store volumes available to the instance.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetadataNoToken</td>
<td>The number of times the instance metadata service was successfully accessed using a method that does not use a token. This metric is used to determine if there are any processes accessing instance metadata that are using Instance Metadata Service Version 1, which does not use a token. If all requests use token-backed sessions, i.e., Instance Metadata Service Version 2, the value is 0. For more information, see Transition to using Instance Metadata Service Version 2 (p. 581).</td>
</tr>
<tr>
<td>NetworkIn</td>
<td>The number of bytes received by the instance on all network interfaces. This metric identifies the volume of incoming network traffic to a single instance. The number reported is the number of bytes received during the period. If you are using basic (five-minute) monitoring and the statistic is Sum, you can divide this number by 300 to find Bytes/second. If you have detailed (one-minute) monitoring and the statistic is Sum, divide it by 60.</td>
</tr>
<tr>
<td>NetworkOut</td>
<td>The number of bytes sent out by the instance on all network interfaces. This metric identifies the volume of outgoing network traffic from a single instance. The number reported is the number of bytes sent during the period. If you are using basic (five-minute) monitoring and the statistic is Sum, you can divide this number by 300 to find Bytes/second. If you have detailed (one-minute) monitoring and the statistic is Sum, divide it by 60.</td>
</tr>
<tr>
<td>NetworkPacketsIn</td>
<td>The number of packets received by the instance on all network interfaces. 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 (five-minute periods). To calculate the number of packets per second (PPS) your instance received, divide this number by 300.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Statistics: Minimum, Maximum, Average</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NetworkPacketsOut</td>
<td>The number of packets sent out by the instance on all network interfaces. This metric identifies the volume of outgoing traffic in terms of the number of packets on a single instance.</td>
</tr>
<tr>
<td></td>
<td>This metric is available for basic monitoring only (five-minute periods). To calculate the number of packets per second (PPS) your instance received, divide this number by 300.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Statistics: Minimum, Maximum, Average</td>
</tr>
</tbody>
</table>

**CPU credit metrics**

The AWS/EC2 namespace includes the following CPU credit metrics for your burstable performance instances (p. 160).

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPUCreditUsage</td>
<td>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).</td>
</tr>
<tr>
<td></td>
<td>CPU credit metrics are available at a five-minute frequency only. If you specify a period greater than five minutes, use the <strong>Sum</strong> statistic instead of the <strong>Average</strong> statistic.</td>
</tr>
<tr>
<td></td>
<td>Units: Credits (vCPU-minutes)</td>
</tr>
<tr>
<td>CPUCreditBalance</td>
<td>The number of earned CPU credits that an instance has accrued since it was launched or started. For T2 Standard, the CPUCreditBalance also includes the number of launch credits that have been accrued.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>The credits in the CPUCreditBalance are available for the instance to spend to burst beyond its baseline CPU utilization.</td>
</tr>
<tr>
<td></td>
<td>When an instance is running, credits in the CPUCreditBalance do not expire. When a T3 or T3a instance stops, the CPUCreditBalance value persists for seven days. Thereafter, all accrued credits are lost. When a T2 instance stops, the CPUCreditBalance value does not persist, and all accrued credits are lost.</td>
</tr>
<tr>
<td></td>
<td>CPU credit metrics are available at a five-minute frequency only.</td>
</tr>
</tbody>
</table>
### List available metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPUSurplusCreditBalance</strong></td>
<td>The number of surplus credits that have been spent by an unlimited instance when its CPUCreditBalance value is zero. 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. CPU credit metrics are available at a five-minute frequency only.</td>
</tr>
</tbody>
</table>
| **CPUSurplusCreditsCharged** | The number of spent surplus credits that are not paid down by earned CPU credits, and which thus incur an additional charge. Spent surplus credits are charged when any of the following occurs:  
  • 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.  
CPU credit metrics are available at a five-minute frequency only. |

### Amazon EBS metrics for Nitro-based instances

The AWS/EC2 namespace includes the following Amazon EBS metrics for the Nitro-based instances that are not bare metal instances. For the list of Nitro-based instance types, see [Instances built on the Nitro System](p. 146).

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.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBSReadOps</td>
<td>Completed read operations from all Amazon EBS volumes attached to the instance in a specified period of time. To calculate the average read I/O operations per second (Read IOPS) for the period, divide the total operations in the period by the number of seconds in that period. If you are using basic (five-minute) monitoring, you can divide this number by 300 to calculate the Read IOPS. If you have detailed (one-minute) monitoring, divide it by 60.</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EBSWriteOps</td>
<td>Completed write operations to all EBS volumes attached to the instance in a specified period of time. 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.</td>
</tr>
<tr>
<td>EBSReadBytes</td>
<td>Bytes read from all EBS volumes attached to the instance in a specified period of time. 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.</td>
</tr>
<tr>
<td>EBSWriteBytes</td>
<td>Bytes written to all EBS volumes attached to the instance in a specified period of time. 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.</td>
</tr>
<tr>
<td>EBSI0Balance%</td>
<td>Provides information about the percentage of I/O credits remaining in the burst bucket. This metric is available for basic monitoring only. Instance sizes that support this metric can be found in the table under EBS optimized by default (p. 1345): the instances in the Instance size column that include an asterisk (*) support this metric. The Sum statistic is not applicable to this metric.</td>
</tr>
</tbody>
</table>

Unit: Count

Unit: Count

Unit: Bytes

Unit: Bytes

Unit: Percent

847
List available metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBSByteBalance%</td>
<td>Provides information about the percentage of throughput credits remaining in the burst bucket. This metric is available for basic monitoring only.</td>
</tr>
<tr>
<td></td>
<td>Instance sizes that support this metric can be found in the table under EBS optimized by default (p. 1345): the instances in the Instance size column that include an asterisk (*) support this metric.</td>
</tr>
<tr>
<td></td>
<td>The Sum statistic is not applicable to this metric.</td>
</tr>
<tr>
<td></td>
<td>Unit: Percent</td>
</tr>
</tbody>
</table>

For information about the metrics provided for your EBS volumes, see Amazon EBS metrics (p. 1375). For information about the metrics provided for your Spot fleets, see CloudWatch metrics for Spot Fleet (p. 737).

**Status check metrics**

The AWS/EC2 namespace includes the following status check metrics. By default, status check metrics are available at a 1-minute frequency at no charge. For a newly-launched instance, status check metric data is only available after the instance has completed the initialization state (within a few minutes of the instance entering the running state). For more information about EC2 status checks, see Status checks for your instances (p. 809).

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StatusCheckFailed</td>
<td>Reports whether the instance has passed both the instance status check and the system status check in the last minute.</td>
</tr>
<tr>
<td></td>
<td>This metric can be either 0 (passed) or 1 (failed).</td>
</tr>
<tr>
<td></td>
<td>By default, this metric is available at a 1-minute frequency at no charge.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td>StatusCheckFailed_Instance</td>
<td>Reports whether the instance has passed the instance status check in the last minute.</td>
</tr>
<tr>
<td></td>
<td>This metric can be either 0 (passed) or 1 (failed).</td>
</tr>
<tr>
<td></td>
<td>By default, this metric is available at a 1-minute frequency at no charge.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td>StatusCheckFailed_System</td>
<td>Reports whether the instance has passed the system status check in the last minute.</td>
</tr>
<tr>
<td></td>
<td>This metric can be either 0 (passed) or 1 (failed).</td>
</tr>
<tr>
<td></td>
<td>By default, this metric is available at a 1-minute frequency at no charge.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
</tbody>
</table>
Traffic mirroring metrics

The AWS/EC2 namespace includes metrics for mirrored traffic. For more information, see Monitoring mirrored traffic using Amazon CloudWatch in the Amazon VPC Traffic Mirroring Guide.

Amazon EC2 metric dimensions

You can use the following dimensions to refine the metrics listed in the previous tables.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoScalingGroupName</td>
<td>This dimension filters the data you request for all instances in a specified capacity group. An Auto Scaling group 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.</td>
</tr>
<tr>
<td>ImageId</td>
<td>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.</td>
</tr>
<tr>
<td>InstanceId</td>
<td>This dimension filters the data you request for the identified instance only. This helps you pinpoint an exact instance from which to monitor data.</td>
</tr>
<tr>
<td>InstanceType</td>
<td>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.</td>
</tr>
</tbody>
</table>

Amazon EC2 usage metrics

You can use CloudWatch usage metrics to provide visibility into your account's usage of resources. Use these metrics to visualize your current service usage on CloudWatch graphs and dashboards.

Amazon EC2 usage metrics correspond to AWS service quotas. You can configure alarms that alert you when your usage approaches a service quota. For more information about CloudWatch integration with service quotas, see Service Quotas Integration and Usage Metrics.

Amazon EC2 publishes the following metrics in the AWS/Usage namespace.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResourceCount</td>
<td>The number of the specified resources running in your account. The resources are defined by the dimensions associated with the metric.</td>
</tr>
<tr>
<td></td>
<td>The most useful statistic for this metric is MAXIMUM, which represents the maximum number of resources used during the 1-minute period.</td>
</tr>
</tbody>
</table>
The following dimensions are used to refine the usage metrics that are published by Amazon EC2.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>The name of the AWS service containing the resource. For Amazon EC2 usage metrics, the value for this dimension is EC2.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of entity that is being reported. Currently, the only valid value for Amazon EC2 usage metrics is Resource.</td>
</tr>
<tr>
<td>Resource</td>
<td>The type of resource that is running. Currently, the only valid value for Amazon EC2 usage metrics is vCPU, which returns information on instances that are running.</td>
</tr>
<tr>
<td>Class</td>
<td>The class of resource being tracked. For Amazon EC2 usage metrics with vCPU as the value of the Resource dimension, the valid values are Standard/OnDemand, F/OnDemand, G/OnDemand, Inf/OnDemand, P/OnDemand, and X/OnDemand. The values for this dimension define the first letter of the instance types that are reported by the metric. For example, Standard/OnDemand returns information about all running instances with types that start with A, C, D, H, I, M, R, T, and Z, and G/OnDemand returns information about all running instances with types that start with G.</td>
</tr>
</tbody>
</table>

**List 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)**

2. In the navigation pane, choose Metrics.
3. Choose the EC2 metric namespace.
4. Select a metric dimension (for example, **Per-Instance 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**.

---

**List 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.

```bash
aws cloudwatch list-metrics --namespace AWS/EC2
```

The following is example output:

```
{
   "Metrics": [
   {
      "Namespace": "AWS/EC2",
      "Dimensions": [
      {
         "Name": "InstanceId",
         "Value": "i-1234567890abcdef0"
      }
      ],
      "MetricName": "NetworkOut"
   },
   {
      "Namespace": "AWS/EC2",
      "Dimensions": [
      {
         "Name": "InstanceId",
         "Value": "i-1234567890abcdef0"
      }
      ],
      "MetricName": "NetworkIn"
   }
   ]
}```
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.

```bash
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.

```bash
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. 853)
- Get statistics for a specific instance (p. 854)
- Aggregate statistics across instances (p. 857)
- Aggregate statistics by Auto Scaling group (p. 859)
- Aggregate statistics by AMI (p. 860)

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.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>The lowest value observed during the specified period. You can use this value to determine low volumes of activity for your application.</td>
</tr>
<tr>
<td>Statistic</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Maximum</td>
<td>The highest value observed during the specified period. You can use this value to determine high volumes of activity for your application.</td>
</tr>
<tr>
<td>Sum</td>
<td>All values submitted for the matching metric added together. This statistic can be useful for determining the total volume of a metric.</td>
</tr>
<tr>
<td>Average</td>
<td>The value of $\text{Sum} / \text{SampleCount}$ during the specified period. By comparing this statistic with the Minimum and Maximum, you can determine the full scope of a metric and how close the average use is to the Minimum and Maximum. This comparison helps you to know when to increase or decrease your resources as needed.</td>
</tr>
<tr>
<td>SampleCount</td>
<td>The count (number) of data points used for the statistical calculation.</td>
</tr>
<tr>
<td>$p_{NN.NN}$</td>
<td>The value of the specified percentile. You can specify any percentile, using up to two decimal places (for example, $p_{95.45}$).</td>
</tr>
</tbody>
</table>

Get statistics for a specific instance

The following examples show you how to use the AWS Management Console or the AWS CLI to determine the maximum CPU utilization of a specific EC2 instance.

Requirements

- You must have the ID of the instance. You can get the instance ID using the AWS Management Console or the `describe-instances` command.
- By default, basic monitoring is enabled, but you can enable detailed monitoring. For more information, see Enable or turn off detailed monitoring for your instances (p. 840).

To display the CPU utilization for a specific instance (console)

2. In the navigation pane, choose Metrics.
3. Choose the EC2 metric namespace.
4. Choose the **Per-Instance Metrics** dimension.
5. 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**.

![Graph of CPUUtilization metric for a specific instance](image1.png)

6. 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.

![Graph options for CPUUtilization metric](image2.png)

**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": 99.670000000000002,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2016-10-19T07:18:00Z",
      "Maximum": 0.34000000000000002,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2016-10-19T12:18:00Z",
      "Maximum": 0.34000000000000002,
      "Unit": "Percent"
    },
    ...
  ],
  "Label": "CPUUtilization"
}
```

### Aggregate statistics across instances

Aggregate statistics are available for instances that have detailed monitoring enabled. Instances that use basic monitoring are not included in the aggregates. Before you can get statistics aggregated across instances, you must enable detailed monitoring (p. 840) (at an additional charge), which provides data in 1-minute periods.

Note that Amazon CloudWatch cannot aggregate data across AWS Regions. Metrics are completely separate between Regions.

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)

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.

```bash
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:

```json
{
  "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": 0.16670833333333332,
      "Unit": "Percent"
    }
  ]
}
```
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 AWS 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)

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.

```bash
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:

```json
{
"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": 859,
    "Unit": "Bytes"
  }
]
```

"Unit": "Bytes"
Aggregate statistics by AMI

You can aggregate statistics for your instances that have detailed monitoring enabled. Instances that use basic monitoring are not included in the aggregates. Before you can get statistics aggregated across instances, you must enable detailed monitoring (p. 840) (at an additional charge), which provides data in 1-minute periods.

Note that Amazon CloudWatch cannot aggregate data across AWS Regions. Metrics are completely separate between Regions.

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)

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": 0.041000000000000009,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T14:00:00Z",
            "Average": 0.079579831932773085,
            "Unit": "Percent"
        },
        {
```

860
Graph metrics for your instances

After you launch an instance, you can open the Amazon EC2 console and view the monitoring graphs for the 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. 842).

Graph Metrics Using the CloudWatch Console

You can also use the CloudWatch console to graph metric data generated by Amazon EC2 and other AWS services. For more information, see Graph Metrics in the Amazon CloudWatch User Guide.

Create a CloudWatch alarm for an instance

You can create a CloudWatch alarm that monitors CloudWatch metrics for one of your instances. CloudWatch will automatically send you a notification when the metric reaches a threshold you specify. You can create a CloudWatch alarm using the Amazon EC2 console, or using the more advanced options provided by the CloudWatch console.

To create an alarm using the CloudWatch console

For examples, see Creating Amazon CloudWatch Alarms in the Amazon CloudWatch User Guide.

New console

To create an alarm using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
Create an alarm

3. Select the instance and choose **Actions, Monitor and troubleshoot, Manage CloudWatch alarms**.

4. On the **Manage CloudWatch alarms** detail page, under **Add or edit alarm**, select **Create an alarm**.

5. For **Alarm notification**, choose whether to turn the toggle on or off to configure Amazon Simple Notification Service (Amazon SNS) notifications. Enter an existing Amazon SNS topic or enter a name to create a new topic.

6. For **Alarm action**, choose whether to turn the toggle on or off to specify an action to take when the alarm is triggered. Select an action from the dropdown.

7. For **Alarm thresholds**, select the metric and criteria for the alarm. For example, you can leave the default settings for **Group samples by (Average)** and **Type of data to sample (CPU utilization)**. For **Alarm when**, choose **>=** and enter **0.80**. For **Consecutive period**, enter **1**. For **Period**, select **5 minutes**.

8. (Optional) For **Sample metric data**, choose **Add to dashboard**.

9. Choose **Create**.

**Old console**

**To create an alarm using the Amazon EC2 console**

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

2. In the navigation pane, choose **Instances**.

3. Select the instance.

4. On the **Monitoring** tab located at the bottom of the page, choose **Create Alarm**. Or, from the **Actions** dropdown, choose **CloudWatch Monitoring, Add/Edit Alarm**.

5. In the **Create Alarm** dialog box, do the following:

   a. Choose **create topic**. For **Send a notification to**, enter a name for the SNS topic. For **With these recipients**, enter one or more email addresses to receive notification.

   b. Specify the metric and the criteria for the policy. For example, you can leave the default settings for **Whenever** (Average of CPU Utilization). For **Is**, choose **>=** and enter **80 percent**. For **For at least**, enter **1** consecutive period of **5 Minutes**.

   c. Choose **Create Alarm**.
Create alarms that stop, terminate, reboot, or recover an instance

You can edit your CloudWatch alarm settings from the Amazon EC2 console or the CloudWatch console. If you want to delete your alarm, you can do so from the CloudWatch console. For more information, see Editing or Deleting a CloudWatch Alarm in the Amazon CloudWatch User Guide.

Create alarms that stop, terminate, reboot, or recover an instance

Using Amazon CloudWatch alarm actions, you can create alarms that automatically stop, terminate, reboot, or recover your instances. You can use the stop or terminate actions to help you save money when you no longer need an instance to be running. You can use the reboot and recover actions to automatically reboot those instances or recover them onto new hardware if a system impairment occurs.

The AWSServiceRoleForCloudWatchEvents service-linked role enables AWS to perform alarm actions on your behalf. The first time you create an alarm in the AWS Management Console, the IAM CLI, or the IAM API, CloudWatch creates the service-linked role for you.

There are a number of scenarios in which you might want to automatically stop or terminate your instance. For example, you might have instances dedicated to batch payroll processing jobs or scientific computing tasks that run for a period of time and then complete their work. Rather than letting those instances sit idle (and accrue charges), you can stop or terminate them, which can help you to save money. The main difference between using the stop and the terminate alarm actions is that you can easily start a stopped instance if you need to run it again later, and you can keep the same instance...
Create alarms that stop, terminate, reboot, or recover an 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 iam:CreateServiceLinkedRole to create or modify an alarm that performs EC2 alarm actions.

Contents

• Add stop actions to Amazon CloudWatch alarms (p. 864)
• Add terminate actions to Amazon CloudWatch alarms (p. 866)
• Add reboot actions to Amazon CloudWatch alarms (p. 867)
• Add recover actions to Amazon CloudWatch alarms (p. 868)
• Use the Amazon CloudWatch console to view alarm and action history (p. 871)
• Amazon CloudWatch alarm action scenarios (p. 871)

Add 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.

New console

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 and choose Actions, Monitor and troubleshoot, Manage CloudWatch alarms.

Alternatively, you can choose the plus sign (+) in the Alarm status column.
4. On the Manage CloudWatch alarms page, do the following:
   a. Choose Create an alarm.
Create alarms that stop, terminate, reboot, or recover an instance

b. To receive an email when the alarm is triggered, for **Alarm notification**, choose an existing Amazon SNS topic. You first need to create an Amazon SNS topic using the Amazon SNS console. For more information, see Using Amazon SNS for application-to-person (A2P) messaging in the Amazon Simple Notification Service Developer Guide.

c. Toggle on **Alarm action**, and choose **Stop**.

d. For **Group samples by** and **Type of data to sample**, choose a statistic and a metric. In this example, choose **Average** and **CPU utilization**.

e. For **Alarm When** and **Percent**, specify the metric threshold. In this example, specify <= and 10 percent.

f. For **Consecutive period** and **Period**, specify the evaluation period for the alarm. In this example, specify 1 consecutive period of **5 Minutes**.

g. Amazon CloudWatch automatically creates an alarm name for you. To change the name, for **Alarm name**, enter a new name. Alarm names must contain only ASCII characters.

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.

h. Choose **Create**.

Old console

**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.
Create alarms that stop, terminate, reboot, or recover an instance

Choose Create Alarm.

Add 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 Enable termination protection (p. 443).

New console

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 and choose Actions, Monitor and troubleshoot, Manage CloudWatch alarms.

Alternatively, you can choose the plus sign (+) in the Alarm status column.

4. On the Manage CloudWatch alarms page, do the following:
   a. Choose Create an alarm.
   b. To receive an email when the alarm is triggered, for Alarm notification, choose an existing Amazon SNS topic. You first need to create an Amazon SNS topic using the Amazon SNS console. For more information, see Using Amazon SNS for application-to-person (A2P) messaging in the Amazon Simple Notification Service Developer Guide.
   c. Toggle on Alarm action, and choose Terminate.
   d. For Group samples by and Type of data to sample, choose a statistic and a metric. In this example, choose Average and CPU utilization.
   e. For Alarm When and Percent, specify the metric threshold. In this example, specify => and 10 percent.
   f. For Consecutive period and Period, specify the evaluation period for the alarm. In this example, specify 24 consecutive periods of 1 Hour.
   g. Amazon CloudWatch automatically creates an alarm name for you. To change the name, for Alarm name, enter a new name. Alarm names must contain only ASCII characters.

   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.

   h. Choose Create.

Old console

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.
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.

---

**Add reboot actions to Amazon CloudWatch alarms**

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically reboots the instance. The reboot alarm action is recommended for Instance Health Check failures (as opposed to the recover alarm action, which is suited for System Health Check failures). An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance. When you reboot an instance, it remains on the same physical host, so your instance keeps its public DNS name, private IP address, and any data on its instance store volumes.

Rebooting an instance doesn't start a new instance billing period (with a minimum one-minute charge), unlike stopping and restarting your instance. For more information, see Reboot your instance (p. 438).

**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.

---

**New console**

**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 and choose Actions, Monitor and troubleshoot, Manage CloudWatch alarms.

Alternatively, you can choose the plus sign (+) in the Alarm status column.
4. On the Manage CloudWatch alarms page, do the following:

   a. Choose Create an alarm.
   b. To receive an email when the alarm is triggered, for Alarm notification, choose an existing Amazon SNS topic. You first need to create an Amazon SNS topic using the Amazon SNS console. For more information, see Using Amazon SNS for application-to-person (A2P) messaging in the Amazon Simple Notification Service Developer Guide.
   c. Toggle on Alarm action, and choose Reboot.
   d. For Group samples by and Type of data to sample, choose a statistic and a metric. In this example, choose Average and Status check failed: instance.
   e. For Consecutive period and Period, specify the evaluation period for the alarm. In this example, enter 3 consecutive periods of 5 Minutes.
   f. Amazon CloudWatch automatically creates an alarm name for you. To change the name, for Alarm name, enter a new name. Alarm names must contain only ASCII characters.
   g. Choose Create.

Old console

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.
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 5 Minutes.
   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.

Add recover actions to Amazon CloudWatch alarms

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance. If the instance becomes impaired due to an underlying hardware failure or a problem that requires AWS involvement to repair, you can automatically recover the instance. Terminated instances cannot be recovered. A recovered instance is identical to the original instance, including the instance ID, private IP addresses, Elastic IP addresses, and all instance metadata.

CloudWatch prevents you from adding a recovery action to an alarm that is on an instance which does not support recovery actions.
When the `StatusCheckFailed_System` alarm is triggered, and the recover action is initiated, you are notified by the Amazon SNS topic that you chose when you created the alarm and associated the recover action. During instance recovery, the instance is migrated during an instance reboot, and any data that is in-memory is lost. When the process is complete, information is published to the SNS topic you've configured for the alarm. Anyone who is subscribed to this SNS topic receives an email notification that includes the status of the recovery attempt and any further instructions. You notice an instance reboot on the recovered instance.

The recover action can be used only with `StatusCheckFailed_System`, not with `StatusCheckFailed_Instance`.

The following problems can cause system status checks to fail:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host that impact network reachability

The recover action is supported only on instances with the following characteristics:

- Use one of the following instance types: C3, C4, C5, C5a, C5n, M3, M4, M5, M5a, M5n, M5zn, M6i, P3, R3, R4, R5, R5a, R5b, R5n, T2, T3, T3a, high memory (virtualized only), X1, 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.

New console

**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 and choose **Actions, Monitor and troubleshoot, Manage CloudWatch alarms**.

   Alternatively, you can choose the plus sign (++) in the **Alarm status** column.

4. On the **Manage CloudWatch alarms** page, do the following:

   a. Choose **Create an alarm**.

   b. To receive an email when the alarm is triggered, for **Alarm notification**, choose an existing Amazon SNS topic. You first need to create an Amazon SNS topic using the Amazon SNS console. For more information, see Using Amazon SNS for application-to-person (A2P) messaging in the Amazon Simple Notification Service Developer Guide.

**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
Create alarms that stop, terminate, reboot, or recover an instance

Notifications when automatic instance recovery actions occur, even if an SNS topic is not specified or the root user is not subscribed to the specified SNS topic.

c. Toggle on Alarm action, and choose Recover.

d. For Group samples by and Type of data to sample, choose a statistic and a metric. In this example, choose Average and Status check failed: system.

e. For Consecutive period and Period, specify the evaluation period for the alarm. In this example, enter 2 consecutive periods of 5 Minutes.

f. Amazon CloudWatch automatically creates an alarm name for you. To change the name, for Alarm name, enter a new name. Alarm names must contain only ASCII characters.

g. Choose Create.

Old console

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.
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 5 Minutes.

   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.
Use 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)**

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 alarms that stop, terminate, reboot, or recover an instance

1. **Alarm notification**
   - Choose an existing topic or enter a name to create a new topic

2. **Alarm action**
   - Specify the action to take when the alarm is triggered

3. **Alarm thresholds**
   - Specify the metric thresholds for the alarm

   - Group samples by
     - Average

   - Alarm When
     - _______

   - Consecutive Period
     - _______

   - Alarm name
     - awsec2-i-04a2b95d0495ac1ee-GreaterThanOrEqualToThreshold-
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.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stop</td>
</tr>
<tr>
<td>2</td>
<td>Maximum</td>
</tr>
<tr>
<td>3</td>
<td>CPU Utilization</td>
</tr>
<tr>
<td>4</td>
<td>&lt;=</td>
</tr>
<tr>
<td>5</td>
<td>10%</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1 Hour</td>
</tr>
</tbody>
</table>
**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.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stop and email</td>
</tr>
<tr>
<td>2</td>
<td>Average</td>
</tr>
<tr>
<td>3</td>
<td>CPU Utilization</td>
</tr>
<tr>
<td>4</td>
<td>&lt;=</td>
</tr>
<tr>
<td>5</td>
<td>5%</td>
</tr>
<tr>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>1 Hour</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Email</td>
</tr>
<tr>
<td>2</td>
<td>Sum</td>
</tr>
<tr>
<td>3</td>
<td>Network Out</td>
</tr>
<tr>
<td>4</td>
<td>&gt;</td>
</tr>
<tr>
<td>5</td>
<td>10 GB</td>
</tr>
<tr>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>1 Hour</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stop and send SMS</td>
</tr>
<tr>
<td>2</td>
<td>Sum</td>
</tr>
<tr>
<td>3</td>
<td>Network Out</td>
</tr>
<tr>
<td>4</td>
<td>&gt;</td>
</tr>
<tr>
<td>5</td>
<td>1 GB</td>
</tr>
</tbody>
</table>
Automate Amazon EC2 with EventBridge

Amazon EventBridge 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 EventBridge 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:

### Scenario 5: Stop an impaired instance

Create an alarm that stops an instance that fails three consecutive status checks (performed at 5-minute intervals).

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stop</td>
</tr>
<tr>
<td>2</td>
<td>Average</td>
</tr>
<tr>
<td>3</td>
<td>Status Check Failed: System</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>15 Minutes</td>
</tr>
</tbody>
</table>

### Scenario 6: 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.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Terminate</td>
</tr>
<tr>
<td>2</td>
<td>Maximum</td>
</tr>
<tr>
<td>3</td>
<td>Network Out</td>
</tr>
<tr>
<td>4</td>
<td>&lt;=</td>
</tr>
<tr>
<td>5</td>
<td>100,000 bytes</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>5 Minutes</td>
</tr>
</tbody>
</table>

---

875
Log API calls with AWS CloudTrail

- Invoking an AWS Lambda function
- Invoking Amazon EC2 Run Command
- Relaying the event to Amazon Kinesis Data Streams
- Activating an AWS Step Functions state machine
- Notifying an Amazon SNS topic or an Amazon SQS queue

Some examples of using EventBridge 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 EventBridge User Guide.

Log Amazon EC2 and Amazon EBS API calls with AWS CloudTrail

Amazon EC2 and Amazon EBS are integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon EC2 and Amazon EBS. CloudTrail captures all API calls for Amazon EC2 and Amazon EBS as events, including calls from the console and from code calls to the APIs. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon EC2 and Amazon EBS. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in Event history. Using the information collected by CloudTrail, you can determine the request that was made to Amazon EC2 and Amazon EBS, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Amazon EC2 and Amazon EBS information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon EC2 and Amazon EBS, that activity is recorded in a CloudTrail event along with other AWS service events in Event history. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for Amazon EC2 and Amazon EBS, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts
All Amazon EC2 actions, and Amazon EBS management actions, are logged by CloudTrail and are documented in the Amazon EC2 API Reference. For example, calls to the RunInstances, DescribeInstances, or CreateImage actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element.

**Understand 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.

```json
{
    "Records": [  
        {
            "eventVersion": "1.03",
            "userIdentity": {
                "type": "Root",
                "principalId": "123456789012",
                "arn": "arn:aws:iam::123456789012:root",
                "accountId": "123456789012",
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
                "userName": "user"
            },
            "eventTime": "2016-05-20T08:27:45Z",
            "eventSource": "ec2.amazonaws.com",
            "eventName": "TerminateInstances",
            "awsRegion": "us-west-2",
            "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,
```
Use AWS CloudTrail to audit users that connect via EC2 Instance Connect

Use AWS CloudTrail to audit the users that connect to your instances via EC2 Instance Connect.

To audit SSH activity via EC2 Instance Connect using the AWS CloudTrail console

1. Open the AWS CloudTrail console at https://console.aws.amazon.com/cloudtrail/.
2. Verify that you are in the correct Region.
3. In the navigation pane, choose Event history.
5. (Optional) For Time range, select a time range.
6. Choose the Refresh events icon.
7. The page displays the events that correspond to the SendSSHPublicKey API calls. Expand an event using the arrow to view additional details, such as the user name and AWS access key that was used to make the SSH connection, and the source IP address.
8. To display the full event information in JSON format, choose View event. The requestParameters field contains the destination instance ID, OS user name, and public key that were used to make the SSH connection.

```json
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDEFGONGNOMOOCB6XYTQEXAMPLE",
        "arn": "arn:aws:iam::1234567890120:user/IAM-friendly-name",
        "accountId": "123456789012",
        "accessKeyId": "ABCDEFGUKZHNAW4OSN2AEXAMPLE",
        "userName": "IAM-friendly-name",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-09-21T21:37:58Z"
            }
        }
    },
    "eventTime": "2018-09-21T21:38:00Z",
    "eventSource": "ec2-instance-connect.amazonaws.com",
    "eventName": "SendSSHPublicKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "123.456.789.012",
    "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
    "requestParameters": {
        "instanceId": "i-0123456789EXAMPLE",
        "osUser": "ec2-user",
        "SSHKey": {
            "name": "running"
        }
    }
}
```
Monitor your .NET and SQL Server applications with CloudWatch Application Insights

CloudWatch Application Insights helps you monitor your .NET and SQL Server applications that use Amazon EC2 instances along with other AWS application resources. It identifies and sets up key metrics logs, and alarms across your application resources and technology stack (for example, your Microsoft SQL Server database, web (IIS) and application servers, OS, load balancers, and queues). It continuously monitors the metrics and logs to detect and correlate anomalies and errors. When errors and anomalies are detected, Application Insights generates CloudWatch Events that you can use to set up notifications or take actions. To aid with troubleshooting, it creates automated dashboards for the detected problems, which include correlated metric anomalies and log errors, along with additional insights to point you to the potential root cause. The automated dashboards help you to take swift remedial actions to keep your applications healthy and to prevent impact to the end users of your application.

To view a complete list of supported logs and metrics, see Logs and Metrics Supported by Amazon CloudWatch Application Insights.

Information provided about detected problems:

- A short summary of the problem
- The start time and date of the problem
- The problem severity: High/Medium/Low
- The status of the detected problem: In-progress/Resolved
- Insights: Automatically generated insights on the detected problem and possible root cause
- Feedback on insights: Feedback you have provided about the usefulness of the insights generated by CloudWatch Application Insights for .NET and SQL Server
- Related observations: A detailed view of the metric anomalies and error snippets of relevant logs related to the problem across various application components

Feedback

You can provide feedback on automatically generated insights on detected problems by designating them useful or not useful. Your feedback on the insights, along with your application diagnostics (metric anomalies and log exceptions), are used to improve the future detection of similar problems.

For more information, see the CloudWatch Application Insights documentation in the Amazon CloudWatch User Guide.
Networking in Amazon EC2

Amazon VPC enables you to launch AWS resources, such as Amazon EC2 instances, into a virtual network dedicated to your AWS account, known as a virtual private cloud (VPC). When you launch an instance, you can select a subnet from the VPC. The instance is configured with a primary network interface, which is a logical virtual network card. The instance receives a primary private IP address from the IPv4 address of the subnet, and it is assigned to the primary network interface.

You can control whether the instance receives a public IP address from Amazon's pool of public IP addresses. The public IP address of an instance is associated with your instance only until it is stopped or terminated. If you require a persistent public IP address, you can allocate an Elastic IP address for your AWS account and associate it with an instance or a network interface. An Elastic IP address remains associated with your AWS account until you release it, and you can move it from one instance to another as needed. You can bring your own IP address range to your AWS account, where it appears as an address pool, and then allocate Elastic IP addresses from your address pool.

To increase network performance and reduce latency, you can launch instances in a placement group. You can get significantly higher packet per second (PPS) performance using enhanced networking. You can accelerate high performance computing and machine learning applications using an Elastic Fabric Adapter (EFA), which is a network device that you can attach to a supported instance type.

Features
- Regions and Zones (p. 880)
- Amazon EC2 instance IP addressing (p. 893)
- Bring your own IP addresses (BYOIP) in Amazon EC2 (p. 908)
- Assigning prefixes to Amazon EC2 network interfaces (p. 917)
- Elastic IP addresses (p. 926)
- Elastic network interfaces (p. 934)
- Amazon EC2 instance network bandwidth (p. 959)
- Enhanced networking on Windows (p. 960)
- Placement groups (p. 975)
- Network maximum transmission unit (MTU) for your EC2 instance (p. 987)
- Virtual private clouds (p. 991)
- Ports and Protocols for Windows Amazon Machine Images (AMIs) (p. 992)
- EC2-Classic (p. 1022)

Regions and Zones

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

- Availability Zones are multiple, isolated locations within each Region.
- Local Zones provide you the ability to place resources, such as compute and storage, in multiple locations closer to your end users.
- AWS Outposts brings native AWS services, infrastructure, and operating models to virtually any data center, co-location space, or on-premises facility.
- Wavelength Zones allow developers to build applications that deliver ultra-low latencies to 5G devices and end users. Wavelength deploys standard AWS compute and storage services to the edge of telecommunication carriers' 5G networks.
AWS operates state-of-the-art, highly available data centers. Although rare, failures can occur that affect the availability of instances that are in the same location. If you host all of your instances in a single location that is affected by a failure, none of your instances would be available.

To help you determine which deployment is best for you, see AWS Wavelength FAQs.

Contents
- Regions (p. 881)
- Availability Zones (p. 885)
- Local Zones (p. 887)
- Wavelength Zones (p. 890)
- AWS Outposts (p. 892)

Regions

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

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

When you launch an instance, you must select an AMI that's in the same Region. If the AMI is in another Region, you can copy the AMI to the Region you're using. For more information, see Copy an AMI (p. 113).

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

Contents
- Available Regions (p. 881)
- Regions and endpoints (p. 882)
- Describe your Regions (p. 883)
- Get the Region name (p. 884)
- Specify the Region for a resource (p. 884)

Available Regions

Your account determines the Regions that are available to you.

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

The following table lists the Regions provided by an AWS account. You can't describe or access additional Regions from an AWS account, such as AWS GovCloud (US) Region or the China Regions. To use a Region introduced after March 20, 2019, you must enable the Region. For more information, see Managing AWS Regions in the AWS General Reference.
For information about available Wavelength Zones, see Available Wavelength Zones in the AWS Wavelength Developer Guide. For information about available Local Zones, see the section called “Available Local Zones” (p. 888).

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Opt-in Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>us-east-2</td>
<td>US East (Ohio)</td>
<td>Not required</td>
</tr>
<tr>
<td>us-east-1</td>
<td>US East (N. Virginia)</td>
<td>Not required</td>
</tr>
<tr>
<td>us-west-1</td>
<td>US West (N. California)</td>
<td>Not required</td>
</tr>
<tr>
<td>us-west-2</td>
<td>US West (Oregon)</td>
<td>Not required</td>
</tr>
<tr>
<td>af-south-1</td>
<td>Africa (Cape Town)</td>
<td>Required</td>
</tr>
<tr>
<td>ap-east-1</td>
<td>Asia Pacific (Hong Kong)</td>
<td>Required</td>
</tr>
<tr>
<td>ap-south-1</td>
<td>Asia Pacific (Mumbai)</td>
<td>Not required</td>
</tr>
<tr>
<td>ap-northeast-3</td>
<td>Asia Pacific (Osaka)</td>
<td>Not required</td>
</tr>
<tr>
<td>ap-northeast-2</td>
<td>Asia Pacific (Seoul)</td>
<td>Not required</td>
</tr>
<tr>
<td>ap-southeast-1</td>
<td>Asia Pacific (Singapore)</td>
<td>Not required</td>
</tr>
<tr>
<td>ap-southeast-2</td>
<td>Asia Pacific (Sydney)</td>
<td>Not required</td>
</tr>
<tr>
<td>ap-northeast-1</td>
<td>Asia Pacific (Tokyo)</td>
<td>Not required</td>
</tr>
<tr>
<td>ca-central-1</td>
<td>Canada (Central)</td>
<td>Not required</td>
</tr>
<tr>
<td>eu-central-1</td>
<td>Europe (Frankfurt)</td>
<td>Not required</td>
</tr>
<tr>
<td>eu-west-1</td>
<td>Europe (Ireland)</td>
<td>Not required</td>
</tr>
<tr>
<td>eu-west-2</td>
<td>Europe (London)</td>
<td>Not required</td>
</tr>
<tr>
<td>eu-south-1</td>
<td>Europe (Milan)</td>
<td>Required</td>
</tr>
<tr>
<td>eu-west-3</td>
<td>Europe (Paris)</td>
<td>Not required</td>
</tr>
<tr>
<td>eu-north-1</td>
<td>Europe (Stockholm)</td>
<td>Not required</td>
</tr>
<tr>
<td>me-south-1</td>
<td>Middle East (Bahrain)</td>
<td>Required</td>
</tr>
<tr>
<td>sa-east-1</td>
<td>South America (São Paulo)</td>
<td>Not required</td>
</tr>
</tbody>
</table>

For more information, see AWS Global Infrastructure.

The number and mapping of Availability Zones per Region may vary between AWS accounts. To get a list of the Availability Zones that are available to your account, you can use the Amazon EC2 console or the command line interface. For more information, see Describe your Regions (p. 883).

## Regions and endpoints

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

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

To find your Regions using the console

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

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

To find your Regions using the AWS CLI

- Use the describe-regions command as follows to describe the Regions that are enabled for your account.

  
  ```
  aws ec2 describe-regions
  ```

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

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

To find your Regions using the AWS Tools for Windows PowerShell

- Use the Get-EC2Region command as follows to describe the Regions for your account.
**Get the Region name**

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

**To view the Region name using the AWS CLI**

- Use the `get-regions` command as follows to describe the name of the specified Region.

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

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

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

  The following is the output:

  ```
  Ohio
  ```

**Specify the Region for a resource**

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

**Considerations**

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

**To specify the Region for a resource using the console**

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

**To specify the default Region using the command line**

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

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

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

For more information about the endpoints for Amazon EC2, see Amazon Elastic Compute Cloud Endpoints.
## Availability Zones

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

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

![Diagram of Availability Zones](image)

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

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

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

You can view AZ IDs to determine the location of resources in one account relative to the resources in another account. For example, if you share a subnet in the Availability Zone with the AZ ID `use-a2` with another account, this subnet is available to that account in the Availability Zone whose AZ ID is also `use-a2`. The AZ ID for each VPC and subnet is displayed in the Amazon VPC console. For more information, see [Working with Shared VPCs](#) in the *Amazon VPC User Guide*.

As Availability Zones grow over time, our ability to expand them can become constrained. If this happens, we might restrict you from launching an instance in a constrained Availability Zone unless you already have an instance in that Availability Zone. Eventually, we might also remove the constrained Availability Zone from the list of Availability Zones for new accounts. Therefore, your account might have a different number of available Availability Zones in a Region than another account.

### Contents

- Describe your Availability Zones (p. 885)
- Launch instances in an Availability Zone (p. 886)
- Migrate an instance to another Availability Zone (p. 886)

### Describe your Availability Zones

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

#### To find your Availability Zones using the console

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

**To find your Availability Zones using the AWS CLI**

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

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

2. Use the `describe-availability-zones` command as follows to describe the Availability Zones regardless of the opt-in status.

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

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

Use the `Get-EC2AvailabilityZone` command as follows to describe the Availability Zones within the specified Region.

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

**Launch instances in an Availability Zone**

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

When you launch an instance, you can optionally specify an Availability Zone in the Region that you are using. If you do not specify an Availability Zone, we select an Availability Zone for you. When you launch your initial instances, we recommend that you accept the default Availability Zone, because this allows us to select the best Availability Zone for you based on system health and available capacity. If you launch additional instances, specify a Zone only if your new instances must be close to, or separated from, your running instances.

**Migrate an instance to another Availability Zone**

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

The migration process involves:

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

**To migrate an instance to another Availability Zone**

1. Create an AMI from the instance. The procedure depends on your operating system and the type of root device volume for the instance. For more information, see the documentation that corresponds to your operating system and root device volume:
2. If you need to preserve the private IPv4 address of the instance, you must delete the subnet in the current Availability Zone and then create a subnet in the new Availability Zone with the same IPv4 address range as the original subnet. Note that you must terminate all instances in a subnet before you can delete it. Therefore, you should create AMIs from all of the instances in your subnet so that you can move all instances from the current subnet to the new subnet.

3. Launch an instance from the AMI that you just created, specifying the new Availability Zone or subnet. You can use the same instance type as the original instance, or select a new instance type. For more information, see Launch instances in an Availability Zone (p. 886).

4. If the original instance has an associated Elastic IP address, associate it with the new instance. For more information, see Disassociate an Elastic IP address (p. 931).

5. If the original instance is a Reserved Instance, change the Availability Zone for your reservation. (If you also changed the instance type, you can also change the instance type for your reservation.) For more information, see Submit modification requests (p. 275).

6. (Optional) Terminate the original instance. For more information, see Terminate an instance (p. 443).

Local Zones

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

A Local Zone is represented by a Region code followed by an identifier that indicates the location, for example, us-west-2-lax-1a. For more information, see Available Local Zones (p. 888).

To use a Local Zone, you must first enable it. For more information, see the section called “Opt in to Local Zones” (p. 889). Next, create a subnet in the Local Zone. Finally, launch any of the following resources in the Local Zone subnet, so that your applications are close to your end users:

- Amazon EC2 instances
- Amazon EBS volumes
- Amazon ECS
- Amazon EKS
- Internet gateways

In addition to the preceding list, the following resources are available in the Los Angeles Local Zones:
Available Local Zones

The following table lists the available Local Zones. For information about how to opt in, see the section called “Opt in to Local Zones” (p. 889).

<table>
<thead>
<tr>
<th>Location</th>
<th>Zone Name</th>
<th>Parent Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boston</td>
<td>us-east-1-bos-1a</td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td>Dallas</td>
<td>us-east-1-dfw-1a</td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td>Denver</td>
<td>us-west-2-den-1a</td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td>Houston</td>
<td>us-east-1-iah-1a</td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>us-west-2-lax-1a</td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>us-west-2-lax-1b</td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td>Miami</td>
<td>us-east-1-mia-1a</td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td>Philadelphia</td>
<td>us-east-1-phl-1a</td>
<td>US East (N. Virginia)</td>
</tr>
</tbody>
</table>

Describe your Local Zones

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

To find your Local Zones using the console

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

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

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

2. Use the `describe-availability-zones` command as follows to describe the Local Zones regardless of whether they are enabled.

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

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

Use the `Get-EC2AvailabilityZone` command as follows to describe the Local Zones in the specified Region.

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

Opt in to Local Zones

Before you can specify a Local Zone for a resource or service, you must opt in to Local Zones.

Consideration

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

To opt in to Local Zones using the console

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

To opt in to Local Zones using the AWS CLI

- Use the `modify-availability-zone-group` command.

Launch instances in a Local Zone

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

You can allocate the following IP addresses from a network border group:
Amazon Elastic Compute Cloud
User Guide for Windows Instances
Wavelength Zones

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

To launch an instance in a Local Zone:

1. Enable Local Zones. For more information, see Opt in to Local Zones (p. 889).
2. Create a VPC in a Region that supports the Local Zone. For more information, see Creating a VPC in the Amazon VPC User Guide.
3. Create a subnet. Select the Local Zone when you create the subnet. For more information, see Creating a subnet in your VPC in the Amazon VPC User Guide.
4. Launch an instance, and select the subnet that you created in the Local Zone. For more information, see Launch your instance (p. 390).

Wavelength Zones

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

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

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

To use a Wavelength Zone, you must first opt in to the Zone. For more information, see the section called "Enable Wavelength Zones" (p. 891). Next, create a subnet in the Wavelength Zone. Finally, launch your resources in the Wavelength Zones subnet, so that your applications are closer to your end users.

Wavelength Zones are not available in every Region. For information about the Regions that support Wavelength Zones, see Available Wavelength Zones in the AWS Wavelength Developer Guide.

Contents

- Describe your Wavelength Zones (p. 891)
- Enable Wavelength Zones (p. 891)
- Launch instances in a Wavelength Zone (p. 892)
Describe your Wavelength Zones

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

To find your Wavelength Zones using the console

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

To find your Wavelength Zones using the AWS CLI

1. Use the describe-availability-zones command as follows to describe the Wavelength Zones within the specified Region.

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

2. Use the describe-availability-zones command as follows to describe the Wavelength Zones regardless of the opt-in status.

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

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

Use the Get-EC2AvailabilityZone command as follows to describe the Wavelength Zone within the specified Region.

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

Enable Wavelength Zones

Before you specify a Wavelength Zone for a resource or service, you must opt in to Wavelength Zones.

Considerations

- Some AWS resources are not available in all Regions. Make sure that you can create the resources that you need in the desired Region or Wavelength Zone before launching an instance in a specific Wavelength Zone.

To opt in to Wavelength Zone using the console

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

**To enable Wavelength Zones using the AWS CLI**

Use the `modify-availability-zone-group` command.

**Launch instances in a Wavelength Zone**

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


**AWS Outposts**

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

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

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

**Launch instances on an Outpost**

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

The root volume must be 30 GB or smaller. You can specify data volumes in the block device mapping of the AMI or the instance to provide additional storage. To trim unused blocks from the boot volume, see [How to Build Sparse EBS Volumes](https://aws.amazon.com/blogs/partner/how-to-build-sparse-ebs-volumes/) in the AWS Partner Network Blog.

We recommend that you increase the NVMe timeout for the root volume. For more information, see [I/O operation timeout](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-ec2-with-containers.html#io-timeout) (p. 1344).


**Create a volume on an Outpost**

You can create EBS volumes in the Outpost subnet that you created. When you create the volume, specify the Amazon Resource Name (ARN) of the Outpost.
The following `create-volume` command creates an empty 50 GB volume on the specified Outpost.

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

You can dynamically modify the size of your Amazon EBS gp2 volumes without detaching them. For more information about modifying a volume without detaching it, see Request modifications to your EBS volumes (p. 1317).

---

**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. 893)
- Public IPv4 addresses and external DNS hostnames (p. 894)
- Elastic IP addresses (IPv4) (p. 895)
- Amazon DNS server (p. 895)
- IPv6 addresses (p. 895)
- Work with the IPv4 addresses for your instances (p. 896)
- Work with the IPv6 addresses for your instances (p. 899)
- Multiple IP addresses (p. 901)

**Private IPv4 addresses and internal DNS hostnames**

A private IPv4 address is an IP address that's not reachable over the Internet. You can use private IPv4 addresses for communication between instances in the same VPC. For more information about the standards and specifications of private IPv4 addresses, see RFC 1918. We allocate private IPv4 addresses to instances using DHCP.

**Note**

You can create a VPC with a publicly routable CIDR block that falls outside of the private IPv4 address ranges specified in RFC 1918. However, for the purposes of this documentation, we refer to private IPv4 addresses (or 'private IP addresses') as the IP addresses that are within the IPv4 CIDR range of your VPC.

When you launch an instance, we allocate a primary private IPv4 address for the instance. Each instance is also given an internal DNS hostname that resolves to the primary private IPv4 address; for example, `ip-10-251-50-12.ec2.internal`. You can use the internal DNS hostname for communication between instances in the same VPC, but we can't resolve the internal DNS hostname outside of the VPC.

An instance receives a primary private IP address from the IPv4 address range of the subnet. For more information, see VPC and subnet sizing in the Amazon VPC User Guide. If you don't specify a primary private IP address when you launch the instance, we select an available IP address in the subnet's IPv4 range for you. Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. You can also specify additional private IPv4 addresses, known as secondary private IPv4
addresses. Unlike primary private IP addresses, secondary private IP addresses can be reassigned from one instance to another. For more information, see Multiple IP addresses (p. 901).

A private IPv4 address, regardless of whether it is a primary or secondary address, remains associated with the network interface when the instance is stopped and started, or hibernated and started, and is released when the instance is terminated.

Public IPv4 addresses and external DNS hostnames

A public IP address is an IPv4 address that's reachable from the Internet. You can use public addresses for communication between your instances and the Internet.

Each instance that receives a public IP address is also given an external DNS hostname; for example, ec2-203-0-113-25.compute-1.amazonaws.com. We resolve an external DNS hostname to the public IP address of the instance from outside its VPC, and to the private IPv4 address of the instance from inside its VPC. The public IP address is mapped to the primary private IP address through network address translation (NAT). For more information, see RFC 1631: The IP Network Address Translator (NAT).

When you launch an instance in a default VPC, we assign it a public IP address by default. When you launch an instance into a nondefault VPC, the subnet has an attribute that determines whether instances launched into that subnet receive a public IP address from the public IPv4 address pool. By default, we don't assign a public IP address to instances launched in a nondefault subnet.

You can control whether your instance receives a public IP address as follows:

- Modifying the public IP addressing attribute of your subnet. For more information, see Modifying the public IPv4 addressing attribute for your subnet in the Amazon VPC User Guide.
- Enabling or disabling the public IP addressing feature during launch, which overrides the subnet's public IP addressing attribute. For more information, see Assign a public IPv4 address during instance launch (p. 898).

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 (IPv4) address from your instance. Instead, in certain cases, we release the public IP address from your instance, or assign it a new one:

- We release your instance's public IP address when it is stopped, hibernated, or terminated. Your stopped or hibernated instance receives a new public IP address when it is started.
- We release your instance's public IP address when you associate an Elastic IP address with it. When you disassociate the Elastic IP address from your instance, it receives a new public IP address.
- If the public IP address of your instance in a VPC has been released, it will not receive a new one if there is more than one network interface attached to your instance.
- If your instance's public IP address is released while it has a secondary private IP address that is associated with an Elastic IP address, the instance does not receive a new public IP address.

If you require a persistent public IP address that can be associated to and from instances as you require, use an Elastic IP address instead.

If you use dynamic DNS to map an existing DNS name to a new instance's public IP address, it might take up to 24 hours for the IP address to propagate through the Internet. As a result, new instances might not receive traffic while terminated instances continue to receive requests. To solve this problem, use an Elastic IP address. You can allocate your own Elastic IP address, and associate it with your instance. For more information, see Elastic IP addresses (p. 926).
If you assign an Elastic IP address to an instance, it receives an IPv4 DNS hostname if DNS hostnames are enabled. For more information, see Using DNS with your VPC in the Amazon VPC User Guide.

**Note**
Instances that access other instances through their public NAT IP address are charged for regional or Internet data transfer, depending on whether the instances are in the same Region.

**Elastic IP addresses (IPv4)**

An Elastic IP address is a public IPv4 address that you can allocate to your account. You can associate it to and disassociate it from instances as you require. It's allocated to your account until you choose to release it. For more information about Elastic IP addresses and how to use them, see Elastic IP addresses (p. 926).

We do not support Elastic IP addresses for IPv6.

**Amazon DNS server**

Amazon provides a DNS server that resolves Amazon-provided IPv4 DNS hostnames to IPv4 addresses. The Amazon DNS server is located at the base of your VPC network range plus two. For more information, see Amazon DNS server in the Amazon VPC User Guide.

**IPv6 addresses**

You can optionally associate an IPv6 CIDR block with your VPC, and associate IPv6 CIDR blocks with your subnets. The IPv6 CIDR block for your VPC is automatically assigned from Amazon's pool of IPv6 addresses; you cannot choose the range yourself. For more information, see the following topics in the Amazon VPC User Guide:

- VPC and subnet sizing for IPv6
- Associating an IPv6 CIDR block with your VPC
- Associating an IPv6 CIDR block with your subnet

IPv6 addresses are globally unique, and therefore reachable over the Internet. Your instance receives an IPv6 address if an IPv6 CIDR block is associated with your VPC and subnet, and if one of the following is true:

- Your subnet is configured to automatically assign an IPv6 address to an instance during launch. For more information, see Modifying the IPv6 addressing attribute for your subnet.
- You assign an IPv6 address to your instance during launch.
- You assign an IPv6 address to the primary network interface of your instance after launch.
- You assign an IPv6 address to a network interface in the same subnet, and attach the network interface to your instance after launch.

When your instance receives an IPv6 address during launch, the address is associated with the primary network interface (eth0) of the instance. You can disassociate the IPv6 address from the network interface. We do not support IPv6 DNS hostnames for your instance.

An IPv6 address persists when you stop and start, or hibernate and start, your instance, and is released when you terminate your instance. You cannot reassign an IPv6 address while it's assigned to another network interface—you must first unassign it.

You can assign additional IPv6 addresses to your instance by assigning them to a network interface attached to your instance. The number of IPv6 addresses you can assign to a network interface and
the number of network interfaces you can attach to an instance varies per instance type. For more information, see IP addresses per network interface per instance type (p. 936).

Work with the IPv4 addresses for your instances

You can assign a public IPv4 address to your instance when you launch it. You can view the IPv4 addresses for your instance in the console through either the Instances page or the Network Interfaces page.

Contents

- View the IPv4 addresses (p. 896)
- Assign a public IPv4 address during instance launch (p. 898)

View the IPv4 addresses

You can use the Amazon EC2 console to view the private IPv4 addresses, public IPv4 addresses, and Elastic IP addresses of your instances. You can also determine the public IPv4 and private IPv4 addresses of your instance from within your instance by using instance metadata. For more information, see Instance metadata and user data (p. 579).

The public IPv4 address is displayed as a property of the network interface in the console, but it's mapped to the primary private IPv4 address through NAT. Therefore, if you inspect the properties of your network interface on your instance, for example, through `ifconfig` (Linux) or `ipconfig` (Windows), the public IPv4 address is not displayed. To determine your instance's public IPv4 address from an instance, use instance metadata.

New console

To view the IPv4 addresses for an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances and select your instance.
3. The following information is available on the Networking tab:
   - **Public IPv4 address** — The public IPv4 address. If you associated an Elastic IP address with the instance or the primary network interface, this is the Elastic IP address.
   - **Public IPv4 DNS** — The external DNS hostname.
   - **Private IPv4 addresses** — The private IPv4 address.
   - **Private IPv4 DNS** — The internal DNS hostname.
   - **Secondary private IPv4 addresses** — Any secondary private IPv4 addresses.
   - **Elastic IP addresses** — Any associated Elastic IP addresses.
4. Alternatively, under Network interfaces on the Networking tab, choose the interface ID for the primary network interface (for example, eni-123abc456def78901). The following information is available:
   - **Private DNS (IPv4)** — The internal DNS hostname.
   - **Primary private IPv4 IP** — The primary private IPv4 address.
   - **Secondary private IPv4 IPs** — Any secondary private IPv4 addresses.
   - **Public DNS** — The external DNS hostname.
   - **IPv4 Public IP** — The public IPv4 address. If you associated an Elastic IP address with the instance or the primary network interface, this is the Elastic IP address.
   - **Elastic IPs** — Any associated Elastic IP addresses.
Old console

To view the IPv4 addresses for an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances and select your instance.
3. The following information is available on the Description tab:
   - **Private DNS** — The internal DNS hostname.
   - **Private IPs** — The private IPv4 address.
   - **Secondary private IPs** — Any secondary private IPv4 addresses.
   - **Public DNS** — The external DNS hostname.
   - **IPv4 Public IP** — The public IPv4 address. If you associated an Elastic IP address with the instance or the primary network interface, this is the Elastic IP address.
   - **Elastic IPs** — Any associated Elastic IP addresses.
4. Alternatively, you can view the IPv4 addresses for the instance using the primary network interface. Under Network interfaces on the Description tab, choose eth0, and then choose the interface ID (for example, eni-123abc456def78901). The following information is available:
   - **Private DNS (IPv4)** — The internal DNS hostname.
   - **Primary private IPv4 IP** — The primary private IPv4 address.
   - **Secondary private IPv4 IPs** — Any secondary private IPv4 addresses.
   - **Public DNS** — The external DNS hostname.
   - **IPv4 Public IP** — The public IPv4 address. If you associated an Elastic IP address with the instance or the primary network interface, this is the Elastic IP address.
   - **Elastic IPs** — Any associated Elastic IP addresses.

To view the IPv4 addresses for an instance using the command line

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

- `describe-instances` (AWS CLI)

To determine your instance's IPv4 addresses using instance metadata

1. Connect to your instance. For more information, see Connect to your Windows instance (p. 413).
2. Use the following command to access the private IP address:

   ```powershell
   ```

3. Use the following command to access the public IP address:

   ```powershell
   ```

   If an Elastic IP address is associated with the instance, the value returned is that of the Elastic IP address.
Assign a public IPv4 address during instance launch

Each subnet has an attribute that determines whether instances launched into that subnet are assigned a public IP address. By default, nondefault subnets have this attribute set to false, and default subnets have this attribute set to true. When you launch an instance, a public IPv4 addressing feature is also available for you to control whether your instance is assigned a public IPv4 address; you can override the default behavior of the subnet's IP addressing attribute. The public IPv4 address is assigned from Amazon's pool of public IPv4 addresses, and is assigned to the network interface with the device index of eth0. This feature depends on certain conditions at the time you launch your instance.

Considerations

- You can't manually disassociate the public IP address from your instance after launch. Instead, it's automatically released in certain cases, after which you cannot reuse it. For more information, see Public IPv4 addresses and external DNS hostnames (p. 894). 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. 926).
- You cannot auto-assign a public IP address if you specify more than one network interface. Additionally, you cannot override the subnet setting using the auto-assign public IP feature if you specify an existing network interface for eth0.
- The public IP addressing feature is only available during launch. However, whether you assign a public IP address to your instance during launch or not, you can associate an Elastic IP address with your instance after it's launched. For more information, see Elastic IP addresses (p. 926). You can also modify your subnet's public IPv4 addressing behavior. For more information, see Modifying the public IPv4 addressing attribute for your subnet.

To enable or disable the public IP addressing feature using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose Launch Instance.
3. Select an AMI and an instance type, and then choose Next: Configure Instance Details.
4. On the Configure Instance Details page, for Network, select a VPC. The Auto-assign Public IP list is displayed. Choose Enable or Disable to override the default setting for the subnet.
5. Follow the steps on the next pages of the wizard to complete your instance's setup. For more information about the wizard configuration options, see Launch an instance using the Launch Instance Wizard (p. 392). On the final Review Instance Launch page, review your settings, and then choose Launch to choose a key pair and launch your instance.
6. On the Instances page, select your new instance and view its public IP address in IPv4 Public IP field in the details pane.

To enable or disable the public IP addressing feature using the command line

You can use one of the following commands. For more information about these command line interfaces, see Access 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)
Work with the IPv6 addresses for your instances

You can view the IPv6 addresses assigned to your instance, assign a public IPv6 address to your instance, or unassign an IPv6 address from your instance. You can view these addresses in the console through either the Instances page or the Network Interfaces page.

Contents

- View the IPv6 addresses (p. 899)
- Assign an IPv6 address to an instance (p. 900)
- Unassign an IPv6 address from an instance (p. 900)

View the IPv6 addresses

You can use the Amazon EC2 console, AWS CLI, and instance metadata to view the IPv6 addresses for your instances.

New console

To view the IPv6 addresses for an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance.
5. Alternatively, under Network interfaces on the Networking tab, choose the interface ID for the network interface (for example, eni-123abc456def78901). Locate IPv6 IPs.

Old console

To view the IPv6 addresses for an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance.
4. On the Networking tab, locate IPv6 IPs.
5. Alternatively, under Network interfaces on the Description tab, choose eth0, and then choose the interface ID (for example, eni-123abc456def78901). Locate IPv6 IPs.

To view the IPv6 addresses for an instance using the command line

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

- describe-instances (AWS CLI)

To view the IPv6 addresses for an instance using instance metadata

1. Connect to your instance. For more information, see Connect to your Windows instance (p. 413).
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/).

```
```

### Assign an IPv6 address to an instance

If your VPC and subnet have IPv6 CIDR blocks associated with them, you can assign an IPv6 address to your instance during or after launch. The IPv6 address is assigned from the IPv6 address range of the subnet, and is assigned to the network interface with the device index of eth0.

#### To assign an IPv6 address to an instance during launch

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Select an AMI and an instance type that supports IPv6, and choose **Next: Configure Instance Details**.
3. On the **Configure Instance Details** page, for **Network**, select a VPC and for **Subnet**, select a subnet. For **Auto-assign IPv6 IP**, choose **Enable**.
4. Follow the remaining steps in the wizard to launch your instance.

#### To assign an IPv6 address to an instance after launch

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select your instance, and choose **Actions**, **Networking**, **Manage IP addresses**.
4. Expand the network interface. Under **IPv6 addresses**, choose **Assign new IP address**. Enter an IPv6 address from the range of the subnet or leave the field blank to let Amazon choose an IPv6 address for you.
5. Choose **Save**.

#### To assign an IPv6 address using the command line

You can use one of the following commands. For more information about these command line interfaces, see **Access 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)

### Unassign an IPv6 address from an instance

You can unassign an IPv6 address from an instance at any time.

#### To unassign an IPv6 address from an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select your instance, and choose **Actions**, **Networking**, **Manage IP addresses**.
4. Expand the network interface. Under **IPv6 addresses**, choose **Unassign** next to the IPv6 address.
5. Choose **Save**.

### To unassign an IPv6 address from an instance using the command line

You can use one of the following commands. For more information about these command line interfaces, see **Access Amazon EC2** (p. 3).

- `unassign-ipv6-addresses` (AWS CLI)

### 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. 936).

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. 901)
- Work with multiple IPv4 addresses (p. 902)
- Work with multiple IPv6 addresses (p. 905)

### How multiple IP addresses work

The following list explains how multiple IP addresses work with network interfaces:

- You can assign a secondary private IPv4 address to any network interface. The network interface need not be attached to the instance.
- You can assign multiple IPv6 addresses to a network interface that's in a subnet that has an associated IPv6 CIDR block.
- You must choose a secondary IPv4 address from the IPv4 CIDR block range of the subnet for the network interface.
- You must choose IPv6 addresses from the IPv6 CIDR block range of the subnet for the network interface.
- You associate security groups with network interfaces, not individual IP addresses. Therefore, each IP address you specify in a network interface is subject to the security group of its network interface.
- Multiple IP addresses can be assigned and unassigned to network interfaces attached to running or stopped instances.
• Secondary private IPv4 addresses that are assigned to a network interface can be reassigned to another one if you explicitly allow it.
• An IPv6 address cannot be reassigned to another network interface; you must first unassign the IPv6 address from the existing network interface.
• When assigning multiple IP addresses to a network interface using the command line tools or API, the entire operation fails if one of the IP addresses can't be assigned.
• Primary private IPv4 addresses, secondary private IPv4 addresses, Elastic IP addresses, and IPv6 addresses remain with a secondary network interface when it is detached from an instance or attached to an instance.
• Although you can't detach the primary network interface from an instance, you can reassign the secondary private IPv4 address of the primary network interface to another network interface.

The following list explains how multiple IP addresses work with Elastic IP addresses (IPv4 only):
• Each private IPv4 address can be associated with a single Elastic IP address, and vice versa.
• When a secondary private IPv4 address is reassigned to another interface, the secondary private IPv4 address retains its association with an Elastic IP address.
• When a secondary private IPv4 address is unassigned from an interface, an associated Elastic IP address is automatically disassociated from the secondary private IPv4 address.

Work 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
• Assign a secondary private IPv4 address (p. 902)
• Configure the operating system on your instance to recognize secondary private IPv4 addresses (p. 904)
• Associate an Elastic IP address with the secondary private IPv4 address (p. 904)
• View your secondary private IPv4 addresses (p. 904)
• Unassign a secondary private IPv4 address (p. 905)

Assign 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. 902)
• To assign a secondary IPv4 address during launch using the command line (p. 903)
• To assign a secondary private IPv4 address to a network interface (p. 903)
• To assign a secondary private IPv4 to an existing instance using the command line (p. 904)

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. 936).

  **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 [Work with Elastic IP addresses](#) (p. 927).

- 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 [Configure the operating system on your instance to recognize secondary private IPv4 addresses](#) (p. 904).

**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 [Access 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/](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 Access Amazon EC2 (p. 3).
  - assign-private-ip-addresses (AWS CLI)
  - Register-EC2PrivatetIpAddress (AWS Tools for Windows PowerShell)

**Configure the operating system on your instance to recognize secondary private IPv4 addresses**

After you assign a secondary private IPv4 address to your instance, you need to configure the operating system on your instance to recognize the secondary private IP address.

For information about configuring a Windows instance, see Configure a secondary private IPv4 address for your Windows instance (p. 567).

**Associate 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 Access Amazon EC2 (p. 3).
  - associate-address (AWS CLI)
  - Register-EC2Address (AWS Tools for Windows PowerShell)

**View 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.

Unassign 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 Access Amazon EC2 (p. 3).
  • unassign-private-ip-addresses (AWS CLI)
  • Unregister-EC2PrivatetlpAddress (AWS Tools for Windows PowerShell)

Work 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

• Assign multiple IPv6 addresses (p. 906)
Assign 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 support IPv6. For more information, see Instance types (p. 141).
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. 936).
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 for different use cases (p. 1149). Choose Review and Launch.
8. On the Review Instance Launch page, review your settings, and then choose Launch to choose a key pair and launch your instance. If you're new to Amazon EC2 and haven't created any key pairs, the wizard prompts you to create one.

You can use the Instances screen Amazon EC2 console to assign multiple IPv6 addresses to an existing instance. This assigns the IPv6 addresses to the primary network interface (eth0) for the instance. To assign a specific IPv6 address to the instance, ensure that the IPv6 address is not already assigned to another instance or network interface.

To assign multiple IPv6 addresses to an existing instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select your instance, choose Actions, Networking, Manage IP Addresses.
4. Under IPv6 Addresses, choose Assign new IP for each IPv6 address you want to add. You can specify an IPv6 address from the range of the subnet, or leave the Auto-assign value to let Amazon choose an IPv6 address for you.
5. Choose Yes, Update.
Alternatively, you can assign multiple IPv6 addresses to an existing network interface. The network interface must have been created in a subnet that has an associated IPv6 CIDR block. To assign a specific IPv6 address to the network interface, ensure that the IPv6 address is not already assigned to another network interface.

To assign multiple IPv6 addresses to a network interface

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Network Interfaces.
3. Select your network interface, choose Actions, Manage IP Addresses.
4. Under IPv6 Addresses, choose Assign new IP for each IPv6 address you want to add. You can specify an IPv6 address from the range of the subnet, or leave the Auto-assign value to let Amazon choose an IPv6 address for you.
5. Choose Yes, Update.

CLI overview

You can use one of the following commands. For more information about these command line interfaces, see Access 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)

View 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 Access Amazon EC2 (p. 3).

- View the IPv6 addresses for an instance:
  - describe-instances (AWS CLI)
Unassign 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 **Access Amazon EC2 (p. 3)**.

- **unassign-ipv6-addresses** (AWS CLI)
- **Unregister-EC2Ipv6AddressList** (AWS Tools for Windows PowerShell).

Bring your own IP addresses (BYOIP) in Amazon EC2

You can bring part or all of your publicly routable IPv4 or IPv6 address range from your on-premises network to your AWS account. You continue to own the address range, but AWS advertises it on the internet by default. After you bring the address range to AWS, it appears in your AWS account as an address pool.

BYOIP is not available in all Regions and for all resources. For a list of supported Regions and resources, see the FAQ for Bring Your Own IP.

**Note**

The following steps describe how to bring your own IP address range for use in Amazon EC2 only. For steps to bring your own IP address range for use in AWS Global Accelerator, see **Bring your own IP addresses (BYOIP)** in the AWS Global Accelerator Developer Guide.
Requirements and quotas

• The address range must be registered with your regional internet registry (RIR), such as the American Registry for Internet Numbers (ARIN), Réseaux IP Européens Network Coordination Centre (RIPE), or Asia-Pacific Network Information Centre (APNIC). It must be registered to a business or institutional entity and cannot be registered to an individual person.
• The most specific IPv4 address range that you can bring is /24.
• The most specific IPv6 address range that you can bring is /48 for CIDRs that are publicly advertised, and /56 for CIDRs that are not publicly advertised (p. 915).
• You can bring each address range to one Region at a time.
• You can bring a total of five IPv4 and IPv6 address ranges per Region to your AWS account.
• You cannot share your IP address range with other accounts using AWS Resource Access Manager (AWS RAM).
• The addresses in the IP address range must have a clean history. We might investigate the reputation of the IP address range and reserve the right to reject an IP address range if it contains an IP address that has a poor reputation or is associated with malicious behavior.
• You must own the IP address that you use. This means that only the following are supported:
  • ARIN - "Direct Allocation" and "Direct Assignment" network types
  • RIPE - "ALLOCATED PA", "LEGACY", "ASSIGNED PI", and "ALLOCATED-BY-RIR" allocation statuses
  • APNIC – “ALLOCATED PORTABLE” and "ASSIGNED PORTABLE" allocation statuses

Configure your BYOIP address range

The process to configure BYOIP has these phases:

• Preparation
  
  For authentication purposes, create an RSA key pair and use it to generate a self-signed X.509 certificate.

• RIR configuration

  Register with the Resource Public Key Infrastructure (RPKI) of your RIR, and file a Route Origin Authorization (ROA) that defines the desired address range, the autonomous system numbers (ASNs) allowed to advertise the address range, and an expiration date. Upload the self-signed certificate to your RDAP record comments.

• Amazon configuration

  Sign a CIDR authorization context message with the private RSA key that you created, and upload the message and signature to Amazon using the AWS Command Line Interface.

To bring on multiple address ranges, you must repeat this process with each address range. Bringing on an address range has no effect on any address ranges that you brought on previously.

To configure BYOIP, complete the following tasks. For some tasks, you run Linux commands. On Windows, you can use the Windows Subsystem for Linux to run the Linux commands.
Configure your BYOIP address range

Tasks

- Create a key pair and certificate (p. 910)
- Create an ROA object in your RIR (p. 913)
- Update the RDAP record in your RIR (p. 913)
- Provision the address range in AWS (p. 914)
- Advertise the address range through AWS (p. 915)
- Deprovision the address range (p. 916)

Create a key pair and certificate

Use the following procedure to create a self-signed X.509 certificate and add it to the RDAP record for your RIR. The `openssl` commands require OpenSSL version 1.0.2 or later.

Copy the commands below and replace only the placeholder values (in colored italic text).

To create a self-signed X.509 certificate and add it to the RDAP record

This procedure follows the best practice of encrypting your private RSA key and requiring a pass phrase to access it.

1. Generate an RSA 2048-bit key pair as shown in the following.

```
$ openssl genpkey -aes256 -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out private-key.pem
```

The `-aes256` parameter specifies the algorithm used to encrypt the private key. The command returns the following output, including prompts to set a pass phrase:

```
......+++ 
.+ . Enter PEM pass phrase: xxxxxxxx
Verifying - Enter PEM pass phrase: xxxxxxxx
```

You can inspect the key using the following command:

```
$ openssl pkey -in private-key.pem -text
```

This returns a pass-phrase prompt and the contents of the key, which should be similar to the following:

```
Enter pass phrase for private-key.pem: xxxxxxxx
-----BEGIN PRIVATE KEY-----
MIIBvqIBADANBgkqhkiG9w0BAQEFAAECBfAqKAgIBAgIu6Gm3x5x67A55dVJXmefozXxHdunposVAhPv6xtNSFUNbshbKxj5mYCOOvYCyกรมI4k9pslOlAHjDFj2DN0wCw8wbsV
```

910
Configure your BYOIP address range

```
-----BEGIN PRIVATE KEY-----
Private-Key: (2048 bit)
modulus:
publicExponent: 65537 (0x10001)
-----END PRIVATE KEY-----
```
Keep your private key in a secure location when it is not in use.

2. Generate your public key from the private key as follows. You will use this later to test that your signed authorization message validates correctly.

```
# openssl rsa -in private-key.pem -pubout > public-key.pem
```

On inspection, your public key should look like this:

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxQVx0SOB1SgIYd7HonIqKIswkU2yTxm4mWTU4u2f8isqVR1BVgx0wBOQ0hS5Tj1Lfjaj+yMPegQDwz20t8aPGKwEqgZ9f9KdubAyab399OyhzUZmWVJp9MxzkhKTZm8n2MSBgoc6JaUCjQ4s3JK
5b7x7yvTGPQwCwClrk7L+ZdLSspMwvdSuPrT1tHQ0eoUp6OVIwMX7AA1t4NcnaW0
wbMflu2f11mnDQGcuxYxSN2g2lyQeq+p7heh/JKyc0C+LSDehDpOQ1SQ25TZXs
Hm6Kp1oncG4G5jig/8/fL5PD/v/oVKwBsf9H0bWxVyyM3NVkRzqzEp9gc7D1bg
yWlDAQAB
-----END PUBLIC KEY-----
```

3. Generate an X.509 certificate using the key pair created in the previous. In this example, the certificate expires in 365 days, after which time it cannot be trusted. Be sure to set the expiration appropriately. The `tr -d \"\n\"` command strips newline characters (line breaks) from the output.

You need to provide a Common Name when prompted, but the other fields can be left blank.
Configure your BYOIP address range

```
# openssl req -new -x509 -key private-key.pem -days 365 | tr -d "\n" > certificate.pem
```

This results in output similar to the following:

```
Enter pass phrase for private-key.pem: xxxxxxx
You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

-----
Country Name (2 letter code) []: 
State or Province Name (full name) []: 
Locality Name (eg, city) []: 
Organization Name (eg, company) []: 
Organizational Unit Name (eg, section) []: 
Common Name (eg, fully qualified host name) []: example.com
Email Address []:

-----
```

You can inspect the certificate with the following command:

```
$ cat certificate.pem
```

The output should be a long, PEM-encoded string without line breaks, prefaced by `-----BEGIN CERTIFICATE-----` and followed by `-----END CERTIFICATE-----`.

Create an ROA object in your RIR

Create an ROA object to authorize Amazon ASNs 16509 and 14618 to advertise your address range, as well as the ASNs that are currently authorized to advertise the address range. You must set the maximum length to the size of the smallest prefix that you want to bring (for example, /24). It might take up to 24 hours for the ROA to become available to Amazon. For more information, consult your RIR:

- ARIN — ROA Requests
- RIPE — Managing ROAs
- APNIC — Route Management

Update the RDAP record in your RIR

Add the certificate that you previously created to the RDAP record for your RIR. Be sure to include the `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----` strings before and after the encoded portion. All of this content must be on a single, long line. The procedure for updating RDAP depends on your RIR:

- For ARIN, add the certificate in the "Public Comments" section for your address range. Do not add it to the comments section for your organization.
- For RIPE, add the certificate as a new "descr" field for your address range. Do not add it to the comments section for your organization.
- For APNIC, email the public key to helpdesk@apnic.net to manually add it to the "remarks" field for your address range. Send the email using the APNIC authorized contact for the IP addresses.
Provision the address range in AWS

When you provision an address range for use with AWS, you are confirming that you own the address range and are authorizing Amazon to advertise it. We also verify that you own the address range through a signed authorization message. This message is signed with the self-signed X.509 key pair that you used when updating the RDAP record with the X.509 certificate. AWS requires a cryptographically signed authorization message that it presents to the RIR. The RIR authenticates the signature against the certificate that you added to RDAP, and checks the authorization details against the ROA.

To provision the address range

1. **Compose message**
   
   Compose the plaintext authorization message. The format of the message is as follows, where the date is the expiry date of the message:
   
   ```
   1|aws|account|cidr|YYYYMMDD|SHA256|RSAPSS
   ```
   
   Replace the account number, address range, and expiry date with your own values to create a message resembling the following:
   
   ```
   1|aws|0123456789AB|198.51.100.0/24|20211231|SHA256|RSAPSS
   ```
   
   This is not to be confused with an ROA message, which has a similar appearance.

2. **Sign message**
   
   Sign the plaintext message using the private key that you created previously. The signature returned by this command is a long string that you need to use in the next step.

   ```
   # echo -n "1|aws|123456789012|198.51.100.0/24|20211231|SHA256|RSAPSS" | openssl dgst -sha256 -sigopt rsa_padding_mode:pss -sigopt rsa_pss_saltlen:-1 -sign private-key.pem -keyform PEM | openssl base64 | tr -- '+/ ' '-_' | tr -d \n
   ```

3. **Provision address**
   
   Use the AWS CLI provision-byoip-cidr command to provision the address range. The `--cidr-authorization-context` option uses the message and signature strings that you created previously.

   ```
   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.

4. **Monitor progress**
   
   It can take up to three weeks to complete the provisioning process for publicly advertisable ranges. Use the `describe-byoip-cidrs` command to monitor progress, as in this example:

   ```
   aws ec2 describe-byoip-cidrs --max-results 5
   ```
If there are issues during provisioning and the status goes to `failed-provision`, you must run the `provision-byoip-cidr` command again after the issues have been resolved.

**Provision an IPv6 address range that’s not publicly advertised**

By default, an address range is provisioned to be publicly advertised to the internet. You can provision an IPv6 address range that will not be publicly advertised. For routes that are not publicly advertisable, the provisioning process generally completes within minutes. When you associate an IPv6 CIDR block from a non-public address range with a VPC, the IPv6 CIDR can only be accessed through an AWS Direct Connect connection.

An ROA is not required to provision a non-public address range.

**Important**
You can only specify whether an address range is publicly advertised during provisioning. You cannot change the advertisable status later on.

To provision an IPv6 address range that will not be publicly advertised, use the following `provision-byoip-cidr` command.

```bash
aws ec2 provision-byoip-cidr --cidr address-range --cidr-authorization-context Message="$text_message",Signature="$signed_message" --no-publicly-advertisable
```

**Advertise the address range through AWS**

After the address range is provisioned, it is ready to be advertised. You must advertise the exact address range that you provisioned. You can’t advertise only a portion of the provisioned address range.

If you provisioned an IPv6 address range that will not be publicly advertised, you do not need to complete this step.

We recommend that you stop advertising the address range from other locations before you advertise it through AWS. If you keep advertising your IP address range from other locations, we can’t reliably support it or troubleshoot issues. Specifically, we can’t guarantee that traffic to the address range will enter our network.

To minimize down time, you can configure your AWS resources to use an address from your address pool before it is advertised, and then simultaneously stop advertising it from the current location and start advertising it through AWS. For more information about allocating an Elastic IP address from your address pool, see Allocate an Elastic IP address (p. 927).

**Limitations**

- You can run the `advertise-byoip-cidr` command at most once every 10 seconds, even if you specify different address ranges each time.
- You can run the `withdraw-byoip-cidr` command at most once every 10 seconds, even if you specify different address ranges each time.

To advertise the address range, use the following `advertise-byoip-cidr` command.

```bash
aws ec2 advertise-byoip-cidr --cidr address-range
```

To stop advertising the address range, use the following `withdraw-byoip-cidr` command.

```bash
aws ec2 withdraw-byoip-cidr --cidr address-range
```
Deprovision the address range

To stop using your address range with AWS, first release any Elastic IP addresses and disassociate any IPv6 CIDR blocks that are still allocated from the address pool. Then stop advertising the address range, and finally, deprovision the address range.

You cannot deprovision a portion of the address range. If you want to use a more specific address range with AWS, deprovision the entire address range and provision a more specific address range.

(IPv4) To release each Elastic IP address, use the following `release-address` command.

```bash
aws ec2 release-address --allocation-id eipalloc-12345678abcabcabc
```

(IPv6) To disassociate an IPv6 CIDR block, use the following `disassociate-vpc-cidr-block` command.

```bash
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-12345abcd1234abc1
```

To stop advertising the address range, use the following `withdraw-byoip-cidr` command.

```bash
aws ec2 withdraw-byoip-cidr --cidr address-range
```

To deprovision the address range, use the following `deprovision-byoip-cidr` command.

```bash
aws ec2 deprovision-byoip-cidr --cidr address-range
```

It can take up to a day to deprovision an address range.

Work with your address range

You can view and use the IPv4 and IPv6 address ranges that you've provisioned in your account.

IPv4 address ranges

You can create an Elastic IP address from your IPv4 address pool and use it with your AWS resources, such as EC2 instances, NAT gateways, and Network Load Balancers.

To view information about the IPv4 address pools that you've provisioned in your account, use the following `describe-public-ipv4-pools` command.

```bash
aws ec2 describe-public-ipv4-pools
```

To create an Elastic IP address from your IPv4 address pool, use the `allocate-address` command. You can use the `--public-ipv4-pool` option to specify the ID of the address pool returned by `describe-byoip-cidrs`. Or you can use the `--address` option to specify an address from the address range that you provisioned.

IPv6 address ranges

To view information about the IPv6 address pools that you've provisioned in your account, use the following `describe-ipv6-pools` command.

```bash
aws ec2 describe-ipv6-pools
```
To create a VPC and specify an IPv6 CIDR from your IPv6 address pool, use the following `create-vpc` command. To let Amazon choose the IPv6 CIDR from your IPv6 address pool, omit the `--ipv6-cidr-block` option.

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --ipv6-cidr-block ipv6-cidr --ipv6-pool pool-id
```

To associate an IPv6 CIDR block from your IPv6 address pool with a VPC, use the following `associate-vpc-cidr-block` command. To let Amazon choose the IPv6 CIDR from your IPv6 address pool, omit the `--ipv6-cidr-block` option.

```
aws ec2 associate-vpc-cidr-block --vpc-id vpc-123456789abc123ab --ipv6-cidr-block ipv6-cidr --ipv6-pool pool-id
```

To view your VPCs and the associated IPv6 address pool information, use the `describe-vpcs` command. To view information about associated IPv6 CIDR blocks from a specific IPv6 address pool, use the following `get-associated-ipv6-pool-cidrs` command.

```
aws ec2 get-associated-ipv6-pool-cidrs --pool-id pool-id
```

If you disassociate the IPv6 CIDR block from your VPC, it's released back into your IPv6 address pool.

For more information about working with IPv6 CIDR blocks in the VPC console, see Working with VPCs and Subnets in the Amazon VPC User Guide.

Learn more

For more information, see the AWS Online Tech talk Deep Dive on Bring Your Own IP.

Assigning prefixes to Amazon EC2 network interfaces

You can assign a private IPv4 or IPv6 CIDR range, either automatically or manually, to your network interfaces. By assigning prefixes, you scale and simplify the management of applications, including container and networking applications that require multiple IP addresses on an instance.

The following assignment options are available:

- **Automatic assignment** — AWS chooses the prefix from your VPC subnet's IPv4 or IPv6 CIDR and assigns to your network interface.
- **Manual Assignment** — You specify the prefix from your VPC subnet's IPv4 and IPv6 CIDRs, and AWS verifies that the prefix is not already assigned to other resources before assigning it to your network interface.

Assigning prefixes has the following benefits:

- **Increased IP addresses on a network interface** — When you use a prefix, you assign a block of IP addresses as opposed to individual IP addresses. This increases the number of IP addresses on a network interface.
- **Simplified VPC management for containers** — In container applications, each container requires a unique IP address. Assigning prefixes to your instance simplifies the management of your VPCs, as you can launch and terminate containers without having to call Amazon EC2 APIs for individual IP assignments.
Basics for assigning prefixes

- You can assign a prefix to new or existing network interfaces.
- To use prefixes, you first assign a prefix to your network interface, then attach the network interface to your instance, and then configure your operating system.
- When you choose the option to specify a prefix, the prefix must meet the following requirements:
  - The IPv4 prefix that you can specify is /28.
  - The IPv6 prefix that you can specify is /80.
  - The prefix is in the subnet CIDR of the network interface, and it does not overlap with other prefixes or IP addresses assigned to existing resources in the subnet.
- You can assign a prefix to the primary or secondary network interface.
- You can assign an Elastic IP address to a network interface that has a prefix assigned to it.
- We resolve the private DNS host name of an instance to the primary private IPv4 address.
- We assign each private IPv4 address on a network interface, including those from prefixes, with the following forms:
  - us-east-1 Region
    - ip-private-ipv4-address.ec2.internal
  - All other Regions
    - ip-private-ipv4-address.region.compute.internal

Considerations and limits for prefixes

Take the following into consideration when you use prefixes:

- Network interfaces with prefixes are supported with nitro-based instances.
- Prefixes for network interfaces are limited to private IPv4 and IPv6 addresses.
- See IP addresses per network interface per instance type (p. 936) for limitations.
- The number of prefixes and IP addresses on a network interface must be less than the limit on the instance that the network interface is associated with. For example, if you have a c5.large instance, the limit is 10 IPv4 addresses and 10 IPv6 addresses on a network interface, and the total number of /28 and /80 prefixes must be less than 10.
- Prefixes are included in source/destination checks.
• Configure your operating system for network interfaces with prefixes (p. 924)
• View the prefixes assigned your network interfaces (p. 924)
• Remove prefixes from your network interfaces (p. 925)

Assign prefixes during network interface creation

You can use the `create-network-interface` AWS CLI command to assign prefixes to a new network interface during creation.

If you use the automatic assignment option, you can reserve a block of IP addresses in your subnet. AWS chooses the prefixes from this block. For more information, see Subnet CIDR reservations in the Amazon VPC User Guide.

After you have created the network interface, use the `attach-network-interface` AWS CLI command to attach the network interface to your instance, and you must configure your operating system to work with network interfaces with prefixes. For more information, see Configure your operating system for network interfaces with prefixes (p. 924).

Topics
• Assign automatic prefixes during network interface creation (p. 919)
• Assign specific prefixes during network interface creation (p. 921)

Assign automatic prefixes during network interface creation

To assign automatic IPv4 prefixes during network interface creation

Use the `create-network-interface` command and set `--ipv4-prefix-count` to the number of prefixes that you want AWS to assign. In the following example, AWS assigns 1 prefix.

```bash
C:\> aws ec2 create-network-interface \
   --subnet-id subnet-047cfed18eEXAMPLE \
   --description "IPv4 automatic example" \
   --ipv4-prefix-count 1
```

Example output

```json
{
   "NetworkInterface": {
      "AvailabilityZone": "us-west-2a",
      "Description": "IPv4 automatic example",
      "Groups": [
         {
            "GroupName": "default",
            "GroupId": "sg-044c2de2c4EXAMPLE"
         }
      ],
      "InterfaceType": "interface",
      "Ipv6Addresses": [],
      "MacAddress": "02:98:65:dd:18:47",
      "NetworkInterfaceId": "eni-02b80b4668EXAMPLE",
      "OwnerId": "123456789012",
      "PrivateIpAddress": "10.0.0.62",
      "PrivateIpAddresses": [
         {
            "Primary": true,
            "PrivateIpAddress": "10.0.0.62"
         }
      ]
   }
}
```
To assign automatic IPv6 prefixes during network interface creation

Use the `create-network-interface` command and set `--ipv6-prefix-count` to the number of prefixes that you want AWS to assign. In the following example, AWS assigns 1 prefix.

```
C:\> aws ec2 create-network-interface \
--subnet-id subnet-047cfed18eEXAMPLE \
--description "IPv6 automatic example" \
--ipv6-prefix-count 1
```

Example output

```
{
   "NetworkInterface": {
      "AvailabilityZone": "us-west-2a",
      "Description": "IPv6 automatic example",
      "Groups": [
         {
            "GroupName": "default",
            "GroupId": "sg-044c2de2c4EXAMPLE"
         }
      ],
      "InterfaceType": "interface",
      "Ipv6Addresses": [],
      "MacAddress": "02:bb:e4:31:fe:09",
      "NetworkInterfaceId": "eni-006edbcfa4EXAMPLE",
      "OwnerId": "123456789012",
      "PrivateIpAddress": "10.0.0.73",
      "PrivateIpAddresses": [
         {
            "Primary": true,
            "PrivateIpAddress": "10.0.0.73"
         }
      ],
      "Ipv6Prefixes": [
         {
            "Ipv6Prefix": "2600:1f13:fc2:a700:1768::/80"
         }
      ],
      "RequesterId": "AIDAIV5AJISLX5XIPDPCO",
      "RequesterManaged": false,
      "SourceDestCheck": true,
      "Status": "pending",
      "SubnetId": "subnet-047cfed18eEXAMPLE",
      "TagSet": [],
      "VpcId": "vpc-0e12f52b21EXAMPLE"
   }
}
```
Assign specific prefixes during network interface creation

To assign specific IPv4 prefixes during network interface creation

Use the `create-network-interface` command and set `--ipv4-prefixes` to the prefixes. AWS selects IP addresses from this range. In the following example, the prefix CIDR is `10.0.0.208/28`.

```bash
C:\> aws ec2 create-network-interface \
   --subnet-id subnet-047cfed18eEXAMPLE \
   --description "IPv4 manual example" \
   --ipv4-prefixes Ipv4Prefix=10.0.0.208/28
```

Example output

```json
{
   "NetworkInterface": {
      "AvailabilityZone": "us-west-2a",
      "Description": "IPv4 manual example",
      "Groups": [
         {
            "GroupName": "default",
            "GroupId": "sg-044c2de2c4EXAMPLE"
         }
      ],
      "InterfaceType": "interface",
      "Ipv6Addresses": [],
      "MacAddress": "02:98:65:dd:18:47",
      "NetworkInterfaceId": "eni-02b808b4668EXAMPLE",
      "OwnerId": "123456789012",
      "PrivateIpAddress": "10.0.0.62",
      "PrivateIpAddresses": [
         {
            "Primary": true,
            "PrivateIpAddress": "10.0.0.62"
         }
      ],
      "Ipv4Prefixes": [
         {
            "Ipv4Prefix": "10.0.0.208/28"
         }
      ],
      "RequesterId": "AIDAIV5AJI5LX53P5F5DPCO",
      "RequesterManaged": false,
      "SourceDestCheck": true,
      "Status": "pending",
      "SubnetId": "subnet-047cfed18eEXAMPLE",
      "TagSet": [],
      "VpcId": "vpc-0e12f52b21EXAMPLE"
   }
}
```

To assign specific IPv6 prefixes during network interface creation

Use the `create-network-interface` command and set `--ipv6-prefixes` to the prefixes. AWS selects IP addresses from this range. In the following example, the prefix CIDR is `2600:1f13:fc2:a700:1768::/80`.

```bash
C:\> aws ec2 create-network-interface \
   --subnet-id subnet-047cfed18eEXAMPLE \
```

Work with prefixes

IPv6 manual example
--ipv6-prefixes Ipv6Prefix=2600:1f13:fc2:a700:1768::/80

Example output

```json
{
  "NetworkInterface": {
    "AvailabilityZone": "us-west-2a",
    "Description": "IPv6 automatic example",
    "Groups": [
      {
        "GroupName": "default",
        "GroupId": "sg-044c2de2c4EXAMPLE"
      }
    ],
    "InterfaceType": "interface",
    "Ipv6Addresses": [],
    "MacAddress": "02:bb:e4:31:fe:09",
    "NetworkInterfaceId": "eni-006edbcfa4EXAMPLE",
    "OwnerId": "123456789012",
    "PrivateIpAddress": "10.0.0.73",
    "PrivateIpAddresses": [
      {
        "Primary": true,
        "PrivateIpAddress": "10.0.0.73"
      }
    ],
    "Ipv6Prefixes": [
      {
        "Ipv6Prefix": "2600:1f13:fc2:a700:1768::/80"
      }
    ],
    "RequesterId": "AIDAIV5AJI5LXF5XXDCO",
    "RequesterManaged": false,
    "SourceDestCheck": true,
    "Status": "pending",
    "SubnetId": "subnet-047cfed18eEXAMPLE",
    "TagSet": [],
    "VpcId": "vpc-0e1f52b21EXAMPLE"
  }
}
```

Assign prefixes to existing network interfaces

You can use the `assign-ipv6-addresses` command to assign IPv6 prefixes and the `assign-private-ip-addresses` command to assign IPv4 prefixes to existing network interfaces.

After you have assigned the prefixes, use the `attach-network-interface` AWS CLI command to attach the network interface to your instance, and you must configure your operating system to work with network interfaces with prefixes. For more information, see Configure your operating system for network interfaces with prefixes (p. 924).

Assign automatic prefixes to an existing network interface

To assign automatic IPv4 prefixes to an existing network interface

Use the `assign-private-ip-addresses` command and set `--ipv4-prefix-count` to the number of prefixes that you want AWS to assign. In the following example, AWS assigns 1 IPv4 prefix.

```bash
C:\> aws ec2 assign-private-ip-addresses --network-interface-id eni-081fbb4095EXAMPLE
```
Work with prefixes

---ipv4-prefix-count 1

Example output

```json
{
    "NetworkInterfaceId": "eni-081fbb4095EXAMPLE",
    "AssignedIpv4Prefixes": [
        {
            "Ipv4Prefix": "10.0.0.176/28"
        }
    ]
}
```

**To assign automatic IPv6 prefixes to an existing network interface**

Use the `assign-ipv6-addresses` command and set `--ipv6-prefix-count` to the number of prefixes that you want AWS to assign. In the following example, AWS assigns 1 IPv6 prefix.

```bash
C:\> aws ec2 assign-ipv6-addresses \
    --network-interface-id eni-00d577338cEXAMPLE \
    --ipv6-prefix-count 1
```

Example output

```json
{
    "AssignedIpv6Prefixes": [
        "2600:1f13:fc2:a700:18bb::/80"
    ],
    "NetworkInterfaceId": "eni-00d577338cEXAMPLE"
}
```

**Assign specific prefixes to an existing network interface**

**Assign specific IPv4 prefixes to an existing network interface**

Use the `assign-private-ip-addresses` command and set `--ipv4-prefixes` to the prefix. AWS selects IPv4 addresses from this range. In the following example, the prefix CIDR is `10.0.0.208/28`.

```bash
C:\> aws ec2 assign-private-ip-addresses \
    --network-interface-id eni-081fbb4095EXAMPLE \
    --ipv4-prefixes 10.0.0.208/28
```

Example output

```json
{
    "NetworkInterfaceId": "eni-081fbb4095EXAMPLE",
    "AssignedIpv4Prefixes": [
        {
            "Ipv4Prefix": "10.0.0.208/28"
        }
    ]
}
```

**Assign specific IPv6 prefixes to an existing network interface**

Use the `assign-ipv6-addresses` command and set `--ipv6-prefixes` to the prefix. AWS selects IPv6 addresses from this range. In the following example, the prefix CIDR is `2600:1f13:fc2:a700:18bb::/80`.
Work with prefixes

C:\> aws ec2 assign-ipv6-addresses
   --network-interface-id eni-00d577338cEXAMPLE
   --ipv6-prefixes 2600:1f13:fc2:a700:18bb::/80

Example output

{
   "NetworkInterfaceId": "eni-00d577338cEXAMPLE",
   "AssignedIpv6Prefixes": [
      {
         "Ipv6Prefix": "2600:1f13:fc2:a700:18bb::/80"
      }
   ]
}

Configure your operating system for network interfaces with prefixes

Amazon Linux AMIs might contain additional scripts installed by AWS, known as ec2-net-utils. These scripts optionally automate the configuration of your network interfaces. They are available for Amazon Linux only.

If you are not using Amazon Linux, you can use a Container Network Interface (CNI) for Kubernetes plugin, or dockerd if you use Docker to manage your containers.

View the prefixes assigned your network interfaces

You can use the describe-network-interfaces AWS CLI command to view the prefixes assigned to your network interfaces.

C:\> aws ec2 describe-network-interfaces

Example output

{
   "NetworkInterfaces": [
      {
         "AvailabilityZone": "us-west-2a",
         "Description": "IPv4 automatic example",
         "Groups": [
            {
               "GroupName": "default",
               "GroupId": "sg-044c2de2c4EXAMPLE"
            }
         ],
         "InterfaceType": "interface",
         "Ipv6Addresses": [],
         "MacAddress": "02:98:65:dd:18:47",
         "NetworkInterfaceId": "eni-02b80b4668EXAMPLE",
         "OwnerId": "123456789012",
         "PrivateIpAddress": "10.0.0.62",
         "PrivateIpAddresses": [
            {
               "Primary": true,
               "PrivateIpAddress": "10.0.0.62"
            }
         ],
         "Ipv4Prefixes": [
            {"Address": "10.0.0.0/24"}
         ]
      }]
}

924
Remove prefixes from your network interfaces

You can use the `unassign-ipv6-addresses` command to remove IPv6 prefixes and the `unassign-private-ip-addresses` commands to remove IPv4 prefixes from your existing network interfaces.

To remove IPv4 prefixes from a network interface

Use the `unassign-private-ip-addresses` command and set `--ipv4-prefix` to the address that you want to remove.

```bash
C:\> aws ec2 unassign-private-ip-addresses \
```
To remove IPv6 prefixes from a network interface

Use the `unassign-ipv6-addresses` command and set `--ipv6-prefix` to the address that you want to remove.

```
C:\> aws ec2 unassign-ipv6-addresses \
   --network-interface-id eni-00d577338cEXAMPLE \
   --ipv6-prefix 2600:1f13:fc2:a700:18bb::/80
```

---

### Elastic IP addresses

An **Elastic IP address** is a static IPv4 address designed for dynamic cloud computing. An Elastic IP address is allocated to your AWS account, and is yours until you release it. By using an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account. Alternatively, you can specify the Elastic IP address in a DNS record for your domain, so that your domain points to your instance. For more information, see the documentation for your domain registrar.

An Elastic IP address is a public IPv4 address, which is reachable from the internet. If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet. For example, this allows you to connect to your instance from your local computer.

We currently do not support Elastic IP addresses for IPv6.

#### Contents
- Elastic IP address pricing (p. 926)
- Elastic IP address basics (p. 926)
- Work with Elastic IP addresses (p. 927)
- Use reverse DNS for email applications (p. 933)
- Elastic IP address limit (p. 934)

### Elastic IP address pricing

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 Elastic IP Addresses on the Amazon EC2 Pricing, On-Demand Pricing page.

### Elastic IP address basics

The following are the basic characteristics of an Elastic IP address:

- An Elastic IP address is static; it does not change over time.
- To use an Elastic IP address, you first allocate one to your account, and then associate it with your instance or a network interface.
• When you associate an Elastic IP address with an instance, it is also associated with the instance's primary network interface. When you associate an Elastic IP address with a network interface that is attached to an instance, it is also associated with the instance.

• When you associate an Elastic IP address with an instance or its primary network interface, the instance's public IPv4 address (if it had one) is released back into Amazon's pool of public IPv4 addresses. You cannot reuse a public IPv4 address, and you cannot convert a public IPv4 address to an Elastic IP address. For more information, see Public IPv4 addresses and external DNS hostnames (p. 894).

• You can disassociate an Elastic IP address from a resource, and then associate it with a different resource. To avoid unexpected behavior, ensure that all active connections to the resource named in the existing association are closed before you make the change. After you have associated your Elastic IP address to a different resource, you can reopen your connections to the newly associated resource.

• A disassociated Elastic IP address remains allocated to your account until you explicitly release it. We impose a small hourly charge for Elastic IP addresses that are not associated with a running instance.

• When you associate an Elastic IP address with an instance that previously had a public IPv4 address, the public DNS host name of the instance changes to match the Elastic IP address.

• We resolve a public DNS host name to the public IPv4 address or the Elastic IP address of the instance outside the network of the instance, and to the private IPv4 address of the instance from within the network of the instance.

• An Elastic IP address comes from Amazon's pool of IPv4 addresses, or from a custom IP address pool that you have brought to your AWS account.

• When you allocate an Elastic IP address from an IP address pool that you have brought to your AWS account, it does not count toward your Elastic IP address limits. For more information, see Elastic IP address limit (p. 934).

• When you allocate the Elastic IP addresses, you can associate the Elastic IP addresses with a network border group. This is the location from which we advertise the CIDR block. Setting the network border group limits the CIDR block to this group. If you do not specify the network border group, we set the border group containing all of the Availability Zones in the Region (for example, us-west-2).

• An Elastic IP address is for use in a specific network border group only.

• An Elastic IP address is for use in a specific Region only, and cannot be moved to a different Region.

Work with Elastic IP addresses

The following sections describe how you can work with Elastic IP addresses.

Tasks

• Allocate an Elastic IP address (p. 927)
• Describe your Elastic IP addresses (p. 928)
• Tag an Elastic IP address (p. 929)
• Associate an Elastic IP address with an instance or network interface (p. 930)
• Disassociate an Elastic IP address (p. 931)
• Release an Elastic IP address (p. 932)
• Recover an Elastic IP address (p. 933)

Allocate an Elastic IP address

You can allocate an Elastic IP address from Amazon's pool of public IPv4 addresses, or from a custom IP address pool that you have brought to your AWS account. For more information about bringing your own IP address range to your AWS account, see Bring your own IP addresses (BYOIP) in Amazon EC2 (p. 908).
You can allocate an Elastic IP address using one of the following methods.

New console

**To allocate an Elastic IP address**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Network & Security, Elastic IPs**.
3. Choose **Allocate Elastic IP address**.
4. For **Public IPv4 address pool**, choose one of the following:
   - **Amazon's pool of IPv4 addresses** — If you want an IPv4 address to be allocated from Amazon's pool of IPv4 addresses.
   - **My pool of public IPv4 addresses** — If you want to allocate an IPv4 address from an IP address pool that you have brought to your AWS account. This option is disabled if you do not have any IP address pools.
   - **Customer owned pool of IPv4 addresses** — If you want to allocate an IPv4 address from a pool created from your on-premises network for use with an AWS Outpost. This option is disabled if you do not have an AWS Outpost.
5. (Optional) Add or remove a tag.
   - [Add a tag] Choose **Add new tag** and do the following:
     - For **Key**, enter the key name.
     - For **Value**, enter the key value.
   - [Remove a tag] Choose **Remove** to the right of the tag's Key and Value.
6. Choose **Allocate**.

Old console

**To allocate an Elastic IP address**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate new address**.
4. For **IPv4 address pool**, choose **Amazon pool**.
5. Choose **Allocate**, and close the confirmation screen.

AWS CLI

**To allocate an Elastic IP address**

Use the `allocate-address` AWS CLI command.

PowerShell

**To allocate an Elastic IP address**


Describe your Elastic IP addresses

You can describe an Elastic IP address using one of the following methods.
New console

**To describe your Elastic IP addresses**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to view and choose **Actions, View details**.

Old console

**To describe your Elastic IP addresses**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Select a filter from the Resource Attribute list to begin searching. You can use multiple filters in a single search.

AWS CLI

**To describe your Elastic IP addresses**

Use the `describe-addresses` AWS CLI command.

PowerShell

**To describe your Elastic IP addresses**


**Tag an Elastic IP address**

You can assign custom tags to your Elastic IP addresses to categorize them in different ways, for example, by purpose, owner, or environment. This helps you to quickly find a specific Elastic IP address based on the custom tags that you assigned to it.

Cost allocation tracking using Elastic IP address tags is not supported.

You can tag an Elastic IP address using one of the following methods.

New console

**To tag an Elastic IP address**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to tag and choose **Actions, View details**.
4. In the **Tags** section, choose **Manage tags**.
5. Specify a tag key and value pair.
6. (Optional) Choose **Add tag** to add additional tags.
7. Choose **Save**.

Old console

**To tag an Elastic IP address**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to tag and choose **Tags**.
4. Choose **Add/Edit Tags**.
5. In the **Add/Edit Tags** dialog box, choose **Create Tag**, and then specify the key and value for the tag.
6. (Optional) Choose **Create Tag** to add additional tags to the Elastic IP address.
7. Choose **Save**.

**AWS CLI**

**To tag an Elastic IP address**

Use the `create-tags` AWS CLI command.

```bash
aws ec2 create-tags --resources eipalloc-12345678 --tags Key=Owner,Value=TeamA
```

**PowerShell**

**To tag an Elastic IP address**

Use the `New-EC2Tag` AWS Tools for Windows PowerShell command.

The `New-EC2Tag` command needs a `Tag` parameter, which specifies the key and value pair to be used for the Elastic IP address tag. The following commands create the `Tag` parameter.

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag
PS C:\> $tag.Key = "Owner"
PS C:\> $tag.Value = "TeamA"

PS C:\> New-EC2Tag -Resource eipalloc-12345678 -Tag $tag
```

**Associate an Elastic IP address with an instance or network interface**

If you’re associating an Elastic IP address with your instance to enable communication with the internet, you must also ensure that your instance is in a public subnet. For more information, see **Internet Gateways** in the Amazon VPC User Guide.

You can associate an Elastic IP address with an instance or network interface using one of the following methods.

**New console**

**To associate an Elastic IP address with an instance**

1. Open the Amazon EC2 console at `https://console.aws.amazon.com/ec2/`.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to associate and choose **Actions, Associate Elastic IP address**.
4. For **Resource type**, choose **Instance**.
5. For instance, choose the instance with which to associate the Elastic IP address. You can also enter text to search for a specific instance.
To work with Elastic IP addresses

6. (Optional) For **Private IP address**, specify a private IP address with which to associate the Elastic IP address.
7. Choose **Associate**.

**To associate an Elastic IP address with a network interface**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to associate and choose **Actions, Associate Elastic IP address**.
4. For **Resource type**, choose **Network interface**.
5. For **Network interface**, choose the network interface with which to associate the Elastic IP address. You can also enter text to search for a specific network interface.
6. (Optional) For **Private IP address**, specify a private IP address with which to associate the Elastic IP address.
7. Choose **Associate**.

**Old console**

To associate an Elastic IP address with an instance

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Elastic IPs**.
3. Select an Elastic IP address and choose **Actions, Associate address**.
4. Select the instance from **Instance** and then choose **Associate**.

**AWS CLI**

To associate an Elastic IP address

Use the `associate-address` AWS CLI command.

**PowerShell**

To associate an Elastic IP address

Use the `Register-EC2Address` AWS Tools for Windows PowerShell command.

**Disassociate an Elastic IP address**

You can disassociate an Elastic IP address from an instance or network interface at any time. After you disassociate the Elastic IP address, you can reassociate it with another resource.

You can disassociate an Elastic IP address using one of the following methods.

**New console**

To **disassociate and reassociate an Elastic IP address**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to disassociate, choose **Actions, Disassociate Elastic IP address**.
4. Choose **Disassociate**.
Work with Elastic IP addresses

Old console

To disassociate and reassociate an Elastic IP address

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Elastic IPs.
3. Select the Elastic IP address, choose Actions, and then select Disassociate address.
4. Choose Disassociate address.

AWS CLI

To disassociate an Elastic IP address

Use the disassociate-address AWS CLI command.

PowerShell

To disassociate an Elastic IP address

Use the Unregister-EC2Address AWS Tools for Windows PowerShell command.

Release an Elastic IP address

If you no longer need an Elastic IP address, we recommend that you release it using one of the following methods. The address to release must not be currently associated with an AWS resource, such as an EC2 instance, NAT gateway, or Network Load Balancer.

New console

To release an Elastic IP address

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Elastic IPs.
3. Select the Elastic IP address to release and choose Actions, Release Elastic IP addresses.

Old console

To release an Elastic IP address

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Elastic IPs.
3. Select the Elastic IP address, choose Actions, and then select Release addresses. Choose Release when prompted.

AWS CLI

To release an Elastic IP address

Use the release-address AWS CLI command.

PowerShell

To release an Elastic IP address
Use the Remove-EC2Address AWS Tools for Windows PowerShell command.

**Recover 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.

**AWS CLI**

**To recover an Elastic IP address**

Use the `allocate-address` AWS CLI command and specify the IP address using the `--address` parameter as follows.

```
aws ec2 allocate-address --domain vpc --address 203.0.113.3
```

**PowerShell**

**To recover an Elastic IP address**

Use the `New-EC2Address` AWS Tools for Windows PowerShell command and specify the IP address using the `-Address` parameter as follows.

```
PS C:\> New-EC2Address -Address 203.0.113.3 -Domain vpc -Region us-east-1
```

**Use reverse DNS for email applications**

If you intend to send email to third parties from an instance, we recommend that you provision one or more Elastic IP addresses and assign static reverse DNS records to the Elastic IP addresses that you use to send email. This can help you avoid having your email flagged as spam by some anti-spam organizations.

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.

**Note**

Before you create a reverse DNS record, you must set a corresponding forward DNS record (record type A) that points to your Elastic IP address.

For the US East (Ohio), Africa (Cape Town), Asia Pacific (Mumbai), Canada (Central), and Europe (Milan) Regions, use one of the following options.

**Console**

**To create a reverse DNS record for your Elastic IP address**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose Elastic IPs from the navigation pane.
3. Select the Elastic IP address and choose Actions, Update reverse DNS.
4. For Reverse DNS domain name, enter the domain name to associate with the Elastic IP address.
5. Enter `update` to confirm.
6. Choose `Update`.

**AWS CLI**

**To create a reverse DNS record for your Elastic IP address**

- Use the `modify-address-attribute` AWS CLI command to associate your domain name to your Elastic IP address.

For all other Regions, you must provide your Elastic IP addresses to AWS to assign static reverse DNS records for you. 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. To provide us with your Elastic IP addresses and reverse DNS records, go to the Request to remove reverse DNS and email sending limitations page.

**Elastic IP address limit**

By default, all AWS accounts are limited to five (5) Elastic IP addresses per Region, because public (IPv4) internet addresses are a scarce public resource. We strongly encourage you to use an Elastic IP address primarily for the ability to remap the address to another instance in the case of instance failure, and to use DNS hostnames for all other inter-node communication.

**To verify how many Elastic IP addresses are in use**

Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/) and choose Elastic IPs from the navigation pane.

**To verify your current account limit for Elastic IP addresses**

You can verify your limit in either the Amazon EC2 console or the Service Quotas console. Do one of the following:

- Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
  
  Choose Limits from the navigation pane, and then enter IP in the search field. The limit is EC2-VPC Elastic IPs. If you have access to EC2-Classic, there is an additional limit, EC2-Classic Elastic IPs.

- Open the Service Quotas console at [https://console.aws.amazon.com/servicequotas/](https://console.aws.amazon.com/servicequotas/).
  
  On the Dashboard, choose Amazon Elastic Compute Cloud (Amazon EC2). If Amazon Elastic Compute Cloud (Amazon EC2) is not listed on the Dashboard, choose AWS services, enter EC2 in the search field, and then choose Amazon Elastic Compute Cloud (Amazon EC2).

  On the Amazon EC2 service quotas page, enter IP in the search field. The limit is EC2-VPC Elastic IPs. If you have access to EC2-Classic, there is an additional limit, EC2-Classic Elastic IPs. For more information, choose the limit.

If you think your architecture warrants additional Elastic IP addresses, you can request a quota increase directly from the Service Quotas console.

**Elastic network interfaces**

An elastic network interface is a logical networking component in a VPC that represents a virtual network card. It can include the following attributes:
Network interface basics

You can create and configure network interfaces and attach them to instances in the same Availability Zone. 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. 958).

This AWS resource is referred to as a network interface in the AWS Management Console and the Amazon EC2 API. Therefore, we use "network interface" in this documentation instead of "elastic network interface". The term "network interface" in this documentation always means "elastic network interface".

Contents
- Network interface basics (p. 935)
- IP addresses per network interface per instance type (p. 936)
- Work with network interfaces (p. 948)
- Scenarios for network interfaces (p. 955)
- Best practices for configuring network interfaces (p. 957)
- Requester-managed network interfaces (p. 958)

Network interface basics

You can create a network interface, attach it to an instance, detach it from an instance, and attach it to another instance. The attributes of a network interface follow it as it's attached or detached from an instance and reattached to another instance. When you move a network interface from one instance to another, network traffic is redirected to the new instance.

Primary network interface

Each instance has a default network interface, called the primary network interface. You cannot detach a primary network interface from an instance. You can create and attach additional network interfaces. The maximum number of network interfaces that you can use varies by instance type. For more information, see IP addresses per network interface per instance type (p. 936).

Public IPv4 addresses for network interfaces

In a VPC, all subnets have a modifiable attribute that determines whether network interfaces created in that subnet (and therefore instances launched into that subnet) are assigned a public IPv4 address. For more information, see IP addressing behavior for your subnet in the Amazon VPC User Guide. The public IPv4 address is assigned from Amazon's pool of public IPv4 addresses. When you launch an instance, the IP address is assigned to the primary network interface that's created.

When you create a network interface, it inherits the public IPv4 addressing attribute from the subnet. If you later modify the public IPv4 addressing attribute of the subnet, the network interface keeps the setting that was in effect when it was created. If you launch an instance and specify an existing network
interface as the primary network interface, the public IPv4 address attribute is determined by this network interface.

For more information, see Public IPv4 addresses and external DNS hostnames (p. 894).

**Elastic IP addresses for network interface**

If you have an Elastic IP address, you can associate it with one of the private IPv4 addresses for the network interface. You can associate one Elastic IP address with each private IPv4 address.

If you disassociate an Elastic IP address from a network interface, you can release it back to the address pool. This is the only way to associate an Elastic IP address with an instance in a different subnet or VPC, as network interfaces are specific to subnets.

**IPv6 addresses for network interfaces**

If you associate IPv6 CIDR blocks with your VPC and subnet, you can assign one or more IPv6 addresses from the subnet range to a network interface. Each IPv6 address can be assigned to one network interface.

All subnets have a modifiable attribute that determines whether network interfaces created in that subnet (and therefore instances launched into that subnet) are automatically assigned an IPv6 address from the range of the subnet. For more information, see IP addressing behavior for your subnet in the Amazon VPC User Guide. When you launch an instance, the IPv6 address is assigned to the primary network interface that's created.

For more information, see IPv6 addresses (p. 895).

**Prefix Delegation**

A Prefix Delegation prefix is a reserved private IPv4 or IPv6 CIDR range that you allocate for automatic or manual assignment to network interfaces that are associated with an instance. By using Delegated Prefixes, you can launch services faster by assigning a range of IP addresses as a single prefix.

**Termination behavior**

You can set the termination behavior for a network interface that's attached to an instance. You can specify whether the network interface should be automatically deleted when you terminate the instance to which it's attached.

**Source/destination checking**

You can enable or disable source/destination checks, which ensure that the instance is either the source or the destination of any traffic that it receives. Source/destination checks are enabled by default. You must disable source/destination checks if the instance runs services such as network address translation, routing, or firewalls.

**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.
<table>
<thead>
<tr>
<th>Instance type</th>
<th>Maximum network interfaces</th>
<th>Private IPv4 addresses per interface</th>
<th>IPv6 addresses per interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1.medium</td>
<td>2</td>
<td>6</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>c1.xlarge</td>
<td>4</td>
<td>15</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>c3.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>c3.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>c3.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>c3.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c3.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c4.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>c4.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>c4.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>c4.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c4.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>c5.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>c5.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>c5.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5.9xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5.18xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>c5.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>c5.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>c5a.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>c5a.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>c5a.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>c5a.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5a.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5a.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5a.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>c5a.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>c5ad.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Instance type</td>
<td>Maximum network interfaces</td>
<td>Private IPv4 addresses per interface</td>
<td>IPv6 addresses per interface</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------------------</td>
<td>------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>c5ad.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>c5ad.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>c5ad.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5ad.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5ad.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5ad.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>c5ad.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>c5d.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>c5d.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>c5d.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>c5d.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5d.9xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5d.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5d.18xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>c5d.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>c5d.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>c5n.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>c5n.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>c5n.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>c5n.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5n.9xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>c5n.18xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>c5n.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>cc2.8xlarge</td>
<td>8</td>
<td>30</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>cr1.8xlarge</td>
<td>8</td>
<td>30</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>d2.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>d2.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>d2.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>d2.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>d3.xlarge</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Instance type</td>
<td>Maximum network interfaces</td>
<td>Private IPv4 addresses per interface</td>
<td>IPv6 addresses per interface</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------</td>
<td>-------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>d3.2xlarge</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>d3.4xlarge</td>
<td>4</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>d3.8xlarge</td>
<td>3</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>d3en.large</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>d3en.xlarge</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>d3en.2xlarge</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>d3en.4xlarge</td>
<td>4</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>d3en.6xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>d3en.8xlarge</td>
<td>4</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>d3en.12xlarge</td>
<td>3</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>f1.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>f1.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>f1.16xlarge</td>
<td>8</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>g2.2xlarge</td>
<td>4</td>
<td>15</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>g2.8xlarge</td>
<td>8</td>
<td>30</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>g3s.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>g3.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>g3.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>g3.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>g4ad.xlarge</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>g4ad.2xlarge</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>g4ad.4xlarge</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>g4ad.8xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>g4ad.16xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>g4dn.xlarge</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>g4dn.2xlarge</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>g4dn.4xlarge</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>g4dn.8xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>g4dn.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>g4dn.16xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Instance type</td>
<td>Maximum network interfaces</td>
<td>Private IPv4 addresses per interface</td>
<td>IPv6 addresses per interface</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------</td>
<td>--------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>g4dn.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>h1.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>h1.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>h1.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>h1.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>hs1.8xlarge</td>
<td>8</td>
<td>30</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>i2.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>i2.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>i2.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>i2.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>i3.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>i3.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>i3.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>i3.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>i3.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>i3.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>i3.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>i3en.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>i3en.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>i3en.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>i3en.3xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>i3en.6xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>i3en.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>i3en.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>i3en.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m1.small</td>
<td>2</td>
<td>4</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>m1.medium</td>
<td>2</td>
<td>6</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>m1.large</td>
<td>3</td>
<td>10</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>m1.xlarge</td>
<td>4</td>
<td>15</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>m2.xlarge</td>
<td>4</td>
<td>15</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>Instance type</td>
<td>Maximum network interfaces</td>
<td>Private IPv4 addresses per interface</td>
<td>IPv6 addresses per interface</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------</td>
<td>-------------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>m2.2xlarge</td>
<td>4</td>
<td>30</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>m2.4xlarge</td>
<td>8</td>
<td>30</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>m3.medium</td>
<td>2</td>
<td>6</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>m3.large</td>
<td>3</td>
<td>10</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>m3.xlarge</td>
<td>4</td>
<td>15</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>m3.2xlarge</td>
<td>4</td>
<td>30</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>m4.large</td>
<td>2</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>m4.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m4.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m4.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m4.10xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m4.16xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>m5.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m5.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m5.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5a.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>m5a.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m5a.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m5a.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5a.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5a.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5a.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5a.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5ad.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Instance type</td>
<td>Maximum network interfaces</td>
<td>Private IPv4 addresses per interface</td>
<td>IPv6 addresses per interface</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------</td>
<td>-------------------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>m5ad.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m5ad.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m5ad.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5ad.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5ad.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5ad.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5ad.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5d.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>m5d.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m5d.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m5d.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5d.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5d.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5d.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5d.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5d.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5dn.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>m5dn.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m5dn.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m5dn.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5dn.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5dn.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5dn.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5dn.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5dn.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5n.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>m5n.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m5n.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m5n.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5n.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Instance type</td>
<td>Maximum network interfaces</td>
<td>Private IPv4 addresses per interface</td>
<td>IPv6 addresses per interface</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------</td>
<td>-------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>m5n.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5n.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5n.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5n.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5zn.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>m5zn.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m5zn.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m5zn.3xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5zn.6xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m5zn.12xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m5zn.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m6i.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>m6i.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m6i.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>m6i.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m6i.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m6i.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>m6i.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m6i.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>m6i.32xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>p2.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>p2.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>p2.16xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>p3.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>p3.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>p3.16xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>p3dn.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r3.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>r3.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r3.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Instance type</td>
<td>Maximum network interfaces</td>
<td>Private IPv4 addresses per interface</td>
<td>IPv6 addresses per interface</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------</td>
<td>-------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>r3.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r3.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r4.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>r4.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r4.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r4.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r4.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r4.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>r5.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r5.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r5.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5a.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>r5a.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r5a.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r5a.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5a.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5a.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5a.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5a.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5ad.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>r5ad.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r5ad.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r5ad.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5ad.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Instance type</td>
<td>Maximum network interfaces</td>
<td>Private IPv4 addresses per interface</td>
<td>IPv6 addresses per interface</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------</td>
<td>-------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>r5ad.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5ad.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5ad.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5b.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>r5b.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r5b.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r5b.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5b.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5b.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5b.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5b.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5b.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5d.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>r5d.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r5d.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r5d.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5d.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5d.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5d.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5d.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5d.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5dn.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>r5dn.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r5dn.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r5dn.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5dn.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5dn.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5dn.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5dn.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5dn.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Instance type</td>
<td>Maximum network interfaces</td>
<td>Private IPv4 addresses per interface</td>
<td>IPv6 addresses per interface</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------</td>
<td>--------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>r5n.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>r5n.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r5n.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>r5n.4xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5n.8xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5n.12xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>r5n.16xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5n.24xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>r5n.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>t1.micro</td>
<td>2</td>
<td>2</td>
<td>IPv6 not supported</td>
</tr>
<tr>
<td>t2.nano</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>t2.micro</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>t2.small</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>t2.medium</td>
<td>3</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>t2.large</td>
<td>3</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>t2.xlarge</td>
<td>3</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>t2.2xlarge</td>
<td>3</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>t3.nano</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>t3.micro</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>t3.small</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>t3.medium</td>
<td>3</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>t3.large</td>
<td>3</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>t3.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>t3.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>t3a.nano</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>t3a.micro</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>t3a.small</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>t3a.medium</td>
<td>3</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>t3a.large</td>
<td>3</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>t3a.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>
### IP addresses per network interface per instance type

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Maximum network interfaces</th>
<th>Private IPv4 addresses per interface</th>
<th>IPv6 addresses per interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>t3a.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>u-6tb1.56xlarge</td>
<td>5</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>u-6tb1.112xlarge</td>
<td>5</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>u-6tb1.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>u-9tb1.112xlarge</td>
<td>5</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>u-9tb1.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>u-12tb1.112xlarge</td>
<td></td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>u-12tb1.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>u-18tb1.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>u-24tb1.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>x1.16xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>x1.32xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>x1e.xlarge</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>x1e.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>x1e.4xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>x1e.8xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>x1e.16xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>x1e.32xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>z1d.large</td>
<td>3</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>z1d.xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>z1d.2xlarge</td>
<td>4</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>z1d.3xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>z1d.6xlarge</td>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>z1d.12xlarge</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>z1d.metal</td>
<td>15</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

You can use the `describe-instance-types` AWS CLI command to display information about an instance type, such as the supported network interfaces and IP addresses per interface. The following example displays this information for all CS instances.

```
aws ec2 describe-instance-types --filters "Name=instance-type,Values=cs.*" --query "InstancesTypes[].{Type: InstanceType, MaxENI: NetworkInfo.MaximumNetworkInterfaces, IPv4addr: NetworkInfo.Ipv4AddressesPerInterface}" --output table
```

| DescribeInstanceTypes | |
Work with network interfaces

You can work with network interfaces using the Amazon EC2 console or the command line.

Contents
- Create a network interface (p. 948)
- View details about a network interface (p. 949)
- Attach a network interface to an instance (p. 950)
- Detach a network interface from an instance (p. 951)
- Manage IP addresses (p. 952)
- Modify network interface attributes (p. 953)
- Add or edit tags (p. 954)
- Delete a network interface (p. 955)

Create 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. You must attach a network interface to an instance in the same Availability Zone.

New console

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. (Optional) For Description, enter a descriptive name.
5. For Subnet, select a subnet.
6. For Private IPv4 address, do one of the following:
   - Choose Auto-assign to allow Amazon EC2 to select an IPv4 address from the subnet.
   - Choose Custom and enter an IPv4 address that you select from the subnet.
7. (Subnets with IPv6 addresses only) For IPv6 address, do one of the following:
   - Choose None if you do not want to assign an IPv6 address to the network interface.
   - Choose Auto-assign to allow Amazon EC2 to select an IPv6 address from the subnet.
   - Choose Custom and enter an IPv6 address that you select from the subnet.
8. (Optional) To create an Elastic Fabric Adapter, choose Elastic Fabric Adapter, Enable.
9. For **Security groups**, select one or more security groups.
10. (Optional) For each tag, choose **Add new tag** and enter a tag key and an optional tag value.
11. Choose **Create network interface**.

**Old console**

**To create a network interface using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Network Interfaces**.
3. Choose **Create Network Interface**.
4. For **Description**, enter a descriptive name.
5. For **Subnet**, select the subnet.
6. For **Private IP** (or **IPv4 Private IP**), enter the primary private IPv4 address. If you don't specify an IPv4 address, we select an available private IPv4 address from within the selected subnet.
7. (IPv6 only) If you selected a subnet that has an associated IPv6 CIDR block, you can optionally specify an IPv6 address in the **IPv6 IP** field.
8. To create an Elastic Fabric Adapter, select **Elastic Fabric Adapter**.
9. For **Security groups**, select one or more security groups.
10. (Optional) Choose **Add Tag** and enter a tag key and a tag value.
11. Choose **Yes, Create**.

**To create a network interface using the command line**

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

- **create-network-interface** (AWS CLI)
- **New-EC2NetworkInterface** (AWS Tools for Windows PowerShell)

**View details about a network interface**

You can view all the network interfaces in your account.

**New console**

**To describe a network interface using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Network Interfaces**.
3. To view the details page for a network interface, select the ID of the network interface. Alternatively, to view information without leaving the network interfaces page, select the checkbox for the network interface.

**Old console**

**To describe a network interface using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](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 *Access 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 *Access Amazon EC2* (p. 3).

- `describe-network-interface-attribute` (AWS CLI)

**Attach a network interface to an instance**

You can attach a network interface to any instance in the same Availability Zone as the network interface, using either the **Instances** or **Network Interfaces** page of the Amazon EC2 console. Alternatively, you can specify existing network interfaces when you launch instances (p. 392).

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. 894).

**Instances page**

**To attach a network interface to an instance using the Instances page**

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

2. In the navigation pane, choose **Instances**.

3. Select the checkbox for the instance.

4. Choose **Actions**, **Networking**, **Attach network interface**.

5. Select a network interface. If the instance supports multiple network cards, you can choose a network card.

6. Choose **Attach**.

**Network Interfaces page**

**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 checkbox for the network interface.

4. Choose **Actions, Attach**.
5. Choose an instance. If the instance supports multiple network cards, you can choose a network card.

6. 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 Access Amazon EC2 (p. 3).

- attach-network-interface (AWS CLI)
- Add-EC2NetworkInterface (AWS Tools for Windows PowerShell)

Detach a network interface from an instance

You can detach a secondary network interface that is attached to an EC2 instance at any time, using either the Instances or Network Interfaces page of the Amazon EC2 console.

If you try to detach a network interface that is attached to a resource from another service, such as an Elastic Load Balancing load balancer, a Lambda function, a WorkSpace, or a NAT gateway, you get an error that you do not have permission to access the resource. To find which service created the resource attached to a network interface, check the description of the network interface. If you delete the resource, then its network interface is deleted.

Instances page

To detach a network interface from an instance using the Instances page

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the checkbox for the instance. Check the Network interfaces section of the Networking tab to verify that the network interface is attached to an instance as a secondary network interface.
5. Select the network interface and choose Detach.

Network Interfaces page

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 checkbox for the network interface. Check the Instance details section of the Details tab to verify that the network interface is attached to an instance as a secondary network interface.
4. Choose Actions, Detach.
5. When prompted for confirmation, choose Detach.
6. If the network interface fails to detach from the instance, choose Force detachment, Enable and then try again. We recommend that force detachment only as a last resort. Forcing a detachment can prevent you from attaching a different network interface on the same index until you restart the instance. It can also prevent the instance metadata from reflecting that the network interface was detached until you restart the instance.
To detach a network interface using the command line

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

- detach-network-interface (AWS CLI)
- Dismount-EC2NetworkInterface (AWS Tools for Windows PowerShell)

Manage IP addresses

You can manage the following IP addresses for your network interfaces:

- Elastic IP addresses (one per private IPv4 address)
- IPv4 addresses
- IPv6 addresses

To manage the Elastic IP addresses of a network interface using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Network Interfaces.
3. Select the checkbox for the network interface.
4. To associate an Elastic IP address, do the following:
   a. Choose Actions, Associate address.
   b. For Elastic IP address, select the Elastic IP address.
   c. For Private IPv4 address, select the private IPv4 address to associate with the Elastic IP address.
   d. (Optional) Choose Allow the Elastic IP address to be reassigned if the network interface is currently associated with another instance or network interface.
   e. Choose Associate.
5. To disassociate an Elastic IP address, do the following:
   a. Choose Actions, Disassociate address.
   b. For Public IP address, select the Elastic IP address.
   c. Choose Disassociate.

To manage the IPv4 and IPv6 addresses of a network interface using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Network Interfaces.
3. Select the network interface.
4. Choose Actions, Manage IP addresses.
5. Expand the network interface.
6. For IPv4 addresses, modify the IP addresses as needed. To assign an IPv4 address, choose Assign new IP address and then specify an IPv4 address from the subnet range or let AWS choose one for you. To unassign an IPv4 address, choose Unassign next to the address.
7. For IPv6 addresses, modify the IP addresses as needed. To assign an IPv6 address, choose Assign new IP address and then specify an IPv6 address from the subnet range or let AWS choose one for you. To unassign an IPv6 address, choose Unassign next to the address.
8. Choose Save.
To manage the IP addresses of a network interface using the AWS CLI

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

- assign-ipv6-addresses
- associate-address
- disassociate-address
- unassign-ipv6-addresses

To manage the IP addresses of a network interface using the Tools for Windows PowerShell

You can use one of the following commands.

- Register-EC2Address
- Register-EC2Ipv6AddressList
- Unregister-EC2Address
- Unregister-EC2Ipv6AddressList

Modify network interface attributes

You can change the following network interface attributes:

- Description (p. 953)
- Security groups (p. 953)
- Delete on termination (p. 954)
- Source/destination check (p. 954)

To change the description of a network interface using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Network Interfaces.
3. Select the checkbox for the network interface.
5. For Description, enter a description for the network interface.
6. Choose Save.

To change the security groups of a network interface using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Network Interfaces.
3. Select the checkbox for the network interface.
4. Choose Actions, Change security groups.
5. For Associated security groups, select the security groups to use, and then choose Save.

The security group and network interface must be created for the same VPC. To change the security group for interfaces owned by other services, such as Elastic Load Balancing, do so through that service.
To change the termination behavior of a network interface using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Network Interfaces.
3. Select the checkbox for the network interface.
4. Choose Actions, Change termination behavior.
5. Select or clear Delete on termination, Enable as needed, and then choose Save.

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 checkbox for the network interface.
5. Select or clear Source/destination check, Enable as needed, and then choose Save.

To modify network interface attributes using the command line

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

- modify-network-interface-attribute (AWS CLI)
- Edit-EC2NetworkInterfaceAttribute (AWS Tools for Windows PowerShell)

Add or edit 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 Tag your Amazon EC2 resources (p. 1450).

New console

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 checkbox for the network interface.
4. In Tags tab, choose Manage tags.
5. For each tag to create, choose Add new tag and enter a key and optional value. When you're done, choose Save.

Old console

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 [Access Amazon EC2 (p. 3)](#).

- **create-tags** (AWS CLI)
- **New-EC2Tag** (AWS Tools for Windows PowerShell)

### Delete a 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.

You cannot delete a network interface that is in use. First, you must [detach the network interface (p. 951)](#).

#### New console

**To delete a network interface using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Network Interfaces**.
3. Select the checkbox for the network interface, and then choose **Actions, Delete**.
4. When prompted for confirmation, choose **Delete**.

#### Old console

**To delete a network interface using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](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 [Access Amazon EC2 (p. 3)](#).

- **delete-network-interface** (AWS CLI)
- **Remove-EC2NetworkInterface** (AWS Tools for Windows PowerShell)

### Scenarios for network interfaces

Attaching multiple network interfaces to an instance is useful when you want to:

- Create a management network.
- Use network and security appliances in your VPC.
• Create dual-homed instances with workloads/roles on distinct subnets.
• Create a low-budget, high-availability solution.

Create a management network

You can create a management network using network interfaces. In this scenario, as illustrated in the following image:

• The primary network interface (eth0) on the instance handles public traffic.
• The secondary network interface (eth1) handles backend management traffic, and is connected to a separate subnet in your VPC that has more restrictive access controls.

The public interface, which may or may not be behind a load balancer, has an associated security group that allows access to the server from the internet (for example, allow TCP port 80 and 443 from 0.0.0.0/0, or from the load balancer).

The private facing interface has an associated security group allowing 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 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 route table or your DNS server are required.

Best practices for configuring network interfaces

- You can attach a network interface to an instance when it's running (hot attach), when it's stopped (warm attach), or when the instance is being launched (cold attach).
- You can detach secondary network interfaces when the instance is running or stopped. However, you can't detach the primary network interface.
- You can move a network interface from one instance to another, if the instances are in the same Availability Zone and VPC but in different subnets.
- When launching an instance using the CLI, API, or an SDK, you can specify the primary network interface and additional network interfaces.
- Launching an Amazon Linux or Windows Server instance with multiple network interfaces automatically configures interfaces, private IPv4 addresses, and route tables on the operating system of the instance.
- A warm or hot attach of an additional network interface might 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.
- You cannot attach another network interface to an instance (for example, a NIC teaming configuration) to increase or double the network bandwidth to or from the dual-homed instance.
- If you attach two or more network interfaces from the same subnet to an instance, you might encounter networking issues such as asymmetric routing. If possible, use a secondary private IPv4 address on the primary network interface instead. If you need to use multiple network interfaces, you must configure the network interfaces to use static routing.
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 Add or edit tags (p. 954).

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.

   ```bash
   aws ec2 describe-network-interfaces
   ```

2. In the output, the RequesterManaged field displays true if the network interface is managed by another AWS service.

   ```json
   {  
     "Status": "in-use",
     ...
     "Description": "VPC Endpoint Interface vpce-089f2123488812123",
     "NetworkInterfaceId": "eni-c8fbc27e",
     "VpcId": "vpc-1a2b3c4d",
     "PrivateIpAddress": [
       {  
         "PrivateDnsName": "ip-10-0-2-227.ec2.internal",
         "Primary": true,
         "PrivateIpAddress": "10.0.2.227"
       }
     ],
     "RequesterManaged": true,
     ...
   }
   ```

Amazon EC2 instance network bandwidth

The network bandwidth available to an EC2 instance depends on the destination of the traffic. The following bandwidth quotas apply to aggregate multi-flow bandwidth on a per-instance basis.

**Inside the Region**

Traffic between an instance and a target destination in the same Region can utilize the full network bandwidth available to the instance.

**Outside the Region**

Traffic between an instance and a target destination has an aggregate bandwidth quota of 5 Gbps if the target is not in the same Region as the instance, including traffic to AWS Direct Connect or the internet.

Single flow (5-tuple) bandwidth is limited to 5 Gbps, regardless of the direction of traffic. For use cases that require low latency and high single flow bandwidth, use a cluster placement group (p. 976) to achieve bandwidth up to 10 Gbps for instances in the same placement group. Alternatively, set up multiple paths between any two endpoints to achieve higher bandwidth using Multipath TCP (MPTCP).

Available instance bandwidth

The available network bandwidth of an instance depends on the number of vCPUs that it has. For example, an m5.8xlarge instance has 32 vCPUs and 10 Gbps network bandwidth, and an m5.16xlarge instance has 64 vCPUs and 20 Gbps network bandwidth. However, instances might not achieve this bandwidth; for example, if they exceed network allowances at the instance level, such as packet per second or number of tracked connections.

Typically, instances with 16 vCPUs or fewer (size 4xlarge and smaller) are documented as having "up to" a specified bandwidth; for example, "up to 10 Gbps". These instances have a baseline bandwidth. To meet additional demand, they can use a network I/O credit mechanism to burst beyond their baseline bandwidth. Instances can use burst bandwidth for a limited time, typically from 5 to 60 minutes, depending on the instance size.

An instance receives the maximum number of network I/O credits at launch. If the instance exhausts its network I/O credits, it returns to its baseline bandwidth. A running instance earns network I/O credits whenever it uses less network bandwidth than its baseline bandwidth. A stopped instance does not earn network I/O credits. Instance burst is on a best effort basis, even when the instance has credits available, as burst bandwidth is a shared resource.

The following documentation describes the network performance for all instances, plus the baseline network bandwidth available for instances that can use burst bandwidth.

- General purpose instances (p. 154)
- Compute optimized instances (p. 193)
- Memory optimized instances (p. 199)
- Storage optimized instances (p. 210)
- Accelerated computing instances (p. 219)

**To view network performance using the AWS CLI**

You can use the describe-instance-types AWS CLI command to display information about an instance type. The following example displays network performance information for all C5 instances.
Monitor instance bandwidth

You can use CloudWatch metrics to monitor instance network bandwidth and the packets sent and received. You can use the network performance metrics provided by the Elastic Network Adapter (ENA) driver to monitor when traffic exceeds the network allowances that Amazon EC2 defines at the instance level.

You can configure whether Amazon EC2 sends metric data for the instance to CloudWatch using one-minute periods or five-minute periods. It is possible that the network performance metrics would show that an allowance was exceeded and packets were dropped while the CloudWatch instance metrics do not. This can happen when the instance has a short spike in demand for network resources (known as a microburst), but the CloudWatch metrics are not granular enough to reflect these microsecond spikes.

Learn more
- Instance metrics (p. 842)
- Network performance metrics (p. 973)

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. 961). SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization when compared to traditional virtualized network interfaces. Enhanced networking provides higher bandwidth, higher packet per second (PPS) performance, and consistently lower inter-instance latencies. There is no additional charge for using enhanced networking.

For information about the supported network speed for each instance type, see Amazon EC2 Instance Types.

Contents
- Enhanced networking support (p. 961)
- Enable enhanced networking on your instance (p. 961)
- Enable enhanced networking with the Elastic Network Adapter (ENA) on Windows instances (p. 961)
- Enable enhanced networking with the Intel 82599 VF interface on Windows instances (p. 968)
- Operating system optimizations (p. 972)
- Monitor network performance for your EC2 instance (p. 973)
Enhanced networking support

All current generation (p. 142) instance types support enhanced networking, except for T2 instances.

You can enable enhanced networking using one of the following mechanisms:

**Elastic Network Adapter (ENA)**

The Elastic Network Adapter (ENA) supports network speeds of up to 100 Gbps for supported instance types.

The current generation instances use ENA for enhanced networking, except for C4, D2, and M4 instances smaller than m4.16xlarge.

**Intel 82599 Virtual Function (VF) interface**

The Intel 82599 Virtual Function interface supports network speeds of up to 10 Gbps for supported instance types.

The following instance types use the Intel 82599 VF interface for enhanced networking: C3, C4, D2, I2, M4 (excluding m4.16xlarge), and R3.

For a summary of the enhanced networking mechanisms by instance type, see Summary of networking and storage features (p. 148).

Enable enhanced networking on your instance

If your instance type supports the Elastic Network Adapter for enhanced networking, follow the procedures in Enable enhanced networking with the Elastic Network Adapter (ENA) on Windows instances (p. 961).

If your instance type supports the Intel 82599 VF interface for enhanced networking, follow the procedures in Enable enhanced networking with the Intel 82599 VF interface on Windows instances (p. 968).

Enable enhanced networking with the Elastic Network Adapter (ENA) on Windows instances

Amazon EC2 provides enhanced networking capabilities through the Elastic Network Adapter (ENA). To use enhanced networking, you must install the required ENA module and enable ENA support.

Contents

- Requirements (p. 961)
- Enhanced networking performance (p. 962)
- Test whether enhanced networking is enabled (p. 962)
- Enable enhanced networking on Windows (p. 963)
- Amazon ENA driver versions (p. 964)
- Subscribe to notifications (p. 523)

Requirements

To prepare for enhanced networking using the ENA, set up your instance as follows:

- Launch the instance using a current generation (p. 142) instance type, other than C4, D2, M4 instances smaller than m4.16xlarge, or T2.
• If the instance is running Windows Server 2008 R2 SP1, ensure that it has the SHA-2 code signing support update.
• Ensure that the instance has internet connectivity.
• Use AWS CloudShell from the AWS Management Console, or 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 Access Amazon EC2 (p. 3) or the AWS CloudShell User Guide. Enhanced networking cannot be managed from the Amazon EC2 console.
• If you have important data on the instance that you want to preserve, you should back that data up now by creating an AMI from your instance. Updating kernels and kernel modules, as well as enabling the enaSupport attribute, might render incompatible instances or operating systems unreachable. If you have a recent backup, your data will still be retained if this happens.

Enhanced networking performance

The following documentation provides a summary of the network performance for the instance types that support ENA enhanced networking:

• Network Performance for Accelerated Computing Instances (p. 219)
• Network Performance for Compute Optimized Instances (p. 195)
• Network Performance for General Purpose Instances (p. 154)
• Network Performance for Memory Optimized Instances (p. 204)
• Network Performance for Storage Optimized Instances (p. 213)

Test 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 CloudShell)

```bash
aws ec2 describe-instances --instance-ids instance_id --query "Reservations[].Instances[].EnaSupport"
```

• Get-EC2Instance (Tools for Windows PowerShell)

```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 CloudShell)

```bash
aws ec2 describe-images --image-id ami_id --query "Images[].EnaSupport"
```

• Get-EC2Image (Tools for Windows PowerShell)
Enable 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 support (p. 961).

To enable enhanced networking

1. Connect to your instance and log in as the local administrator.
2. [Windows Server 2016 and later only] Run the following EC2Launch PowerShell script to configure the instance after the driver is installed.

   ```powershell
   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.

   **Note**
   If you get an execution policy error, set the policy to Unrestricted (by default it is set to Restricted or RemoteSigned). In a command line, run `Set-ExecutionPolicy -ExecutionPolicy Unrestricted`, and then run the `install.ps1` PowerShell script again.

4. From your local computer, stop the instance using the Amazon EC2 console or one of the following commands: `stop-instances` (AWS CLI/AWS CloudShell), `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.

      ```powershell
      aws ec2 describe-instances --instance-ids instance_id --query "Reservations[].Instances[].EnaSupport"
      ```

      ```powershell
      (Get-EC2Instance -InstanceId instance-id).Instances.EnaSupport
      ```

   b. To enable ENA support, run one of the following commands:

      ```powershell
      aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
      ```
• **Edit-EC2InstanceAttribute** (AWS Tools for Windows PowerShell)

  ```bash
  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 CloudShell)

  ```bash
  aws ec2 modify-instance-attribute --instance-id instance_id --no-ena-support
  ```

• **Edit-EC2InstanceAttribute** (AWS Tools for Windows PowerShell)

  ```bash
  Edit-EC2InstanceAttribute -InstanceId instance_id -EnaSupport $false
  ```

c. 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:

```json
[
  true
]
```

6. From your local computer, start the instance using the Amazon EC2 console or one of the following commands: **start-instances** (AWS CLI/AWS CloudShell), **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.

7. On the instance, validate that the ENA driver is installed and enabled as follows:

   a. Right-click the network icon and choose **Open Network and Sharing Center**.
   b. Choose the Ethernet adapter (for example, **Ethernet 2**).
   c. Choose **Details**. For **Network Connection Details**, check that **Description** is **Amazon Elastic Network Adapter**.

8. (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 Create a custom Windows AMI (p. 37).

### Amazon ENA driver versions

Windows AMIs include the Amazon ENA driver to enable enhanced networking. The following table summarizes the changes for each release.

<table>
<thead>
<tr>
<th>Driver version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.3</td>
<td>New Feature</td>
<td>March 25, 2021</td>
</tr>
<tr>
<td></td>
<td>• Adds support for new Nitro cards with up to 400 Gbps instance networking.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bug Fix</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Fixes race condition between system time change and system time query by the ENA driver, which causes false-positive detection of HW unresponsiveness.</td>
<td></td>
</tr>
<tr>
<td>Driver version</td>
<td>Details</td>
<td>Release date</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>2.2.2</td>
<td><strong>New Feature</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Adds support to query network adapter performance metrics with CloudWatch and the Performance Counters for Windows consumers.</td>
<td>December 21, 2020</td>
</tr>
<tr>
<td></td>
<td><strong>Bug Fix</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Fixes performance issues on bare metal instances.</td>
<td></td>
</tr>
<tr>
<td>2.2.1</td>
<td><strong>New Feature</strong></td>
<td>October 1, 2020</td>
</tr>
<tr>
<td></td>
<td>• Adds a method to allow the host to query the Elastic Network Adapter for network performance metrics.</td>
<td></td>
</tr>
<tr>
<td>2.2.0</td>
<td><strong>New Features</strong></td>
<td>August 12, 2020</td>
</tr>
<tr>
<td></td>
<td>• Adds support for next generation hardware types.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Improves instance start time after resuming from stop- hibernate, and eliminates false positive ENA error messages.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Performance Optimizations</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Optimizes processing of inbound traffic.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Improves shared memory management in low resource environment.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Bug Fix</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Avoids system crash upon ENA device removal in rare scenario where driver fails to reset.</td>
<td></td>
</tr>
<tr>
<td>2.1.5</td>
<td><strong>Bug Fix</strong></td>
<td>June 23, 2020</td>
</tr>
<tr>
<td></td>
<td>• Fixes occasional network adapter initialization failure on bare metal instances.</td>
<td></td>
</tr>
<tr>
<td>2.1.4</td>
<td><strong>Bug Fixes</strong></td>
<td>November 25, 2019</td>
</tr>
<tr>
<td></td>
<td>• Prevent connectivity issues caused by corrupted LSO packet metadata arriving from the network stack.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Prevent system crash caused by a rare race condition that results in accessing an already released packet memory.</td>
<td></td>
</tr>
<tr>
<td>Driver version</td>
<td>Details</td>
<td>Release date</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>2.1.2</td>
<td>New Feature</td>
<td>November 4, 2019</td>
</tr>
<tr>
<td></td>
<td>• Added support for vendor ID report to allow OS to generate MAC-based UUIDs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bug Fixes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Improved DHCP network configuration performance during initialization.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Properly calculate L4 checksum on inbound IPv6 traffic when the maximum transmission unit (MTU) exceeds 4K.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• General improvements to driver stability and minor bug fixes.</td>
<td></td>
</tr>
<tr>
<td>2.1.1</td>
<td>Bug Fixes</td>
<td>September 16, 2019</td>
</tr>
<tr>
<td></td>
<td>• Prevent drops of highly fragmented TCP LSO packets arriving from operating system.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Properly handle Encapsulating Security Payload (ESP) protocol within the IPSec in IPv6 networks.</td>
<td></td>
</tr>
</tbody>
</table>
### Driver version | Details | Release date
--- | --- | ---
2.1.0 | ENA Windows driver v2.1 introduces new ENA device capabilities, provides a performance boost, adds new features, and includes multiple stability improvements.  
- New features  
  - Use standardized Windows registry key for Jumbo frames configuration.  
  - Allow VLAN ID setting via the ENA driver properties GUI.  
  - Improved Recovery flows  
    - Improved failure identification mechanism.  
    - Added support for tunable recovery parameters.  
    - Support up to 32 I/O queues for newer EC2 instances that have more than 8 vCPUs.  
  - ~90% reduction of driver memory footprint.  
- Performance optimizations  
  - Reduced transmit path latency.  
  - Support for receive checksum offload.  
  - Performance optimization for heavily loaded system (optimized usage of locking mechanisms).  
  - Further enhancements to reduce CPU utilization and improve system responsiveness under load.  
- Bug Fixes  
  - Fix crash due to invalid parsing of non-contiguous Tx headers.  
  - Fix driver v1.5 crash during ENI detach on Bare Metal instances.  
  - Fix LSO pseudo-header checksum calculation error over IPv6.  
  - Fix potential memory resource leak upon initialization failure.  
  - Disable TCP/UDP checksum offload for IPv4 fragments.  
  - Fix for VLAN configuration. VLAN was incorrectly disabled when only VLAN priority should have been disabled.  
  - Enable correct parsing of custom driver messages by the event viewer.  
  - Fix failure to initialize driver due to invalid timestamp handling.  
  - Fix race condition between data processing and ENA device disabling. | July 1, 2019
1.5.0 | 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 4, 2018
1.2.3 | Includes reliability fixes and unifies support for Windows Server 2008 R2 through Windows Server 2016. | February 13, 2018

967
Enhanced networking: Intel 82599 VF

<table>
<thead>
<tr>
<th>Driver version</th>
<th>Details</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.9</td>
<td>Includes some reliability fixes. Applies only to Windows Server 2008 R2. Not recommended for other versions of Windows Server.</td>
<td>December 2016</td>
</tr>
</tbody>
</table>

Subscribe 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

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, enter 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

2. In the navigation pane, choose Subscriptions.
3. Select the check box for the subscription and then choose Actions, Delete subscriptions. When prompted for confirmation, choose Delete.

Enable 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.
• Requirements (p. 969)
• Test whether enhanced networking is enabled (p. 969)
• Enable enhanced networking on Windows (p. 970)

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 and later 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.
• Use AWS CloudShell from the AWS Management Console, or 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 Access Amazon EC2 (p. 3) or the AWS CloudShell User Guide. 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.

Test 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 CloudShell)

```bash
aws ec2 describe-instance-attribute --instance-id instance_id --attribute sriovNetSupport
```

• Get-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

```powershell
Get-EC2InstanceAttribute -InstanceId instance_id -Attribute sriovNetSupport
```

If the attribute isn't set, SriovNetSupport is empty. If the attribute is set, the value is simple, as shown in the following example output.

```
"SriovNetSupport": {
```
Image attribute (sriovNetSupport)

To check whether an AMI already has the enhanced networking sriovNetSupport attribute set, use one of the following commands:

- **describe-images** (AWS CLI/AWS CloudShell)
  
  ```bash
  aws ec2 describe-images --image-id ami_id --query "Images[].SriovNetSupport"
  ```

- **Get-EC2Image** (AWS Tools for Windows PowerShell)
  
  ```powershell
  (Get-EC2Image -ImageId ami-id).SriovNetSupport
  ```

If the attribute isn’t set, SriovNetSupport is empty. If the attribute is set, the value is simple.

Enable 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 sriovNetSupport instance attribute to activate enhanced networking. You can only enable this attribute on supported instance types. For more information, see Enhanced networking support (p. 961).

**Important**

To view the latest version of the Intel driver in the Windows AMIs, see Details about AWS Windows AMI versions (p. 30).

**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 and later] Run the following EC2Launch PowerShell script to configure the instance after the driver is installed.

   ```powershell
   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. 488).

3. From the instance, download the Intel network adapter driver for your operating system:

   - **Windows Server 2019** including for Server version 1809 and later*
     
     Visit the [download page](#) and download Wired_driver_version_x64.zip.

   - **Windows Server 2016** including for Server version 1803 and earlier*
     
     Visit the [download page](#) and download Wired_driver_version_x64.zip.
Enhanced networking: Intel 82599 VF

- **Windows Server 2012 R2**
  
  Visit the [download page](#) and download `Wired_driver_version_x64.zip`.

- **Windows Server 2012**
  
  Visit the [download page](#) and download `Wired_driver_version_x64.zip`.

- **Windows Server 2008 R2**
  
  Visit the [download page](#) and download `PROWinx64Legacy.exe`.

*Server versions 1803 and earlier as well as 1809 and later are not specifically addressed on the Intel Drivers and Software pages.

4. Install the Intel network adapter driver for your operating system.

   - **Windows Server 2008 R2**
     1. In the **Downloads** folder, locate the `PROWinx64Legacy.exe` file and rename it to `PROWinx64Legacy.zip`.
     2. Extract the contents of the `PROWinx64Legacy.zip` file.
     3. Open the command line, navigate to the extracted folder, and run the following command to use the `pnputil` utility to add and install the INF file in the driver store.

     ```cmd
     C:\> pnputil -a PROXGB\Winx64\NDIS62\vxn62x64.inf
     ```

     1. In the **Downloads** folder, extract the contents of the `Wired_driver_version_x64.zip` file.
     2. In the extracted folder, locate the `Wired_driver_version_x64.exe` file and rename it to `Wired_driver_version_x64.zip`.
     3. Extract the contents of the `Wired_driver_version_x64.zip` file.
     4. Open the command line, navigate to the extracted folder, and run one of the following commands to use the `pnputil` utility to add and install the INF file in the driver store.

     - **Windows Server 2019**
       ```cmd
       C:\> pnputil -i -a PROXGB\Winx64\NDIS68\vxn68x64.inf
       ```

     - **Windows Server 2016**
       ```cmd
       C:\> pnputil -i -a PROXGB\Winx64\NDIS65\vxn65x64.inf
       ```

     - **Windows Server 2012 R2**
       ```cmd
       C:\> pnputil -i -a PROXGB\Winx64\NDIS64\vxn64x64.inf
       ```

     - **Windows Server 2012**
       ```cmd
       C:\> pnputil -i -a PROXGB\Winx64\NDIS63\vxn63x64.inf
       ```

   5. 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.

   6. From your local computer, enable the enhanced networking attribute using one of the following commands:
Operating system optimizations

To achieve the maximum network performance on instances with enhanced networking, you might 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. 987).

AWS regularly measures average round trip latencies between instances launched in a cluster placement group of 50us and tail latencies of 200us at the 99.9 percentile. If your applications require consistently low latencies, we recommend using the latest version of the ENA drivers on fixed performance instances built on the Nitro System.

Configure RSS CPU affinity

Receive side scaling (RSS) is used to distribute network traffic CPU load across multiple processors. By default, the official Amazon Windows AMIs are configured with RSS enabled. ENA ENIs provide up to eight RSS queues. By defining CPU affinity for RSS queues, as well as for other system processes, it is possible to spread the CPU load out over multi-core systems, enabling more network traffic to be processed. On instance types with more than 16 vCPUs, we recommend you use the Set-NetAdapterRSS PowerShell cmdlt (available from Windows Server 2012 and later), which manually excludes the boot processor (logical processor 0 and 1 when hyper-threading is enabled) from the RSS configuration for all ENIs, in order to prevent contention with various system components.

Windows is hyper-thread aware and will ensure the RSS queues of a single NIC are always placed on different physical cores. Therefore, unless hyper-threading is disabled, in order to completely prevent contention with other NICs, spread the RSS configuration of each NIC between a range of 16 logical processors. The Set-NetAdapterRSS cmdlt allows you to define the per-NIC range of valid logical processors by defining the values of BaseProcessorGroup, BaseProcessorNumber, MaxProcessingGroup, MaxProcessorNumber, and NumaNode (optional). If there are not enough physical cores to completely eliminate inter-NIC contention, minimize the overlapping ranges or reduce the number of logical processors in the ENI ranges depending on the expected workload of the ENI (in other words, a low
volume admin network ENI may not need as many RSS queues assigned). Also, as noted above, various components must run on CPU 0, and therefore we recommend excluding it from all RSS configurations when sufficient vCPUs are available.

For example, when there are three ENIs on a 72 vCPU instance with 2 NUMA nodes with hyper-threading enabled, the following commands spread the network load between the two CPUs without overlap and prevent the use of core 0 completely.

```plaintext
Set-NetAdapterRss -Name NIC1 -BaseProcessorGroup 0 -BaseProcessorNumber 2 -MaxProcessorNumber 16
Set-NetAdapterRss -Name NIC2 -BaseProcessorGroup 1 -BaseProcessorNumber 0 -MaxProcessorNumber 14
Set-NetAdapterRss -Name NIC3 -BaseProcessorGroup 1 -BaseProcessorNumber 16 -MaxProcessorNumber 30
```

Note that these settings are persistent for each network adapter. If an instance is resized to one with a different number of vCPUs, you should reevaluate the RSS configuration for each enabled ENI. The complete Microsoft documentation for the `Set-NetAdapterRss` cmdlet can be found here: https://docs.microsoft.com/en-us/powershell/module/netadapter/set-netadapterrss.

Special note for SQL workloads: We also recommend that you review your IO thread affinity settings along with your ENI RSS configuration to minimize IO and network contention for the same CPUs. See affinity mask Server Configuration Option.

## Monitor network performance for your EC2 instance

The Elastic Network Adapter (ENA) driver publishes network performance metrics from the instances where they are enabled. You can use these metrics to troubleshoot instance performance issues, choose the right instance size for a workload, plan scaling activities proactively, and benchmark applications to determine whether they maximize the performance available on an instance.

Amazon EC2 defines network maximums at the instance level to ensure a high-quality networking experience, including consistent network performance across instance sizes. AWS provides maximums for the following for each instance:

- **Bandwidth capability** – Each EC2 instance has a maximum bandwidth for aggregate inbound and outbound traffic, based on instance type and size. Some instances use a network I/O credit mechanism to allocate network bandwidth based on average bandwidth utilization. Amazon EC2 also has maximum bandwidth for traffic to AWS Direct Connect and the internet.
- **Packet-per-second (PPS) performance** – Each EC2 instance has a maximum PPS performance, based on instance type and size.
- **Connections tracked** – The security group tracks each connection established to ensure that return packets are delivered as expected. There is a maximum number of connections that can be tracked per instance.
- **Link-local service access** – Amazon EC2 provides a maximum PPS per network interface for traffic to services such as the DNS service, the Instance Metadata Service, and the Amazon Time Sync Service.

When the network traffic for an instance exceeds a maximum, AWS shapes the traffic that exceeds the maximum by queueing and then dropping network packets. You can monitor when traffic exceeds a maximum using the network performance metrics. These metrics inform you, in real time, of impact to network traffic and possible network performance issues.

## Contents

- Requirements (p. 974)
- Metrics for the ENA driver (p. 974)
- View the network performance metrics for your Windows instance (p. 974)
Requirements

- Install ENA driver version 2.2.2 or later. To verify the installed version, use Device Manager as follows.
  1. Open Device Manager by running `devmgmt.msc`.
  2. Expand Network Adapters.

To upgrade your ENA driver, see Enhanced networking (p. 961).

- To import these metrics to Amazon CloudWatch, install the CloudWatch agent. For more information, see Collect advanced network metrics in the Amazon CloudWatch User Guide.

Metrics for the ENA driver

The ENA driver delivers the following metrics to the instance in real time. They provide the cumulative number of packets queued or dropped on each network interface since the last driver reset.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bw_in_allowance_exceeded</code></td>
<td>The number of packets queued or dropped because the inbound aggregate bandwidth exceeded the maximum for the instance.</td>
</tr>
<tr>
<td><code>bw_out_allowance_exceeded</code></td>
<td>The number of packets queued or dropped because the outbound aggregate bandwidth exceeded the maximum for the instance.</td>
</tr>
<tr>
<td><code>conntrack_allowance_exceeded</code></td>
<td>The number of packets dropped because connection tracking exceeded the maximum for the instance and new connections could not be established. This can result in packet loss for traffic to or from the instance.</td>
</tr>
<tr>
<td><code>linklocal_allowance_exceeded</code></td>
<td>The number of packets dropped because the PPS of the traffic to local proxy services exceeded the maximum for the network interface. This impacts traffic to the DNS service, the Instance Metadata Service, and the Amazon Time Sync Service.</td>
</tr>
<tr>
<td><code>pps_allowance_exceeded</code></td>
<td>The number of packets queued or dropped because the bidirectional PPS exceeded the maximum for the instance.</td>
</tr>
</tbody>
</table>

View the network performance metrics for your Windows instance

You can view the metrics using any consumer of Windows performance counters. The data can be parsed according to the EnaPerfCounters manifest. This is an XML file that defines the performance counter provider and its countersets.

Manifest installation

If you launched the instance using an AMI that contains ENA driver 2.2.2 or later, or used the install script in the driver package for ENA driver 2.2.2, the manifest is already installed. To install the manifest manually, use the following steps:
Placement groups

When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures. You can use placement groups to influence the placement of a group of interdependent instances to meet the needs of your workload. Depending on the type of workload, you can create a placement group using one of the following placement strategies:

- **Cluster** – packs instances close together inside an Availability Zone. This strategy enables workloads to achieve the low-latency network performance necessary for tightly-coupled node-to-node communication that is typical of HPC applications.
- **Partition** – spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions. This strategy is typically used by large distributed and replicated workloads, such as Hadoop, Cassandra, and Kafka.
- **Spread** – strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.

There is no charge for creating a placement group.

Contents

- Cluster placement groups (p. 976)
- Partition placement groups (p. 976)
- Spread placement groups (p. 977)
- Placement group rules and limitations (p. 978)
- Create a placement group (p. 979)
- Tag a placement group (p. 980)
- Launch instances in a placement group (p. 982)
- Describe instances in a placement group (p. 983)
- Change the placement group for an instance (p. 985)
Cluster placement groups

A cluster placement group is a logical grouping of instances within a single Availability Zone. A cluster placement group can span peered VPCs in the same Region. Instances in the same cluster placement group enjoy a higher per-flow throughput limit for TCP/IP traffic and are placed in the same high-bisection bandwidth segment of the network.

The following image shows instances that are placed into a cluster placement group.

Cluster placement groups are recommended for applications that benefit from low network latency, high network throughput, or both. They are also recommended when the majority of the network traffic is between the instances in the group. To provide the lowest latency and the highest packet-per-second network performance for your placement group, choose an instance type that supports enhanced networking. For more information, see Enhanced Networking (p. 960).

We recommend that you launch your instances in the following way:

- Use a single launch request to launch the number of instances that you need in the placement group.
- Use the same instance type for all instances in the placement group.

If you try to add more instances to the placement group later, or if you try to launch more than one instance type in the placement group, you increase your chances of getting an insufficient capacity error.

If you stop an instance in a placement group and then start it again, it still runs in the placement group. However, the start fails if there isn't enough capacity for the instance.

If you receive a capacity error when launching an instance in a placement group that already has running instances, stop and start all of the instances in the placement group, and try the launch again. Starting the instances may migrate them to hardware that has capacity for all of the requested instances.

Partition placement groups

Partition placement groups help reduce the likelihood of correlated hardware failures for your application. When using partition placement groups, Amazon EC2 divides each group into logical segments called partitions. Amazon EC2 ensures that each partition within a placement group has
its own set of racks. Each rack has its own network and power source. No two partitions within a placement group share the same racks, allowing you to isolate the impact of hardware failure within your application.

The following image is a simple visual representation of a partition placement group in a single Availability Zone. It shows instances that are placed into a partition placement group with three partitions—**Partition 1**, **Partition 2**, and **Partition 3**. Each partition comprises multiple instances. The instances in a partition do not share racks with the instances in the other partitions, allowing you to contain the impact of a single hardware failure to only the associated partition.

Partition placement groups can be used to deploy large distributed and replicated workloads, such as HDFS, HBase, and Cassandra, across distinct racks. When you launch instances into a partition placement group, Amazon EC2 tries to distribute the instances evenly across the number of partitions that you specify. You can also launch instances into a specific partition to have more control over where the instances are placed.

A partition placement group can have partitions in multiple Availability Zones in the same Region. A partition placement group can have a maximum of seven partitions per Availability Zone. The number of instances that can be launched into a partition placement group is limited only by the limits of your account.

In addition, partition placement groups offer visibility into the partitions—you can see which instances are in which partitions. You can share this information with topology-aware applications, such as HDFS, HBase, and Cassandra. These applications use this information to make intelligent data replication decisions for increasing data availability and durability.

If you start or launch an instance in a partition placement group and there is insufficient unique hardware to fulfill the request, the request fails. Amazon EC2 makes more distinct hardware available over time, so you can try your request again later.

### Spread placement groups

A spread placement group is a group of instances that are each placed on distinct racks, with each rack having its own network and power source.

The following image shows seven instances in a single Availability Zone that are placed into a spread placement group. The seven instances are placed on seven different racks.

Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other. Launching instances in a spread placement
group reduces the risk of simultaneous failures that might occur when instances share the same racks. Spread placement groups provide access to distinct racks, and are therefore suitable for mixing instance types or launching instances over time.

A spread placement group can span multiple Availability Zones in the same Region. You can have a maximum of seven running instances per Availability Zone per group.

If you start or launch an instance in a spread placement group and there is insufficient unique hardware to fulfill the request, the request fails. Amazon EC2 makes more distinct hardware available over time, so you can try your request again later.

## Placement group rules and limitations

### General rules and limitations

Before you use placement groups, be aware of the following rules:

- The name that you specify for a placement group must be unique within your AWS account for the Region.
- You can't merge placement groups.
- An instance can be launched in one placement group at a time; it cannot span multiple placement groups.
- On-Demand Capacity Reservation (p. 367) and zonal Reserved Instances (p. 247) provide a capacity reservation for EC2 instances in a specific Availability Zone. The capacity reservation can be used by instances in a placement group. However, it is not possible to explicitly reserve capacity for a placement group.
- You cannot launch Dedicated Hosts in placement groups.

### Cluster placement group rules and limitations

The following rules apply to cluster placement groups:

- The following instance types are supported:
  - Current generation (p. 142) instances, except for burstable performance (p. 160) instances (for example, T2).
  - The following previous generation (p. 145) instances: A1, C3, cc2.8xlarge, cr1.8xlarge, G2, hs1.8xlarge, I2, and R3.
- A cluster placement group can't span multiple Availability Zones.
- The maximum network throughput speed of traffic between two instances in a cluster placement group is limited by the slower of the two instances. For applications with high-throughput requirements, choose an instance type with network connectivity that meets your requirements.
- For instances that are enabled for enhanced networking, the following rules apply:
  - Instances within a cluster placement group can use up to 10 Gbps for single-flow traffic. Instances that are not within a cluster placement group can use up to 5 Gbps for single-flow traffic.
  - Traffic to and from Amazon S3 buckets within the same Region over the public IP address space or through a VPC endpoint can use all available instance aggregate bandwidth.
- You can launch multiple instance types into a cluster placement group. However, this reduces the likelihood that the required capacity will be available for your launch to succeed. We recommend using the same instance type for all instances in a cluster placement group.
- Network traffic to the internet and over an AWS Direct Connect connection to on-premises resources is limited to 5 Gbps.
Partition placement group rules and limitations

The following rules apply to partition placement groups:

- A partition placement group supports a maximum of seven partitions per Availability Zone. The number of instances that you can launch in a partition placement group is limited only by your account limits.
- When instances are launched into a partition placement group, Amazon EC2 tries to evenly distribute the instances across all partitions. Amazon EC2 doesn’t guarantee an even distribution of instances across all partitions.
- A partition placement group with Dedicated Instances can have a maximum of two partitions.

Spread placement group rules and limitations

The following rules apply to spread placement groups:

- A spread placement group supports a maximum of seven running instances per Availability Zone. For example, in a Region with three Availability Zones, you can run a total of 21 instances in the group (seven per zone). If you try to start an eighth instance in the same Availability Zone and in the same spread placement group, the instance will not launch. If you need to have more than seven instances in an Availability Zone, then the recommendation is to use multiple spread placement groups. Using multiple spread placement groups does not provide guarantees about the spread of instances between groups, but it does ensure the spread for each group, thus limiting impact from certain classes of failures.
- Spread placement groups are not supported for Dedicated Instances.

Create a placement group

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

Note
You can tag a placement group on creation using the command line tools only.

New console

To create a placement group using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Placement Groups, Create placement group.
3. Specify a name for the group.
4. Choose the placement strategy for the group. If you choose Partition, choose the number of partitions within the group.
5. Choose Create group.

Old console

To create a placement group using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Placement Groups, Create Placement Group.
3. Specify a name for the group.
4. Choose the placement strategy for the group. If you choose Partition, specify the number of partitions within the group.
Tag a placement group

5. Choose Create.

AWS CLI

To create a placement group using the AWS CLI

Use the create-placement-group command. The following example creates a placement group named my-cluster that uses the cluster placement strategy, and it applies a tag with a key of purpose and a value of production.

```
aws ec2 create-placement-group --group-name my-cluster --strategy cluster --tag-specifications 'ResourceType=placement-group,Tags={Key=purpose,Value=production}'
```

To create a partition placement group using the AWS CLI

Use the create-placement-group command. Specify the --strategy parameter with the value partition, and specify the --partition-count parameter with the desired number of partitions. In this example, the partition placement group is named HDFS-Group-A and is created with five partitions.

```
aws ec2 create-placement-group --group-name HDFS-Group-A --strategy partition --partition-count 5
```

PowerShell

To create a placement group using the AWS Tools for Windows PowerShell

Use the New-EC2PlacementGroup command.

Tag a placement group

To help categorize and manage your existing placement groups, you can tag them with custom metadata. For more information about how tags work, see Tag your Amazon EC2 resources (p. 1450).

When you tag a placement group, the instances that are launched into the placement group are not automatically tagged. You need to explicitly tag the instances that are launched into the placement group. For more information, see Add a tag when you launch an instance (p. 1458).

You can view, add, and delete tags using the new console and the command line tools.

New console

To view, add, or delete a tag for an existing placement group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Placement Groups.
3. Select a placement group, and then choose Actions, Manage tags.
4. The Manage tags section displays any tags that are assigned to the placement group. Do the following to add or remove tags:
   a. To add a tag, choose Add tag, and then enter the tag key and value. You can add up to 50 tags per placement group. For more information, see Tag restrictions (p. 1454).
   b. To delete a tag, choose Remove next to the tag that you want to delete.
5. Choose Save changes.
AWS CLI

To view placement group tags

Use the `describe-tags` command to view the tags for the specified resource. In the following example, you describe the tags for all of your placement groups.

```
aws ec2 describe-tags \
  --filters Name=resource-type,Values=placement-group
```

```
{
 "Tags": [
   
   
   
   "Key": "Environment",
   "ResourceId": "pg-0123456789EXAMPLE",
   "ResourceType": "placement-group",
   "Value": "Production"
   
   
   "Key": "Environment",
   "ResourceId": "pg-9876543210EXAMPLE",
   "ResourceType": "placement-group",
   "Value": "Production"
  
  
  ]
}
```

You can also use the `describe-tags` command to view the tags for a placement group by specifying its ID. In the following example, you describe the tags for `pg-0123456789EXAMPLE`.

```
aws ec2 describe-tags \
  --filters Name=resource-id,Values=pg-0123456789EXAMPLE
```

```
{
 "Tags": [
   
   "Key": "Environment",
   "ResourceId": "pg-0123456789EXAMPLE",
   "ResourceType": "placement-group",
   "Value": "Production"
   
   ]
}
```

You can also view the tags of a placement group by describing the placement group.

Use the `describe-placement-groups` command to view the configuration of the specified placement group, which includes any tags that were specified for the placement group.

```
aws ec2 describe-placement-groups \
  --group-name my-cluster
```

```
{
 "PlacementGroups": [
   
   "GroupName": "my-cluster",
   "State": "available",
   "Strategy": "cluster",
   "GroupId": "pg-0123456789EXAMPLE",
   
   ]
}
```
To tag an existing placement group using the AWS CLI

You can use the create-tags command to tag existing resources. In the following example, the existing placement group is tagged with Key=Cost-Center and Value=CC-123.

```bash
aws ec2 create-tags
   --resources pg-0123456789EXAMPLE
   --tags Key=Cost-Center,Value=CC-123
```

To delete a tag from a placement group using the AWS CLI

You can use the delete-tags command to delete tags from existing resources. For examples, see Examples in the AWS CLI Command Reference.

PowerShell

To view placement group tags

Use the Get-EC2Tag command.

To describe the tags for a specific placement group

Use the Get-EC2PlacementGroup command.

To tag an existing placement group

Use the New-EC2Tag command.

To delete a tag from a placement group

Use the Remove-EC2Tag command.

Launch instances in a placement group

You can launch an instance into a placement group if the placement group rules and limitations are met (p. 978) using one of the following methods.

Console

To launch instances into a placement group using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Choose Launch Instance. Complete the wizard as directed, taking care to do the following:
   - On the Choose an Instance Type page, select an instance type that can be launched into a placement group.
   - On the Configure Instance Details page, the following fields are applicable to placement groups:
For **Number of instances**, enter the total number of instances that you need in this placement group, because you might not be able to add instances to the placement group later.

For **Placement group**, select the **Add instance to placement group** check box. If you do not see **Placement group** on this page, verify that you have selected an instance type that can be launched into a placement group. Otherwise, this option is not available.

For **Placement group name**, you can choose to add the instances to an existing placement group or to a new placement group that you create.

For **Placement group strategy**, choose the appropriate strategy. If you choose **partition**, for **Target partition**, choose **Auto distribution** to have Amazon EC2 do a best effort to distribute the instances evenly across all the partitions in the group. Alternatively, specify the partition in which to launch the instances.

### AWS CLI

**To launch instances into a placement group using the AWS CLI**

Use the `run-instances` command and specify the placement group name using the `--placement GroupName = my-cluster` parameter. In this example, the placement group is named `my-cluster`.

```bash
aws ec2 run-instances --placement "GroupName = my-cluster"
```

**To launch instances into a specific partition of a partition placement group using the AWS CLI**

Use the `run-instances` command and specify the placement group name and partition using the `--placement GroupName = HDFS-Group-A, PartitionNumber = 3` parameter. In this example, the placement group is named `HDFS-Group-A` and the partition number is 3.

```bash
aws ec2 run-instances --placement "GroupName = HDFS-Group-A, PartitionNumber = 3"
```

### PowerShell

**To launch instances into a placement group using AWS Tools for Windows PowerShell**

Use the `New-EC2Instance` command and specify the placement group name using the `-Placement_GroupName` parameter.

### Describe instances in a placement group

You can view the placement information of your instances using one of the following methods. You can also filter partition placement groups by the partition number using the AWS CLI.

### New console

**To view the placement group and partition number of an instance using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. On the **Details** tab, under **Host and placement group**, find **Placement group**. If the instance is not in a placement group, the field is empty. Otherwise, it contains the name of the placement group name. If the placement group is a partition placement group, **Partition number** contains the partition number for the instance.
Old console

To view the placement group and partition number of an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance.
4. In the Description tab, find Placement group. If the instance is not in a placement group, the field is empty. Otherwise, it contains the name of the placement group name. If the placement group is a partition placement group, Partition number contains the partition number for the instance.

AWS CLI

To view the partition number for an instance in a partition placement group using the AWS CLI

Use the describe-instances command and specify the --instance-id parameter.

```bash
aws ec2 describe-instances --instance-id i-0123a456700123456
```

The response contains the placement information, which includes the placement group name and the partition number for the instance.

```
"Placement": {
  "AvailabilityZone": "us-east-1c",
  "GroupName": "HDFS-Group-A",
  "PartitionNumber": 3,
  "Tenancy": "default"
}
```

To filter instances for a specific partition placement group and partition number using the AWS CLI

Use the describe-instances command and specify the --filters parameter with the placement-group-name and placement-partition-number filters. In this example, the placement group is named HDFS-Group-A and the partition number is 7.

```bash
aws ec2 describe-instances --filters "Name = placement-group-name, Values = HDFS-Group-A" "Name = placement-partition-number, Values = 7"
```

The response lists all the instances that are in the specified partition within the specified placement group. The following is example output showing only the instance ID, instance type, and placement information for the returned instances.

```
"Instances": [
  {
    "InstanceId": "i-0a1bc23d4567e8f90",
    "InstanceType": "r4.large",
  },
  "Placement": {
    "AvailabilityZone": "us-east-1c",
    "GroupName": "HDFS-Group-A",
    "PartitionNumber": 7,
    "Tenancy": "default"
  }
]
```
Change the placement group for an instance

You can change the placement group for an instance in any of the following ways:

- Move an existing instance to a placement group
- Move an instance from one placement group to another
- Remove an instance from a placement group

Before you move or remove the instance, the instance must be in the stopped state. You can move or remove an instance using the AWS CLI or an AWS SDK.

**AWS CLI**

**To move an instance to a placement group using the AWS CLI**

1. Stop the instance using the `stop-instances` command.
2. Use the `modify-instance-placement` command and specify the name of the placement group to which to move the instance.

   ```bash
   aws ec2 modify-instance-placement --instance-id i-0a9b876cd5d4ef321 --group-name MySpreadGroup
   ```
3. Start the instance using the `start-instances` command.

**PowerShell**

**To move an instance to a placement group using the AWS Tools for Windows PowerShell**

1. Stop the instance using the `Stop-EC2Instance` command.
2. Use the `Edit-EC2InstancePlacement` command and specify the name of the placement group to which to move the instance.
3. Start the instance using the `Start-EC2Instance` command.

**AWS CLI**

**To remove an instance from a placement group using the AWS CLI**

1. Stop the instance using the `stop-instances` command.
2. Use the `modify-instance-placement` command and specify an empty string for the placement group name.
aws ec2 modify-instance-placement --instance-id i-0123a456700123456 --group-name ""

3. Start the instance using the `start-instances` command.

PowerShell

**To remove an instance from a placement group using the AWS Tools for Windows PowerShell**

1. Stop the instance using the `Stop-EC2Instance` command.
2. Use the `Edit-EC2InstancePlacement` command and specify an empty string for the placement group name.
3. Start the instance using the `Start-EC2Instance` command.

Delete a placement group

If you need to replace a placement group or no longer need one, you can delete it. You can delete a placement group using one of the following methods.

**Requirement**

Before you can delete a placement group, it must contain no instances. You can terminate (p. 443) all instances that you launched into the placement group, move (p. 985) them to another placement group, or remove (p. 985) them from the placement group.

New console

**To delete a placement group using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Placement Groups**.
3. Select the placement group and choose **Actions, Delete**.
4. When prompted for confirmation, enter **Delete** and then choose **Delete**.

Old console

**To delete a placement group using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Placement Groups**.
3. Select the placement group and choose **Actions, Delete Placement Group**.
4. When prompted for confirmation, choose **Delete**.

AWS CLI

**To delete a placement group using the AWS CLI**

Use the `delete-placement-group` command and specify the placement group name to delete the placement group. In this example, the placement group name is `my-cluster`.

```
aws ec2 delete-placement-group --group-name my-cluster
```
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.

The following rules apply to instances that are in Wavelength Zones:

- Traffic that goes from one instance to another within a VPC in the same Wavelength Zone has an MTU of 1300.
- Traffic that goes from one instance to another that uses the carrier IP within a Wavelength Zone has an MTU of 1500.
- Traffic that goes from one instance to another between a Wavelength Zone and the Region that uses a public IP address has an MTU of 1500.
- Traffic that goes from one instance to another between a Wavelength Zone and the Region that uses a private IP address has an MTU of 1300.

To see Network MTU information for Linux instances, switch to this page in the Amazon EC2 User Guide for Linux Instances guide: Network maximum transmission unit (MTU) for your EC2 instance.

Contents

- Jumbo frames (9001 MTU) (p. 987)
- Path MTU Discovery (p. 988)
- Check the path MTU between two hosts (p. 988)
- Check and set the MTU on your Windows instance (p. 989)
- Troubleshoot (p. 991)

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, traffic is limited to a maximum MTU of 1500 in the following cases:

- Traffic outside of a given AWS Region for EC2-Classic
- Traffic outside of a single VPC
- Traffic over an inter-region VPC peering connection
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.

For IPv4, when 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 drops the packet, and then returns the following ICMP message: Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (Type 3, Code 4). This instructs the transmitting host to split the payload into multiple smaller packets, and then retransmit them.

The IPv6 protocol does not support fragmentation in the network. When 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 drops the packet, and then returns the following ICMP message: ICMPv6 Packet Too Big (PTB) (Type 2). This instructs the transmitting host to split the payload into multiple smaller packets, and then retransmit them.

By default, security groups do not allow any inbound ICMP traffic. However, security groups are stateful, therefore ICMP responses to outbound requests are allowed to flow in, regardless of security group rules. Therefore, you do not need to explicitly add an inbound ICMP rule to ensure that your instance can receive the ICMP message response. For more information about configuring ICMP rules in a network ACL, see Path MTU Discovery in the Amazon VPC User Guide.

Important

Path MTU Discovery does not guarantee that jumbo frames will not be dropped by some routers. An internet gateway in your VPC will forward packets up to 1500 bytes only. 1500 MTU packets are recommended for internet traffic.

Check the path MTU between two hosts

You can check the path MTU between two hosts using the mturute.exe command, which you can download and install from http://www.elifulkerson.com/projects/mturoute.php.
To check path MTU using mturoute.exe

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.

```bash
\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.

If your instance runs in a Wavelength Zone, the maximum MTU value is 1300.

ENA Driver

For Driver Versions 1.5 and Earlier

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
```

For Driver Versions 2.1.0 and Later

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 9015 indicates that Jumbo frames are enabled. Jumbo frames are disabled by default.

Run `Get-NetAdapterAdvancedProperty` or use wildcard (asterisk) to detect all corresponding Ethernet names.

```
Get-NetAdapterAdvancedProperty -Name "Ethernet*"
```

Run the following commands and include the Ethernet name you want to query.

```
Get-NetAdapterAdvancedProperty -Name "Ethernet"
```

Enable jumbo frames as follows.

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -RegistryValue 9015
```

Disable jumbo frames as follows:

```
Set-NetAdapterAdvancedProperty -Name "Ethernet" -RegistryKeyword "*JumboPacket" -RegistryValue 1514
```

**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
```

**Troubleshoot**

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. 1022).

**Amazon VPC documentation**

For more information about Amazon VPC, see the following documentation.

<table>
<thead>
<tr>
<th>Guide</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon VPC User Guide</td>
<td>Describes key concepts and provides instructions for using the features of Amazon VPC.</td>
</tr>
<tr>
<td>Amazon VPC Peering Guide</td>
<td>Describes VPC peering connections and provides instructions for using them.</td>
</tr>
<tr>
<td>Amazon VPC Transit Gateways</td>
<td>Describes transit gateways and provides instructions for configuring and using them.</td>
</tr>
</tbody>
</table>
Ports and Protocols for Windows Amazon Machine Images (AMIs)

The following tables list the ports, protocols, and directions by workload for Windows Amazon Machine Images.

Contents

- AllJoyn Router (p. 992)
- Cast to Device (p. 993)
- Core Networking (p. 995)
- Delivery Optimization (p. 1013)
- Diag Track (p. 1014)
- DIAL Protocol Server (p. 1014)
- Distributed File System (DFS) Management (p. 1014)
- File and Printer Sharing (p. 1015)
- File Server Remote Management (p. 1017)
- ICMP v4 All (p. 1018)
- Multicast (p. 1018)
- Remote Desktop (p. 1019)
- Windows Device Management (p. 1021)
- Windows Firewall Remote Management (p. 1021)
- Windows Remote Management (p. 1022)

AllJoyn Router

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Description</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2016</td>
<td>AllJoyn Router (TCP-In)</td>
<td>Inbound rule for AllJoyn Router traffic [TCP]</td>
<td>Local: 9955</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td>Windows Server 2019</td>
<td>AllJoyn Router (TCP-Out)</td>
<td>Outbound rule for AllJoyn Router traffic [TCP]</td>
<td>Local: Any</td>
<td>TCP</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td>AllJoyn Router (UDP-In)</td>
<td>Inbound rule for AllJoyn Router traffic [UDP]</td>
<td>Local: Any</td>
<td>UDP</td>
<td>In</td>
</tr>
</tbody>
</table>
## Cast to Device

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Description</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllJoyn Router</td>
<td>Outbound rule for AllJoyn</td>
<td>Outbound rule for AllJoyn Router traffic [UDP]</td>
<td>Local: Any</td>
<td>UDP</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td>(UDP-Out)</td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2016</td>
<td>Cast to Device functionality</td>
<td>Inbound rule for the Cast to Device functionality to allow use of the Quality Windows Audio Video Experience Service. [TCP 2177]</td>
<td>Local: 2177</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td>Windows Server 2019</td>
<td>Cast to Device functionality</td>
<td>Outbound rule for the Cast to Device functionality to allow use of the Quality Windows Audio Video Experience Service. [TCP 2177]</td>
<td>Local: Any</td>
<td>TCP</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td>(qWave-TCP-Out)</td>
<td></td>
<td>Remote: 2177</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cast to Device functionality</td>
<td>Inbound rule for the Cast to Device functionality to allow use of the Quality Windows Audio Video Experience Service. [UDP 2177]</td>
<td>Local: 2177</td>
<td>UDP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>(qWave-UDP-In)</td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cast to Device functionality</td>
<td>Outbound rule for the Cast to Device functionality to allow use of the Quality Windows Audio Video Experience Service. [UDP 2177]</td>
<td>Local: Any</td>
<td>UDP</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td>(qWave-UDP-Out)</td>
<td></td>
<td>Remote: 2177</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>Rule</td>
<td>Description</td>
<td>Port</td>
<td>Protocol</td>
<td>Direction</td>
</tr>
<tr>
<td>----</td>
<td>------</td>
<td>-------------</td>
<td>------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>Experience Service. [UDP 2177]</td>
<td>Inbound rule to allow discovery of Cast to Device targets using SSDP</td>
<td>Local: Ply2Disc Remote: Any</td>
<td>UDP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>Cast to Device SSDP Discovery (UDP-In)</td>
<td>Inbound rule for the Cast to Device server to allow streaming using HTTP. [TCP 10246]</td>
<td>Local: 10246 Remote: Any</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>Cast to Device Streaming Server (HTTP-Streaming-In)</td>
<td>Inbound rule for the Cast to Device server to allow streaming using RTSP and RTP. [UDP]</td>
<td>Local: Any Remote: Any</td>
<td>UDP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>Cast to Device Streaming Server (RTCP-Streaming-In)</td>
<td>Outbound rule for the Cast to Device server to allow streaming using RTSP and RTP. [UDP]</td>
<td>Local: Any Remote: Any</td>
<td>UDP</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td>Cast to Device Streaming Server (RTP-Streaming-Out)</td>
<td>Inbound rule for the Cast to Device server to allow streaming using RTSP and RTP. [TCP 23554, 23555, 23556]</td>
<td>Local: 235, 542, 355, 523, 556 Remote: Any</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>Cast to Device Streaming Server (RTSP-Streaming-In)</td>
<td>Inbound rule to allow receiving UPnP Events from Cast to Device targets</td>
<td>Local: 2869 Remote: Any</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>Cast to Device UPnP Events (TCP-In)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Core Networking


<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2012</td>
<td>Destination Unreachable (ICMPv6-In)</td>
<td>Destination Unreachable error messages are sent from any node that a packet traverses which is unable to forward the packet for any reason except congestion.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2016</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2019</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2012</td>
<td>Destination Unreachable Fragmentation Needed (ICMPv4-In)</td>
<td>Destination Unreachable Fragmentation Needed error messages are sent from any node that a packet traverses which is unable to forward the packet because fragmentation was needed and the don't fragment bit was set.</td>
<td>Local: 68</td>
<td>ICMPv4</td>
<td>In</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2016</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2019</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core Networking - DNS (UDP-Out)</td>
<td>Outbound rule to allow DNS requests. DNS responses based on requests that match this rule are permitted regardless of source address. This behavior is classified as</td>
<td>Local: Any</td>
<td>UDP</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>Rule</td>
<td>Definition</td>
<td>Port</td>
<td>Protocol</td>
<td>Direction</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>------------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>Dynamic Host Configuration Protocol (DHCP-In)</td>
<td>Allows DHCP (Dynamic Host Configuration Protocol) messages for stateful auto-configuration.</td>
<td>Local: 68</td>
<td>UDP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dynamic Host Configuration Protocol (DHCP-Out)</td>
<td>Allows DHCP (Dynamic Host Configuration Protocol) messages for stateful auto-configuration.</td>
<td>Local: 68</td>
<td>UDP</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dynamic Host Configuration Protocol for IPv6(DHCPV6-In)</td>
<td>Allows DHCPV6 (Dynamic Host Configuration Protocol for IPv6) messages for stateful and stateless configuration.</td>
<td>Local: 546</td>
<td>UDP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 547</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 547</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Core Networking - Group Policy (LSASS-Out)</td>
<td>Outbound rule to allow remote LSASS traffic for Group Policy updates.</td>
<td>Local: Any</td>
<td>TCP</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Core Networking - Group Policy (NP-Out)</td>
<td>Core Networking - Group Policy (NP-Out)</td>
<td>Local: Any</td>
<td>TCP</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 445</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>Rule</td>
<td>Definition</td>
<td>Port</td>
<td>Protocol</td>
<td>Direction</td>
</tr>
<tr>
<td>----</td>
<td>------</td>
<td>------------</td>
<td>------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td>Core Networking - Group Policy (TCP-Out)</td>
<td>Outbound rule to allow remote RPC traffic for Group Policy updates.</td>
<td>Local: Any Remote: Any</td>
<td>TCP</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>Internet Group Management Protocol (IGMP-In)</td>
<td>IGMP messages are sent and received by nodes to create, join, and depart multicast groups.</td>
<td>Local: 68 Remote: 67</td>
<td>TCP</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>Core Networking - Internet Group Management Protocol (IGMP-Out)</td>
<td>IGMP messages are sent and received by nodes to create, join, and depart multicast groups.</td>
<td>Local: 68 Remote: 67</td>
<td>TCP</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>Core Networking - IPHTTPS (TCP-In)</td>
<td>Inbound TCP rule to allow IPHTTPS tunneling technology to provide connectivity across HTTP proxies and firewalls.</td>
<td>Local: IPHTTPS Remote: Any</td>
<td>TCP</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>Core Networking - IPHTTPS (TCP-Out)</td>
<td>Outbound TCP rule to allow IPHTTPS tunneling technology to provide connectivity across HTTP proxies and firewalls.</td>
<td>Local: Any Remote: IPHTTPS</td>
<td>TCP</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>Rule</td>
<td>Definition</td>
<td>Port</td>
<td>Protocol</td>
<td>Direction</td>
</tr>
<tr>
<td>----</td>
<td>------</td>
<td>------------</td>
<td>------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td>IPv6 (IPv6-In)</td>
<td>Inbound rule required to permit IPv6 traffic for ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) and 6to4 tunneling services.</td>
<td>Local: Any Remote: 445</td>
<td>41</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>IPv6 (IPv6-Out)</td>
<td>Outbound rule required to permit IPv6 traffic for ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) and 6to4 tunneling services.</td>
<td>Local: Any Remote: 445</td>
<td>41</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>Multicast Listener Done (ICMPv6-In)</td>
<td>Multicast Listener Done messages inform local routers that there are no longer any members remaining for a specific multicast address on the subnet.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>Multicast Listener Done (ICMPv6-Out)</td>
<td>Multicast Listener Done messages inform local routers that there are no longer any members remaining for a specific multicast address on the subnet.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>Rule</td>
<td>Definition</td>
<td>Port</td>
<td>Protocol</td>
<td>Direction</td>
</tr>
<tr>
<td>----</td>
<td>------</td>
<td>------------</td>
<td>------</td>
<td>----------</td>
<td>-----------</td>
</tr>
</tbody>
</table>
|    | Multicast Listener Query (ICMPv6-In) | An IPv6 multicast-capable router uses the Multicast Listener Query message to query a link for multicast group membership. | Local: 68  
Remote: 67 | ICMPv6 | In |
|    | Multicast Listener Query (ICMPv6-Out) | An IPv6 multicast-capable router uses the Multicast Listener Query message to query a link for multicast group membership. | Local: 68  
Remote: 67 | ICMPv6 | Out |
|    | Multicast Listener Report (ICMPv6-In) | The Multicast Listener Report message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address or in response to a Multicast Listener Query. | Local: 68  
Remote: 67 | ICMPv6 | In |
<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Multicast Listener Report (ICMPv6-Out)</td>
<td>The Multicast Listener Report message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address or in response to a Multicast Listener Query.</td>
<td>Local: 68&lt;br&gt;Remote: 67</td>
<td>ICMPv6</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td>Multicast Listener Report v2 (ICMPv6-In)</td>
<td>Multicast Listener Report v2 message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address or in response to a Multicast Listener Query.</td>
<td>Local: 68&lt;br&gt;Remote: 67</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td>OS</td>
<td>Rule</td>
<td>Definition</td>
<td>Port</td>
<td>Protocol</td>
<td>Direction</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------</td>
<td>----------------------------------------------------------------------------</td>
<td>---------------</td>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>Multicast Listener Report v2 (ICMPv6-Out)</td>
<td>Multicast Listener Report v2 message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address or in response to a Multicast Listener Query.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td>Neighbor Discovery Advertisement (ICMPv6-In)</td>
<td>Neighbor Discovery Advertisement messages are sent by nodes to notify other nodes of link-layer address changes or in response to a Neighbor Discovery Solicitation request.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>Neighbor Discovery Advertisement (ICMPv6-Out)</td>
<td>Neighbor Discovery Advertisement messages are sent by nodes to notify other nodes of link-layer address changes or in response to a Neighbor Discovery Solicitation request.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>Out</td>
</tr>
<tr>
<td>OS</td>
<td>Rule</td>
<td>Definition</td>
<td>Port</td>
<td>Protocol</td>
<td>Direction</td>
</tr>
<tr>
<td>----</td>
<td>------</td>
<td>------------</td>
<td>------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>Neighbor Discovery Solicitation (ICMPv6-In)</td>
<td>Neighbor Discovery Solicitations are sent by nodes to discover the link-layer address of another on-link IPv6 node.</td>
<td>Local: 68&lt;br&gt;Remote: 67</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>Neighbor Discovery Solicitation (ICMPv6-Out)</td>
<td>Neighbor Discovery Solicitations are sent by nodes to discover the link-layer address of another on-link IPv6 node.</td>
<td>Local: 68&lt;br&gt;Remote: 67</td>
<td>ICMPv6</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td>Packet Too Big (ICMPv6-In)</td>
<td>Packet Too Big error messages are sent from any node that a packet traverses which is unable to forward the packet because the packet is too large for the next link.</td>
<td>Local: 68&lt;br&gt;Remote: 67</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>Packet Too Big (ICMPv6-Out)</td>
<td>Packet Too Big error messages are sent from any node that a packet traverses which is unable to forward the packet because the packet is too large for the next link.</td>
<td>Local: 68&lt;br&gt;Remote: 67</td>
<td>ICMPv6</td>
<td>Out</td>
</tr>
<tr>
<td>OS</td>
<td>Rule</td>
<td>Definition</td>
<td>Port</td>
<td>Protocol</td>
<td>Direction</td>
</tr>
<tr>
<td>----</td>
<td>------</td>
<td>------------</td>
<td>------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>Parameter Problem (ICMPv6-In)</td>
<td>Parameter Problem error messages are sent by nodes when packets are incorrectly generated.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>Parameter Problem (ICMPv6-Out)</td>
<td>Parameter Problem error messages are sent by nodes when packets are incorrectly generated.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td>Router Advertisement (ICMPv6-In)</td>
<td>Router Advertisement messages are sent by routers to other nodes for stateless auto-configuration.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>Router Advertisement (ICMPv6-Out)</td>
<td>Router Advertisement messages are sent by routers to other nodes for stateless auto-configuration.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td>Router Solicitation (ICMPv6-In)</td>
<td>Router Solicitation messages are sent by nodes seeking routers to provide stateless auto-configuration.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>Router Solicitation (ICMPv6-Out)</td>
<td>Router Solicitation messages are sent by nodes seeking routers to provide stateless auto-configuration.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>Out</td>
</tr>
</tbody>
</table>
### Core Networking

**Core Networking - Teredo (UDP-In)**

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inbound UDP rule to allow Teredo edge traversal. This technology provides address assignment and automatic tunneling for unicast IPv6 traffic when an IPv6/IPv4 host is located behind an IPv4 network address translator.</td>
<td>Local: Teredo</td>
<td>UDP</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Core Networking - Teredo (UDP-Out)**

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Outbound UDP rule to allow Teredo edge traversal. This technology provides address assignment and automatic tunneling for unicast IPv6 traffic when an IPv6/IPv4 host is located behind an IPv4 network address translator.</td>
<td>Local: Any</td>
<td>UDP</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Time Exceeded (ICMPv6-In)**

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time Exceeded error messages are generated from any node that a packet traverses if the Hop Limit value is decremented to zero at any point on the path.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Time Exceeded (ICMPv6-Out)

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2008 R2</td>
<td>Time Exceeded error messages are generated from any node that a packet traverses if the Hop Limit value is decremented to zero at any point on the path.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

### Destination Unreachable (ICMPv6-In)

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2008 R2</td>
<td>Destination Unreachable error messages are sent from any node that a packet traverses which is unable to forward the packet for any reason except congestion.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>Windows Server 2008 SP2</td>
<td>Destination Unreachable Fragmentation Needed error messages are sent from any node that a packet traverses which is unable to forward the packet because fragmentation was needed and the don't fragment bit was set.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv4</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

Windows Server 2008 R2 and SP2
<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Host Configuration Protocol</td>
<td>Dynamic Host Configuration Protocol (DHCP-In)</td>
<td>Allows DHCP (Dynamic Host Configuration Protocol) messages for stateful auto-configuration.</td>
<td>Local: 68</td>
<td>UDP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dynamic Host Configuration Protocol (DHCP-Out)</td>
<td>Allows DHCP (Dynamic Host Configuration Protocol) messages for stateful auto-configuration.</td>
<td>Local: 68</td>
<td>UDP</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic Host Configuration Protocol for IPv6 (DHCPV6-In)</td>
<td>Allows DHCPV6 (Dynamic Host Configuration Protocol for IPv6) messages for stateful and stateless configuration.</td>
<td>Local: 546</td>
<td>UDP</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 547</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 547</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internet Group Management Protocol</td>
<td>Internet Group Management Protocol (IGMP-In)</td>
<td>IGMP messages are sent and received by nodes to create, join, and depart multicast groups.</td>
<td>Local: 68</td>
<td>2</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>Rule</td>
<td>Definition</td>
<td>Port</td>
<td>Protocol</td>
<td>Direction</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>--------------------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td>IPv6 (IPv6-In)</td>
<td>Inbound rule</td>
<td>required to permit IPv6 traffic for ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) and 6to4 tunneling services.</td>
<td>Local: Any</td>
<td>41</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 445</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPv6 (IPv6-Out)</td>
<td>Outbound rule</td>
<td>required to permit IPv6 traffic for ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) and 6to4 tunneling services.</td>
<td>Local: Any</td>
<td>41</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 445</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multicast Listener Done (ICMPv6-In)</td>
<td>Multicast Listener Done messages inform local routers that there are no longer any members remaining for a specific multicast address on the subnet.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multicast Listener Done (ICMPv6-Out)</td>
<td>Multicast Listener Done messages inform local routers that there are no longer any members remaining for a specific multicast address on the subnet.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>Rule</td>
<td>Definition</td>
<td>Port</td>
<td>Protocol</td>
<td>Direction</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>----------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>Multicast Listener Query (ICMPv6-In)</td>
<td>An IPv6 multicast-capable router uses the Multicast Listener Query message to query a link for multicast group membership.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>Multicast Listener Query (ICMPv6-Out)</td>
<td>An IPv6 multicast-capable router uses the Multicast Listener Query message to query a link for multicast group membership.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td>Multicast Listener Report (ICMPv6-In)</td>
<td>The Multicast Listener Report message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address, or in response to a Multicast Listener Query.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td>OS</td>
<td>Rule</td>
<td>Definition</td>
<td>Port</td>
<td>Protocol</td>
<td>Direction</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>Multicast Listener Report (ICMPv6-Out)</td>
<td>The Multicast Listener Report message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address, or in response to a Multicast Listener Query.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Multicast Listener Report v2 (ICMPv6-In)</td>
<td>Multicast Listener Report v2 message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address, or in response to a Multicast Listener Query.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Multicast Listener Report v2 (ICMPv6-Out)

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Multicast Listener Report v2 message is used by a listening node to either immediately report its interest in receiving multicast traffic at a specific multicast address, or in response to a Multicast Listener Query.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

### Neighbor Discovery Advertisement (ICMPv6-In)

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Neighbor Discovery Advertisement messages are sent by nodes to notify other nodes of link-layer address changes or in response to a Neighbor Discovery Solicitation request.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

### Neighbor Discovery Advertisement (ICMPv6-Out)

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Neighbor Discovery Advertisement messages are sent by nodes to notify other nodes of link-layer address changes or in response to a Neighbor Discovery Solicitation request.</td>
<td>Local: 68 Remote: 67</td>
<td>ICMPv6</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>Rule</td>
<td>Definition</td>
<td>Port</td>
<td>Protocol</td>
<td>Direction</td>
</tr>
<tr>
<td>----</td>
<td>------</td>
<td>------------</td>
<td>------</td>
<td>----------</td>
<td>-----------</td>
</tr>
</tbody>
</table>
|    | Neighbor Discovery Solicitation (ICMPv6-In) | Neighbor Discovery Solicitations are sent by nodes to discover the link-layer address of another on-link IPv6 node. | Local: 68  
Remote: 67 | ICMPv6 | In |
|    | Neighbor Discovery Solicitation (ICMPv6-Out) | Neighbor Discovery Solicitations are sent by nodes to discover the link-layer address of another on-link IPv6 node. | Local: 68  
Remote: 67 | ICMPv6 | Out |
|    | Packet Too Big (ICMPv6-In) | Packet Too Big error messages are sent from any node that a packet traverses which is unable to forward the packet because the packet is too large for the next link. | Local: 68  
Remote: 67 | ICMPv6 | In |
|    | Packet Too Big (ICMPv6-Out) | Packet Too Big error messages are sent from any node that a packet traverses which is unable to forward the packet because the packet is too large for the next link. | Local: 68  
Remote: 67 | ICMPv6 | Out |
<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Parameter Problem (ICMPv6-In)</td>
<td>Parameter Problem error messages are sent by nodes when packets are incorrectly generated.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Parameter Problem (ICMPv6-Out)</td>
<td>Parameter Problem error messages are sent by nodes when packets are incorrectly generated.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router Advertisement (ICMPv6-In)</td>
<td>Router Advertisement messages are sent by routers to other nodes for stateless auto-configuration.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router Advertisement (ICMPv6-Out)</td>
<td>Router Advertisement messages are sent by routers to other nodes for stateless auto-configuration.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router Solicitation (ICMPv6-In)</td>
<td>Router Solicitation messages are sent by nodes seeking routers to provide stateless auto-configuration.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router Solicitation (ICMPv6-Out)</td>
<td>Router Solicitation messages are sent by nodes seeking routers to provide stateless auto-configuration.</td>
<td>Local: 68</td>
<td>ICMPv6</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 67</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Delivery Optimization

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2019</td>
<td>DeliveryOptimization-TCP-In</td>
<td>Inbound rule to allow Delivery Optimization to connect to remote endpoints.</td>
<td>Local: 7680</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DeliveryOptimization-UDP-In</td>
<td>Inbound rule to allow Delivery Optimization to connect to remote endpoints.</td>
<td>Local: 7680</td>
<td>UDP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Time Exceeded (ICMPv6-In)**

Messages are generated from any node that a packet traverses if the Hop Limit value is decremented to zero at any point on the path.

Local: 68  
Remote: 67

**Time Exceeded (ICMPv6-Out)**

Messages are generated from any node that a packet traverses if the Hop Limit value is decremented to zero at any point on the path.

Local: 68  
Remote: 67

**ICMPv6**

In

Out
## Diag Track

### Windows Server 2019

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>and Telemetry</td>
<td></td>
<td>Remote: 443</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Windows Server 2016

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2016</td>
<td>Connected User Experiences</td>
<td>Unified Telemetry Client Outbound Traffic</td>
<td>Local: Any</td>
<td>TCP</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td>and Telemetry</td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## DIAL Protocol Server

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2016</td>
<td>DIAL protocol server (HTTP-</td>
<td>Inbound rule for DIAL protocol server to allow remote</td>
<td>Local: 10247</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>In)</td>
<td>control of Apps using HTTP.</td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2019</td>
<td>DIAL protocol server (HTTP-</td>
<td>Inbound rule for DIAL protocol server to allow remote</td>
<td>Local: 10247</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td>In)</td>
<td>control of Apps using HTTP.</td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Distributed File System (DFS) Management

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2008</td>
<td>DFS Management (SMB-In)</td>
<td>Inbound rule to allow SMB traffic to manage the File Services role.</td>
<td>Local: 445</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td>R2</td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DFS Management</td>
<td></td>
<td>Inbound rule to allow WMI traffic to manage the</td>
<td>Local: RPC</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td>(WMI-In)</td>
<td></td>
<td>File Services role.</td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>Rule</td>
<td>Definition</td>
<td>Port</td>
<td>Protocol</td>
<td>Direction</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-----------------------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>File and Printer Sharing (Echo Request - ICMPv4-In)</td>
<td>Echo Request messages are sent as ping requests to other nodes.</td>
<td>Local: 5355</td>
<td>ICMPv4</td>
<td>In</td>
</tr>
<tr>
<td>Windows Server 2008 SP2</td>
<td>File and Printer Sharing (Echo Request - ICMPv4-Out)</td>
<td>Echo Request messages are sent as ping requests to other nodes.</td>
<td>Local: 5355</td>
<td>ICMPv4</td>
<td>Out</td>
</tr>
<tr>
<td>Windows Server 2012</td>
<td>File and Printer Sharing (Echo Request - ICMPv6-In)</td>
<td>Echo Request messages are sent as ping requests to other nodes.</td>
<td>Local: 5355</td>
<td>ICMPv6</td>
<td>In</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td>File and Printer Sharing (Echo Request - ICMPv6-Out)</td>
<td>Echo Request messages are sent as ping requests to other nodes.</td>
<td>Local: 5355</td>
<td>ICMPv6</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td>File and Printer Sharing (LLMNR-UDP-In)</td>
<td>Inbound rule for File and Printer Sharing to allow Link Local Multicast Name Resolution.</td>
<td>Local: 5355</td>
<td>UDP</td>
<td>In</td>
</tr>
</tbody>
</table>

**File and Printer Sharing**

- **DFS Management (DCOM-In)**: Inbound rule to allow DCOM traffic to manage the File Services role.
  - Local: 135
  - Remote: Any
  - Protocol: TCP
  - Direction: In

- **DFS Management (TCP-In)**: Inbound rule to allow TCP traffic to manage the File Services role.
  - Local: RPC
  - Remote: Any
  - Protocol: TCP
  - Direction: In
<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>File and Printer Sharing (LLMNR-UDP-Out)</td>
<td>Outbound rule for File and Printer Sharing to allow Link Local Multicast Name Resolution.</td>
<td>Local: Any Remote: 5355</td>
<td>UDP</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>File and Printer Sharing (NB-Datagram-In)</td>
<td>Inbound rule for File and Printer Sharing to allow NetBIOS Datagram transmission and reception.</td>
<td>Local: 138 Remote: Any</td>
<td>UDP</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>File and Printer Sharing (NB-Datagram-Out)</td>
<td>Outbound rule for File and Printer Sharing to allow NetBIOS Datagram transmission and reception.</td>
<td>Local: Any Remote: 138</td>
<td>UDP</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>File and Printer Sharing (NB-Name-In)</td>
<td>Inbound rule for File and Printer Sharing to allow NetBIOS Name Resolution.</td>
<td>Local: 137 Remote: Any</td>
<td>UDP</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>File and Printer Sharing (NB-Name-Out)</td>
<td>Outbound rule for File and Printer Sharing to allow NetBIOS Name Resolution.</td>
<td>Local: Any Remote: 137</td>
<td>UDP</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>File and Printer Sharing (NB-Session-In)</td>
<td>Inbound rule for File and Printer Sharing to allow NetBIOS Session Service connections.</td>
<td>Local: 139 Remote: Any</td>
<td>TCP</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>File and Printer Sharing (NB-Session-Out)</td>
<td>Outbound rule for File and Printer Sharing to allow NetBIOS Session Service connections.</td>
<td>Local: Any Remote: 139</td>
<td>TCP</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>
### File and Printer Sharing (SMB-In)

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>File and Printer Sharing (SMB-In)</td>
<td>Inbound rule for File and Printer Sharing to allow Server Message Block transmission and reception via Named Pipes.</td>
<td>Local: 445 Remote: Any</td>
<td>TCP</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

### File and Printer Sharing (SMB-Out)

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
</table>

### File and Printer Sharing (Spooler Service - RPC)

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>File and Printer Sharing (Spooler Service - RPC)</td>
<td>Inbound rule for File and Printer Sharing to allow the Print Spooler Service to communicate via TCP/RPC.</td>
<td>Local: RPC Remote: Any</td>
<td>TCP</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

### File and Printer Sharing (Spooler Service - RPC-EPMap)

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>File and Printer Sharing (Spooler Service - RPC-EPMap)</td>
<td>Inbound rule for the RPCSS service to allow RPC/TCP traffic for the Spooler Service.</td>
<td>Local: RPC-EPMap Remote: Any</td>
<td>TCP</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

### File Server Remote Management

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2008 SP2</td>
<td>File Server Remote Management (DCOM-In)</td>
<td>Inbound rule to allow DCOM traffic to manage the File Services role.</td>
<td>Local: 135 Remote: Any</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td>Windows Server 2012</td>
<td>File Server Remote Management (SMB-In)</td>
<td>Inbound rule to allow SMB traffic to manage the</td>
<td>Local: 445 Remote: Any</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>Rule</td>
<td>Definition</td>
<td>Port</td>
<td>Protocol</td>
<td>Direction</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>File Services role.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WMI-In</td>
<td>Inbound</td>
<td>rule to allow WMI traffic to manage the File Services role.</td>
<td>Local: RPC</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## ICMP v4 All

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2012</td>
<td>All ICMP v4</td>
<td>Local: 139</td>
<td>ICMPv4</td>
<td>In</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td>All ICMP v4</td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Multicast

**Windows Server 2019**

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2019</td>
<td>mDNS (UDP-In)</td>
<td>Inbound rule for mDNS traffic.</td>
<td>Local: 5353</td>
<td>UDP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mDNS (UDP-Out)</td>
<td>Outbound rule for mDNS traffic.</td>
<td>Local: Any</td>
<td>UDP</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: 5353</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Windows Server 2016**

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2016</td>
<td>mDNS (UDP-In)</td>
<td>Inbound rule for mDNS traffic.</td>
<td>Local: mDNS</td>
<td>UDP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mDNS (UDP-Out)</td>
<td>Outbound rule for mDNS traffic.</td>
<td>Local: 5353</td>
<td>UDP</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Remote Desktop

Windows Server 2012 R2, 2016, and 2019

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2012 R2</td>
<td>Remote Desktop - Shadow (TCP-In)</td>
<td>Inbound rule for the Remote Desktop service to allow shadowing of an existing Remote Desktop session.</td>
<td>Local: Any</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td>Windows Server 2016</td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2019</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remote Desktop - User Mode (TCP-In)</td>
<td>Inbound rule for the Remote Desktop service to allow RDP traffic.</td>
<td>Local: 3389</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remote Desktop - User Mode (UDP-In)</td>
<td>Inbound rule for the Remote Desktop service to allow RDP traffic.</td>
<td>Local: 3389</td>
<td>UDP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Windows Server 2012

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2012</td>
<td>Remote Desktop - User Mode (TCP-In)</td>
<td>Inbound rule for the Remote Desktop service to allow RDP traffic.</td>
<td>Local: 3389</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remote Desktop - User Mode (UDP-In)</td>
<td>Inbound rule for the Remote Desktop service to allow RDP traffic.</td>
<td>Local: 3389</td>
<td>UDP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Windows Server 2008 SP2

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2008 SP2</td>
<td>Remote Desktop - Shadow (TCP-In)</td>
<td>Inbound rule for the Remote Desktop service to allow shadowing of an existing Remote Desktop session.</td>
<td>Local: Any</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remote Desktop - User Mode (TCP-In)</td>
<td>Inbound rule for the Remote Desktop service to allow RDP traffic.</td>
<td>Local: 3389</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remote Desktop - User Mode (UDP-In)</td>
<td>Inbound rule for the Remote Desktop service to allow RDP traffic.</td>
<td>Local: 3389</td>
<td>UDP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Windows Server 2008 R2

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2008 R2</td>
<td>RemoteFX (TCP-In)</td>
<td>Inbound rule for the Remote Desktop service to allow RDP traffic.</td>
<td>Local: 3389</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP-In</td>
<td>Inbound rule for the Remote Desktop service to allow RDP traffic.</td>
<td>Local: 3389</td>
<td>TCP</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Windows Device Management

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Enrollment WinRT</td>
<td>Windows Enrollment WinRT (TCP Out)</td>
<td>Allow outbound TCP traffic from Windows Enrollment WinRT.</td>
<td>Local: Any</td>
<td>TCP</td>
<td>Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Windows Firewall Remote Management

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2008 SP2</td>
<td>Windows Firewall Remote Management (RPC)</td>
<td>Inbound rule for the Windows Firewall to be remotely managed via RPC/TCP.</td>
<td>Local: RPC</td>
<td>TCP</td>
<td>In</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td>Windows Firewall Remote Management (RPC-EPMAP)</td>
<td>Inbound rule for the RPCSS service to allow RPC/TCP traffic for RPC-EPMAP</td>
<td>Local: RPC-EPMAP</td>
<td>TCP</td>
<td>In</td>
</tr>
</tbody>
</table>
### Windows Remote Management

<table>
<thead>
<tr>
<th>OS</th>
<th>Rule</th>
<th>Definition</th>
<th>Port</th>
<th>Protocol</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(HTTP-In)</td>
<td></td>
<td>Remote: Any</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For more information about Amazon EC2 security groups, see Amazon EC2 Security Groups for Windows Instances.

### 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.

### Detect 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.

<table>
<thead>
<tr>
<th>Account Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported Platforms</td>
</tr>
<tr>
<td>EC2</td>
</tr>
<tr>
<td>VPC</td>
</tr>
</tbody>
</table>

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`.

<table>
<thead>
<tr>
<th>Account Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported Platforms</td>
</tr>
<tr>
<td>VPC</td>
</tr>
</tbody>
</table>

Default VPC

vpc-1a2b3c4d

The output of the `describe-account-attributes` command for the specified Region includes only the VPC value for the `supported-platforms` attribute.

```
aws ec2 describe-account-attributes --attribute-names supported-platforms --region us-east-2
{
    "AccountAttributes": [
        {
            "AttributeName": "supported-platforms",
            "AttributeValues": [
                {
                    "AttributeValue": "VPC"
                }
            ]
        }
    ]
}
```
Instance types available in EC2-Classic

Most of the newer instance types require a VPC. The following are the only instance types supported in EC2-Classic:

- General purpose: M1, M3, and T1
- Compute optimized: C1, C3, and CC2
- Memory optimized: CR1, M2, and R3
- Storage optimized: D2, HS1, and I2
- Accelerated computing: G2

If your account supports EC2-Classic but you have not created a nondefault VPC, you can do one of the following to launch instances that require a VPC:

- Create a nondefault VPC and launch your VPC-only instance into it by specifying a subnet ID or a network interface ID in the request. Note that you must create a nondefault VPC if you do not have a default VPC and you are using the AWS CLI, Amazon EC2 API, or AWS SDK to launch a VPC-only instance.
- Launch your VPC-only instance using the Amazon EC2 console. The Amazon EC2 console creates a nondefault VPC in your account and launches the instance into the subnet in the first Availability Zone. The console creates the VPC with the following attributes:
  - One subnet in each Availability Zone, with the public IPv4 addressing attribute set to true so that instances receive a public IPv4 address. For more information, see IP Addressing in Your VPC in the Amazon VPC User Guide.
  - An Internet gateway, and a main route table that routes traffic in the VPC to the Internet gateway. This enables the instances you launch in the VPC to communicate over the Internet. For more information, see Internet Gateways in the Amazon VPC User Guide.
  - A default security group for the VPC and a default network ACL that is associated with each subnet. For more information, see Security Groups for Your VPC in the Amazon VPC User Guide.

If you have other resources in EC2-Classic, you can take steps to migrate them to a VPC. For more information, see Migrate from EC2-Classic to a VPC (p. 1041).

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.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>EC2-Classic</th>
<th>Default VPC</th>
<th>Nondefault VPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public IPv4 address</td>
<td>Your instance receives a public IPv4 address from the EC2-Classic public IPv4 address pool.</td>
<td>Your instance launched in a default subnet receives a public IPv4 address by default, unless you specify otherwise during launch, or you modify</td>
<td>Your instance doesn't receive a public IPv4 address by default, unless you specify otherwise during launch, or you modify</td>
</tr>
</tbody>
</table>
### Differences between instances in EC2-Classic and a VPC

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>EC2-Classic</th>
<th>Default VPC</th>
<th>Nondefault VPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Private IPv4 address</td>
<td>Your instance receives a private IPv4 address from the EC2-Classic range each time it's started.</td>
<td>Your instance receives a static private IPv4 address from the address range of your default VPC.</td>
<td>Your instance receives a static private IPv4 address from the address range of your VPC.</td>
</tr>
<tr>
<td>Multiple private IPv4 addresses</td>
<td>We select a single private IP address for your instance; multiple IP addresses are not supported.</td>
<td>You can assign multiple private IPv4 addresses to your instance.</td>
<td>You can assign multiple private IPv4 addresses to your instance.</td>
</tr>
<tr>
<td>Elastic IP address (IPv4)</td>
<td>An Elastic IP is disassociated from your instance when you stop it.</td>
<td>An Elastic IP remains associated with your instance when you stop it.</td>
<td>An Elastic IP remains associated with your instance when you stop it.</td>
</tr>
<tr>
<td>Associating an Elastic IP address</td>
<td>You associate an Elastic IP address with an instance.</td>
<td>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.</td>
<td>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.</td>
</tr>
<tr>
<td>Reassociating an Elastic IP address</td>
<td>If the Elastic IP address is already associated with another instance, the address is automatically associated with the new instance.</td>
<td>If the Elastic IP address is already associated with another instance, the address is automatically associated with the new instance.</td>
<td>If the Elastic IP address is already associated with another instance, it succeeds only if you allowed reassociation.</td>
</tr>
<tr>
<td>Tagging Elastic IP addresses</td>
<td>You cannot apply tags to an Elastic IP address.</td>
<td>You can apply tags to an Elastic IP address.</td>
<td>You can apply tags to an Elastic IP address.</td>
</tr>
<tr>
<td>DNS hostnames</td>
<td>DNS hostnames are enabled by default.</td>
<td>DNS hostnames are enabled by default.</td>
<td>DNS hostnames are disabled by default.</td>
</tr>
<tr>
<td>Security group</td>
<td>A security group can reference security groups that belong to other AWS accounts.</td>
<td>A security group can reference security groups for your VPC, or for a peer VPC in a VPC peering connection.</td>
<td>A security group can reference security groups for your VPC only.</td>
</tr>
</tbody>
</table>
## Differences between instances in EC2-Classic and a VPC

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>EC2-Classic</th>
<th>Default VPC</th>
<th>Nondefault VPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security group association</td>
<td>You can't change the security groups of your running instance. You can</td>
<td>You can assign up to 5 security groups to an instance.</td>
<td>You can assign up to 5 security groups to an instance.</td>
</tr>
<tr>
<td></td>
<td>either modify the rules of the assigned security groups, or replace the</td>
<td>You can assign security groups to your instance when you launch it and</td>
<td>You can assign security groups to your instance when you launch it and</td>
</tr>
<tr>
<td></td>
<td>instance with a new one (create an AMI from the instance, launch a new</td>
<td>while it's running.</td>
<td>while it's running.</td>
</tr>
<tr>
<td></td>
<td>instance from this AMI with the security groups that you need, disassociate</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>any Elastic IP address from the original instance and associate it with the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>new instance, and then terminate the original instance).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security group rules</td>
<td>You can add rules for inbound traffic only.</td>
<td>You can add rules for inbound and outbound traffic.</td>
<td>You can add rules for inbound and outbound traffic.</td>
</tr>
<tr>
<td>Tenancy</td>
<td>Your instance runs on shared hardware.</td>
<td>You can run your instance on shared hardware or single-tenant hardware.</td>
<td>You can run your instance on shared hardware or single-tenant hardware.</td>
</tr>
<tr>
<td>Accessing the Internet</td>
<td>Your instance can access the Internet. Your instance automatically receives</td>
<td>By default, your instance can access the Internet. Your instance receives</td>
<td>By default, your instance cannot access the Internet. Your instance doesn't</td>
</tr>
<tr>
<td></td>
<td>a public IP address, and can access the Internet directly through the AWS</td>
<td>a public IP address by default. An Internet gateway is attached to your</td>
<td>receive a public IP address by default. Your VPC may have an Internet</td>
</tr>
<tr>
<td></td>
<td>network edge.</td>
<td>default VPC, and your default subnet has a route to the Internet gateway.</td>
<td>gateway, depending on how it was created.</td>
</tr>
<tr>
<td>IPv6 addressing</td>
<td>IPv6 addressing is not supported. You cannot assign IPv6 addresses to your</td>
<td>You can optionally associate an IPv6 CIDR block with your VPC, and assign</td>
<td>You can optionally associate an IPv6 CIDR block with your VPC, and assign</td>
</tr>
<tr>
<td></td>
<td>instances.</td>
<td>IPv6 addresses to instances in your VPC.</td>
<td>IPv6 addresses to instances in your VPC.</td>
</tr>
</tbody>
</table>

### Security groups for EC2-Classic

If you're using EC2-Classic, you must use security groups created specifically for EC2-Classic. When you launch an instance in EC2-Classic, you must specify a security group in the same Region as the instance. You can't specify a security group that you created for a VPC when you launch an instance in EC2-Classic.

After you launch an instance in EC2-Classic, you can't change its security groups. However, you can add rules to or remove rules from a security group, and those changes are automatically applied to all instances that are associated with the security group after a short period.
Your AWS account automatically has a default security group per Region for EC2-Classic. If you try to delete the default security group, you'll get the following error: Client.InvalidGroup.Reserved: The security group 'default' is reserved.

You can create custom security groups. The security group name must be unique within your account for the Region. To create a security group for use in EC2-Classic, choose No VPC for the VPC.

You can add inbound rules to your default and custom security groups. You can't change the outbound rules for an EC2-Classic security group. When you create a security group rule, you can use a different security group for EC2-Classic in the same Region as the source or destination. To specify a security group for another AWS account, add the AWS account ID as a prefix; for example, 111122223333/sg-edcd9784.

In EC2-Classic, you can have up to 500 security groups in each Region for each account. You can add up to 100 rules to a security group. You can have up to 800 security group rules per instance. This is calculated as the multiple of rules per security group and security groups per instance. If you reference other security groups in your security group rules, we recommend that you use security group names that are 22 characters or less in length.

**IP addressing and DNS**

Amazon provides a DNS server that resolves Amazon-provided IPv4 DNS hostnames to IPv4 addresses. In EC2-Classic, the Amazon DNS server is located at 172.16.0.23.

If you create a custom firewall configuration in EC2-Classic, you must create a rule in your firewall that allows inbound traffic from port 53 (DNS)—with a destination port from the ephemeral range—from the address of the Amazon DNS server; otherwise, internal DNS resolution from your instances fails. If your firewall doesn't automatically allow DNS query responses, then you need to allow traffic from the IP address of the Amazon DNS server. To get the IP address of the Amazon DNS server, use the following command from within your instance:

```
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.

**Migrate 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 Disassociate an Elastic IP address (p. 931).

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. 934).

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 Access 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 Access 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 Access Amazon EC2 (p. 3).

- `describe-moving-addresses` (AWS CLI)
- `Get-EC2Address` (AWS Tools for Windows PowerShell)

Share and access 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. 1030).

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 Migrate from EC2-Classic to a VPC (p. 1041).

The following resources can be shared or accessed between EC2-Classic and a VPC.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMI</td>
<td></td>
</tr>
<tr>
<td>Bundle task</td>
<td></td>
</tr>
<tr>
<td>EBS volume</td>
<td></td>
</tr>
<tr>
<td>Elastic IP address (IPv4)</td>
<td>You can migrate an Elastic IP address from EC2-Classic to a VPC. You can't migrate an Elastic IP address that was originally allocated for use in a VPC to EC2-Classic. For more information, see Migrate an Elastic IP Address from EC2-Classic (p. 1027).</td>
</tr>
<tr>
<td>Instance</td>
<td>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 Migrate from EC2-Classic to a VPC (p. 1041).</td>
</tr>
<tr>
<td>Key pair</td>
<td></td>
</tr>
</tbody>
</table>
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 Modify Reserved Instances (p. 273).

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 Create a security group (p. 1141).

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 Migrate from EC2-Classic to a VPC (p. 1041).

There is no additional charge for using ClassicLink. Standard charges for data transfer and instance usage apply.

## Contents

- ClassicLink basics (p. 1031)
- ClassicLink limitations (p. 1033)
- Work with ClassicLink (p. 1034)
- Example IAM policies for ClassicLink (p. 1037)
ClassicLink basics

There are two steps to linking an EC2-Classic instance to a VPC using ClassicLink. First, you must enable the VPC for ClassicLink. By default, all VPCs in your account are not enabled for ClassicLink, to maintain their isolation. After you've enabled the VPC for ClassicLink, you can then link any running EC2-Classic instance in the same Region in your account to that VPC. Linking your instance includes selecting security groups from the VPC to associate with your EC2-Classic instance. After you've linked the instance, it can communicate with instances in your VPC using their private IP addresses, provided the VPC security groups allow it. Your EC2-Classic instance does not lose its private IP address when linked to the VPC.

Linking your instance to a VPC is sometimes referred to as attaching your instance.

A linked EC2-Classic instance can communicate with instances in a VPC, but it does not form part of the VPC. If you list your instances and filter by VPC, for example, through the DescribeInstances API request, or by using the Instances screen in the Amazon EC2 console, the results do not return any EC2-Classic instances that are linked to the VPC. For more information about viewing your linked EC2-Classic instances, see View your ClassicLink-enabled VPCs and linked instances (p. 1036).

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 Enable ClassicLink DNS support (p. 1036).

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.

Use 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.

Control 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. 1060).

For more information about policies for working with ClassicLink, see the following example: Example IAM policies for ClassicLink (p. 1037).

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. 1039).

After you've linked your instance to a VPC, you cannot change which VPC security groups are associated with the instance. To associate different security groups with your instance, you must first unlink the instance, and then link it to the VPC again, choosing the required security groups.

Routing for ClassicLink

When you enable a VPC for ClassicLink, a static route is added to all of the VPC route tables with a destination of 10.0.0.0/8 and a target of local. This allows communication between instances in the VPC and any EC2-Classic instances that are then linked to the VPC. If you add a custom route table to a ClassicLink-enabled VPC, a static route is automatically added with a destination of 10.0.0.0/8 and a target of local. When you disable ClassicLink for a VPC, this route is automatically deleted in all of the VPC route tables.

VPCs that are in the 10.0.0.0/16 and 10.1.0.0/16 IP address ranges can be enabled for ClassicLink only if they do not have any existing static routes in route tables in the 10.0.0.0/8 IP address range, excluding the local routes that were automatically added when the VPC was created. Similarly, if you've enabled a VPC for ClassicLink, you may not be able to add any more specific routes to your route tables within the 10.0.0.0/8 IP address range.

Important
If your VPC CIDR block is a publicly routable IP address range, consider the security implications before you link an EC2-Classic instance to your VPC. For example, if your linked EC2-Classic instance receives an incoming Denial of Service (DoS) request flood attack from a source IP
address that falls within the VPC's IP address range, the response traffic is sent into your VPC. We strongly recommend that you create your VPC using a private IP address range as specified in RFC 1918.

For more information about route tables and routing in your VPC, see Route Tables in the Amazon VPC User Guide.

Enable a VPC peering connection for ClassicLink

If you have a VPC peering connection between two VPCs, and there are one or more EC2-Classic instances that are linked to one or both of the VPCs via ClassicLink, you can extend the VPC peering connection to enable communication between the EC2-Classic instances and the instances in the VPC on the other side of the VPC peering connection. This enables the EC2-Classic instances and the instances in the VPC to communicate using private IP addresses. To do this, you can enable a local VPC to communicate with a linked EC2-Classic instance in a peer VPC, or you can enable a local linked EC2-Classic instance to communicate with instances in a peer VPC.

If you enable a local VPC to communicate with a linked EC2-Classic instance in a peer VPC, a static route is automatically added to your route tables with a destination of 10.0.0.0/8 and a target of local.

For more information and examples, see Configurations With ClassicLink in the Amazon VPC Peering Guide.

ClassicLink limitations

To use the ClassicLink feature, you need to be aware of the following limitations:

- You can link an EC2-Classic instance to only one VPC at a time.
- If you stop your linked EC2-Classic instance, it's automatically unlinked from the VPC and the VPC security groups are no longer associated with the instance. You can link your instance to the VPC again after you've restarted it.
- You cannot link an EC2-Classic instance to a VPC that's in a different Region or a different AWS account.
- You cannot use ClassicLink to link a VPC instance to a different VPC, or to a EC2-Classic resource. To establish a private connection between VPCs, you can use a VPC peering connection. For more information, see the Amazon VPC Peering Guide.
- You cannot associate a VPC Elastic IP address with a linked EC2-Classic instance.
- You cannot enable EC2-Classic instances for IPv6 communication. You can associate an IPv6 CIDR block with your VPC and assign IPv6 address to resources in your VPC, however, communication between a ClassicLinked instance and resources in the VPC is over IPv4 only.
- VPCs with routes that conflict with the EC2-Classic private IP address range of 10/8 cannot be enabled for ClassicLink. This does not include VPCs with 10.0.0.0/16 and 10.1.0.0/16 IP address ranges that already have local routes in their route tables. For more information, see Routing for ClassicLink (p. 1032).
- VPCs configured for dedicated hardware tenancy cannot be enabled for ClassicLink. Contact Amazon Web Services 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 Amazon Web Services 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.

Work 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**

- Enable a VPC for ClassicLink (p. 1034)
- Create a VPC with ClassicLink enabled (p. 1034)
- Link an instance to a VPC (p. 1035)
- Link an instance to a VPC at launch (p. 1035)
- View your ClassicLink-enabled VPCs and linked instances (p. 1036)
- Enable ClassicLink DNS support (p. 1036)
- Disable ClassicLink DNS support (p. 1036)
- Unlink an instance from a VPC (p. 1036)
- Disable ClassicLink for a VPC (p. 1037)

**Enable 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. 1032).

**To enable a VPC for ClassicLink**

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose Your VPCs.
3. Select the VPC.
4. Choose Actions, Enable ClassicLink.
5. When prompted for confirmation, choose Enable ClassicLink.
6. (Optional) If you want the public DNS hostname to resolve to the private IP address, enable ClassicLink DNS support for the VPC before you link any instances. For more information, see Enable ClassicLink DNS support (p. 1036).

**Create a VPC with ClassicLink enabled**

You can create a new VPC and immediately enable it for ClassicLink by using the VPC wizard in the Amazon VPC console.
To create a VPC with ClassicLink enabled

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. From the Amazon VPC dashboard, choose Launch VPC Wizard.
3. Select one of the VPC configuration options and choose Select.
4. On the next page of the wizard, choose Yes for Enable ClassicLink. Complete the rest of the steps in the wizard to create your VPC. For more information about using the VPC wizard, see Scenarios for Amazon VPC in the Amazon VPC User Guide.
5. (Optional) If you want the public DNS hostname to resolve to the private IP address, enable ClassicLink DNS support for the VPC before you link any instances. For more information, see Enable ClassicLink DNS support (p. 1036).

Link an instance to a VPC

After you've enabled a VPC for ClassicLink, you can link an EC2-Classic instance to it. The instance must be in the running state.

If you want the public DNS hostname to resolve to the private IP address, enable ClassicLink DNS support for the VPC before you link the instance. For more information, see Enable ClassicLink DNS support (p. 1036).

To link an instance to a VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select one or more running EC2-Classic instances.
4. Choose Actions, ClassicLink, Link to VPC.
5. Choose the VPC. The console displays only VPCs that are enabled for ClassicLink.
6. Select one or more security groups to associate with your instances. The console displays security groups only for VPCs enabled for ClassicLink.
7. Choose Link.

Link an instance to a VPC at launch

You can use the launch wizard in the Amazon EC2 console to launch an EC2-Classic instance and immediately link it to a ClassicLink-enabled VPC.

To link an instance to a VPC at launch

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Amazon EC2 dashboard, choose Launch Instance.
3. Select an AMI, and then choose an instance type that is supported on EC2-Classic. For more information, see Instance types available in EC2-Classic (p. 1024).
4. On the Configure Instance Details page, do the following:
   a. For Network, choose Launch into EC2-Classic. If this option is disabled, then the instance type is not supported on EC2-Classic.
   b. Expand Link to VPC (ClassicLink) and choose a VPC from Link to VPC. The console displays only VPCs with ClassicLink enabled.
5. Complete the rest of the steps in the wizard to launch your instance. For more information, see Launch an instance using the Launch Instance Wizard (p. 392).
View your ClassicLink-enabled VPCs and linked instances

You can view all of your ClassicLink-enabled VPCs in the Amazon VPC console, and your linked EC2-Classic instances in the Amazon EC2 console.

To view your ClassicLink-enabled VPCs
1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose Your VPCs.
3. Select the VPC.
4. If the value of ClassicLink is Enabled, then the VPC is enabled for ClassicLink.

Enable ClassicLink DNS support

You can enable ClassicLink DNS support for your VPC so that DNS hostnames that are addressed between linked EC2-Classic instances and instances in the VPC resolve to private IP addresses and not public IP addresses. For this feature to work, your VPC must be enabled for DNS hostnames and DNS resolution.

Note
If you enable ClassicLink DNS support for your VPC, your linked EC2-Classic instance can access any private hosted zone associated with the VPC. For more information, see Working with Private Hosted Zones in the Amazon Route 53 Developer Guide.

To enable ClassicLink DNS support
1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose Your VPCs.
3. Select the VPC.
5. For ClassicLink DNS support, select Enable.
6. Choose Save changes.

Disable ClassicLink DNS support

You can disable ClassicLink DNS support for your VPC so that DNS hostnames that are addressed between linked EC2-Classic instances and instances in the VPC resolve to public IP addresses and not private IP addresses.

To disable ClassicLink DNS support
1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose Your VPCs.
3. Select the VPC.
5. For ClassicLink DNS Support, clear Enable.
6. Choose Save changes.

Unlink an instance from a VPC

If you no longer require a ClassicLink connection between your EC2-Classic instance and your VPC, you can unlink the instance from the VPC. Unlinking the instance disassociates the VPC security groups from the instance.
A stopped instance is automatically unlinked from a VPC.

To unlink an instance from a VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select one or more of your instances.
4. Choose Actions, ClassicLink, Unlink from VPC.
5. When prompted for confirmation, choose Unlink.

Disable ClassicLink for a VPC

If you no longer require a connection between EC2-Classic instances and your VPC, you can disable ClassicLink on the VPC. You must first unlink all linked EC2-Classic instances that are linked to the VPC.

To disable ClassicLink for a VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose Your VPCs.
3. Select your VPC.
4. Choose Actions, Disable ClassicLink.
5. When prompted for confirmation, choose Disable ClassicLink.

Example IAM policies for ClassicLink

You can enable a VPC for ClassicLink and then link an EC2-Classic instance to the VPC. You can also view your ClassicLink-enabled VPCs, and all of your EC2-Classic instances that are linked to a VPC. You can create policies with resource-level permission for the ec2:EnableVpcClassicLink, ec2:DisableVpcClassicLink, ec2:AttachClassicLinkVpc, and ec2:DetachClassicLinkVpc actions to control how users are able to use those actions. Resource-level permissions are not supported for ec2:Describe* actions.

Examples

- Full permissions to work with ClassicLink (p. 1037)
- Enable and disable a VPC for ClassicLink (p. 1038)
- Link instances (p. 1038)
- Unlink instances (p. 1039)

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.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeClassicLinkInstances", "ec2:DescribeVpcClassicLink",
        "ec2:EnableVpcClassicLink", "ec2:DisableVpcClassicLink",
        "ec2:AttachClassicLinkVpc", "ec2:DetachClassicLinkVpc"
      ]
    }
  ]
}
```
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.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": "ec2:*VpcClassicLink",
         "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.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": "ec2:AttachClassicLinkVpc",
         "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-122aabb 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.
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.

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).

<table>
<thead>
<tr>
<th>Inbound</th>
<th>Source</th>
<th>Type</th>
<th>Port Range</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0.0.0/0</td>
<td>HTTPS</td>
<td>443</td>
<td>Allows Internet traffic to reach your web server.</td>
</tr>
</tbody>
</table>

The following are the security group rules for the VPC security group (sg-2b2b2b2b).

<table>
<thead>
<tr>
<th>Outbound</th>
<th>Destination</th>
<th>Type</th>
<th>Port Range</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sg-3c3c3c3c</td>
<td>TCP</td>
<td>6001</td>
<td>Allows outbound traffic from your web server to your application server in your VPC.</td>
</tr>
</tbody>
</table>
(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.

<table>
<thead>
<tr>
<th>Inbound</th>
<th>Source</th>
<th>Type</th>
<th>Port Range</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sg-2b2b2b2b2b</td>
<td>TCP</td>
<td>6001</td>
<td>Allows the specified type of traffic from your web server (or any other instance associated with sg-2b2b2b2b2b) to reach your application server.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outbound</th>
<th>Destination</th>
<th>Type</th>
<th>Port Range</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sg-4d4d4d4d4d</td>
<td>TCP</td>
<td>6004</td>
<td>Allows outbound traffic from the application server to the database server (or to any other instance associated with sg-4d4d4d4d4d).</td>
</tr>
</tbody>
</table>

Security group for your database server (sg-4d4d4d4d4d)

The following are the security group rules for the VPC security group that's associated with your database server.

<table>
<thead>
<tr>
<th>Inbound</th>
<th>Source</th>
<th>Type</th>
<th>Port Range</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sg-3c3c3c3c3c</td>
<td>TCP</td>
<td>6004</td>
<td>Allows the specified type of traffic from your application server (or any other instance associated with sg-3c3c3c3c3c) to reach your database server.</td>
</tr>
</tbody>
</table>

Migrate from EC2-Classic to a VPC

If you created your AWS account before December 4, 2013, you might have support for EC2-Classic in some AWS Regions. Some Amazon EC2 resources and features, such as enhanced networking and newer instance types, require a virtual private cloud (VPC). Some resources can be shared between EC2-Classic
and a VPC, while some can't. For more information, see *Share and access resources between EC2-Classic and a VPC* (p. 1029). We recommend that you migrate to a VPC to take advantage of VPC-only features.

To migrate from EC2-Classic to a VPC, you must migrate or recreate your EC2-Classic resources in a VPC. You can migrate and recreate your resources in full, or you can perform an incremental migration over time using ClassicLink.

**Contents**

- Options for getting a default VPC (p. 1042)
- Migrate your resources to a VPC (p. 1043)
- Use ClassicLink for an incremental migration (p. 1046)
- Example: Migrate a simple web application (p. 1048)

**Options for getting a default VPC**

A *default VPC* is a VPC that is configured and ready for you to use, and is only available in Regions that are VPC-only. For Regions that support EC2-Classic, you can create a nondefault VPC to set up your resources. However, you might want to use a default VPC if you prefer not to set up a VPC yourself, or if you do not have specific requirements for your VPC configuration. For more information about default VPCs, see Default VPC and Default Subnets in the *Amazon VPC User Guide*.

The following are options for using a default VPC when you have an AWS account that supports EC2-Classic.

**Options**

- Switch to a VPC-only Region (p. 1042)
- Create a new AWS account (p. 1042)
- Convert your existing AWS account to VPC-only (p. 1042)

**Switch to a VPC-only Region**

Use this option if you want to use your existing account to set up your resources in a default VPC and you do not need to use a specific Region. To find a Region that has a default VPC, see Detect supported platforms (p. 1022).

**Create a new AWS account**

New AWS accounts support VPC only. Use this option if you want an account that has a default VPC in every Region.

**Convert your existing AWS account to VPC-only**

Use this option if you want a default VPC in every Region in your existing account. Before you can convert your account, you must delete all of your EC2-Classic resources. You can also migrate some resources to a VPC. For more information, see Migrate your resources to a VPC (p. 1043).

**To convert your EC2-Classic account**

1. Delete or migrate (if applicable) the resources that you have created for use in EC2-Classic. These include the following:

   - Amazon EC2 instances
   - EC2-Classic security groups (excluding the default security group, which you cannot delete yourself)
Migrate from EC2-Classic to a VPC

• EC2-Classic Elastic IP addresses
• Classic Load Balancers
• Amazon RDS resources
• Amazon ElastiCache resources
• Amazon Redshift resources
• AWS Elastic Beanstalk resources
• AWS Data Pipeline resources
• Amazon EMR resources
• AWS OpsWorks resources

2. Go to the Amazon Web Services Support Center at console.aws.amazon.com/support.
3. Choose Create case.
4. Choose Account and billing support.
5. For Type, choose Account. For Category, choose Convert EC2 Classic to VPC.
6. Fill in the other details as required, and choose Submit. We will review your request and contact you to guide you through the next steps.

Migrate your resources to a VPC

You can migrate or move some of your resources to a VPC. Some resources can only be migrated from EC2-Classic to a VPC that's in the same Region and in the same AWS account. If the resource cannot be migrated, you must create a new resource for use in your VPC.

Prerequisites

Before you begin, you must have a VPC. If you don't have a default VPC, you can create a nondefault VPC using one of these methods:

• In the Amazon VPC console, use the VPC wizard to create a new VPC. For more information, see Amazon VPC Console Wizard Configurations. Use this option if you want to set up a VPC quickly, using one of the available configuration options.
• In the Amazon VPC console, set up the components of a VPC according to your requirements. For more information, see VPCs and Subnets. Use this option if you have specific requirements for your VPC, such as a particular number of subnets.

Topics

• Security groups (p. 1043)
• Elastic IP addresses (p. 1044)
• AMIs and instances (p. 1044)
• Amazon RDS DB instances (p. 1046)

Security groups

If you want your instances in your VPC to have the same security group rules as your EC2-Classic instances, you can use the Amazon EC2 console to copy your existing EC2-Classic security group rules to a new VPC security group.

You can only copy security group rules to a new security group in the same AWS account in the same Region. If you are using a different Region or a different AWS account, you must create a new security group and manually add the rules yourself. For more information, see Amazon EC2 security groups for Windows instances (p. 1135).
To copy your security group rules to a new security group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Security Groups.
3. Select the security group that's associated with your EC2-Classic instance, then choose Actions, and select Copy to new.
   
   **Note**
   To identify an EC2-Classic security group, check the VPC ID column. For each EC2-Classic security group, the value in the column is blank or a – symbol.

4. In the Create Security Group dialog box, specify a name and description for your new security group. Select your VPC from the VPC list.
5. The Inbound tab is populated with the rules from your EC2-Classic security group. You can modify the rules as required. In the Outbound tab, a rule that allows all outbound traffic has automatically been created for you. For more information about modifying security group rules, see Amazon EC2 security groups for Windows instances (p. 1135).
   
   **Note**
   If you've defined a rule in your EC2-Classic security group that references another security group, you cannot use the same rule in your VPC security group. Modify the rule to reference a security group in the same VPC.

6. Choose Create.

Elastic IP addresses

You can migrate an Elastic IP address that is allocated for use in EC2-Classic for use with a VPC. You cannot migrate an Elastic IP address to another Region or AWS account. For more information, see Migrate an Elastic IP Address from EC2-Classic (p. 1027).

To identify an Elastic IP address that is allocated for use in EC2-Classic

In the Amazon EC2 console, choose Elastic IPs in the navigation pane. In the Scope column, the value is standard.

Alternatively, use the following describe-addresses command.

```
aws ec2 describe-addresses --filters Name=domain,Values=standard
```

AMIs and instances

An AMI is a template for launching your Amazon EC2 instance. You can create your own AMI based on an existing EC2-Classic instance, then use that AMI to launch instances into your VPC.

**Contents**
- Identify EC2-Classic instances (p. 1044)
- Create an AMI (p. 1045)
- (Optional) Share or copy your AMI (p. 1045)
- (Optional) Store your data on Amazon EBS volumes (p. 1046)
- Launch an instance into your VPC (p. 1046)

**Identify EC2-Classic instances**

If you have instances running in both EC2-Classic and a VPC, you can identify your EC2-Classic instances.

**Amazon EC2 console**
Choose **Instances** in the navigation pane. In the **VPC ID** column, the value for each EC2-Classic instance is blank or a – symbol. If the **VPC ID** column is not present, choose the gear icon and make the column visible.

**AWS CLI**

Use the following `describe-instances` AWS CLI command. The `--query` parameter displays only instances where the value for `VpcId` is null.

```bash
aws ec2 describe-instances --query 'Reservations[*].Instances[?VpcId==`null`]'
```

**Create an AMI**

After you've identified your EC2-Classic instance, you can create an AMI from it.

**To create a Windows AMI**

For more information, see [Creating a custom Windows AMI](#).

**To create a Linux AMI**

The method that you use to create your Linux AMI depends on the root device type of your instance, and the operating system platform on which your instance runs. To find out the root device type of your instance, go to the **Instances** page, select your instance, and look at the information in the **Root device type** field in the **Description** tab. If the value is `ebs`, then your instance is EBS-backed. If the value is `instance-store`, then your instance is instance store-backed. You can also use the `describe-instances` AWS CLI command to find out the root device type.

The following table provides options for you to create your Linux AMI based on the root device type of your instance, and the software platform.

**Important**

Some instance types support both PV and HVM virtualization, while others support only one or the other. If you plan to use your AMI to launch a different instance type than your current instance type, verify that the instance type supports the type of virtualization that your AMI offers. If your AMI supports PV virtualization, and you want to use an instance type that supports HVM virtualization, you might have to reinstall your software on a base HVM AMI. For more information about PV and HVM virtualization, see [Linux AMI virtualization types](#).

<table>
<thead>
<tr>
<th>Instance root device type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBS</td>
<td>Create an EBS-backed AMI from your instance. For more information, see <a href="#">Creating an Amazon EBS-backed Linux AMI</a>.</td>
</tr>
<tr>
<td>Instance store</td>
<td>Create an instance store-backed AMI from your instance using the AMI tools. For more information, see <a href="#">Creating an instance store-backed Linux AMI</a>.</td>
</tr>
<tr>
<td>Instance store</td>
<td>Convert your instance store-backed instance to an EBS-backed instance. For more information, see <a href="#">Converting your instance store-backed AMI to an Amazon EBS-backed AMI</a>.</td>
</tr>
</tbody>
</table>

(Optional) **Share or copy your AMI**

To use your AMI to launch an instance in a new AWS account, you must first share the AMI with your new account. For more information, see [Share an AMI with specific AWS accounts (p. 106)](#).

To use your AMI to launch an instance in a VPC in a different Region, you must first copy the AMI to that Region. For more information, see [Copy an AMI (p. 113)](#).
(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. 1160)
- Create an Amazon EBS volume (p. 1183)
- Attach an Amazon EBS volume to an instance (p. 1186)

To back up the data on your Amazon EBS volume, you can take periodic snapshots of your volume.
For more information, see Create Amazon EBS snapshots (p. 1211). If you need to, you can create
an Amazon EBS volume from your snapshot. For more information, see Create a volume from a
snapshot (p. 1185).

Launch an instance into your VPC

After you've created an AMI, you can use the Amazon EC2 launch wizard to launch an instance into your
VPC. The instance will have the same data and configurations as your existing EC2-Classic instance.

**Note**
You can use this opportunity to **upgrade to a current generation instance type**.

To launch an instance into your VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the dashboard, choose **Launch instance**.
3. On the **Choose an Amazon Machine Image** page, select the **My AMIs** category, and select the AMI
   you created. Alternatively, if you shared an AMI from another account, in the **Ownership** filter list,
   choose **Shared with me**. Select the AMI that you shared from your EC2-Classic account.
4. On the **Choose an Instance Type** page, select the type of instance, and choose **Next: Configure
   Instance Details**.
5. On the **Configure Instance Details** page, select your VPC from the **Network** list. Select the required
   subnet from the **Subnet** list. Configure any other details that you require, then go through the next
   pages of the wizard until you reach the **Configure Security Group** page.
6. Select **Select an existing group**, and select the security group that you created for your VPC. Choose
   **Review and Launch**.
7. Review your instance details, then choose **Launch** to specify a key pair and launch your instance.

For more information about the parameters that you can configure in each step of the wizard, see
Launch an instance using the Launch Instance Wizard (p. 392).

**Amazon RDS DB instances**

You can move your EC2-Classic DB instance to a VPC in the same Region, in the same account. For more
information, see Updating the VPC for a DB Instance in the Amazon RDS User Guide.

**Use ClassicLink for an incremental migration**

The ClassicLink feature makes it easier to manage an incremental migration to a VPC. ClassicLink enables
you to link an EC2-Classic instance to a VPC in your account in the same Region, allowing your new VPC
resources to communicate with the EC2-Classic instance using private IPv4 addresses. You can then migrate functionality one component at a time until your application is running fully in your VPC.

Use this option if you cannot afford downtime during the migration, for example, if you have a multi-tier application with processes that cannot be interrupted.

For more information about ClassicLink, see ClassicLink (p. 1030).

**Tasks**
- Step 1: Prepare your migration sequence (p. 1047)
- Step 2: Enable your VPC for ClassicLink (p. 1047)
- Step 3: Link your EC2-Classic instances to your VPC (p. 1047)
- Step 4: Complete the VPC migration (p. 1048)

**Step 1: Prepare your migration sequence**

To use ClassicLink effectively, you must first identify the components of your application that must be migrated to the VPC, and then confirm the order in which to migrate that functionality.

For example, you have an application that relies on a presentation web server, a backend database server, and authentication logic for transactions. You may decide to start the migration process with the authentication logic, then the database server, and finally, the web server.

Then, you can start migrating or recreating your resources. For more information, see Migrate your resources to a VPC (p. 1043).

**Step 2: Enable your VPC for ClassicLink**

After you've configured your new VPC instances and made the functionality of your application available in the VPC, you can use ClassicLink to enable private IP communication between your new VPC instances and your EC2-Classic instances. First, you must enable your VPC for ClassicLink.

**To enable a VPC for ClassicLink**

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose Your VPCs.
3. Select a VPC.
4. Choose Actions, Enable ClassicLink.
5. When prompted for confirmation, choose Enable ClassicLink.

**Step 3: Link your EC2-Classic instances to your VPC**

After you've enabled ClassicLink in your VPC, you can link your EC2-Classic instances to the VPC. The instance must be in the running state.

**To link an instance to a VPC**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select one or more running EC2-Classic instances.
4. Choose Actions, ClassicLink, Link to VPC.
5. Choose a VPC. The console displays only VPCs that are enabled for ClassicLink.
6. Select one or more security groups to associate with your instances. The console displays security groups only for VPCs enabled for ClassicLink.

7. Choose Link.

**Step 4: Complete the VPC migration**

Depending on the size of your application and the functionality that must be migrated, repeat the preceding steps until you've moved all of the components of your application from EC2-Classic into your VPC.

After you've enabled internal communication between the EC2-Classic and VPC instances, you must update your application to point to your migrated service in your VPC, instead of your service in the EC2-Classic platform. The exact steps for this depend on your application’s design. Generally, this includes updating your destination IP addresses to point to the IP addresses of your VPC instances instead of your EC2-Classic instances.

After you've completed this step and you've tested that the application is functioning from your VPC, you can terminate your EC2-Classic instances, and disable ClassicLink for your VPC. You can also clean up any EC2-Classic resources that you no longer need to avoid incurring charges for them. For example, you can release Elastic IP addresses and delete the volumes that were associated with your EC2-Classic instances.

**Example: Migrate a simple web application**

In this example, you use AWS to host your gardening website. To manage your website, you have three running instances in EC2-Classic. Instances A and B host your public-facing web application, and you use Elastic Load Balancing to load balance the traffic between these instances. You've assigned Elastic IP addresses to instances A and B so that you have static IP addresses for configuration and administration tasks on those instances. Instance C holds your MySQL database for your website. You've registered the domain name www.garden.example.com, and you've used Route 53 to create a hosted zone with an alias record set that's associated with the DNS name of your load balancer.

The first part of migrating to a VPC is deciding what kind of VPC architecture suits your needs. In this case, you've decided on the following: one public subnet for your web servers, and one private subnet for your database server. As your website grows, you can add more web servers and database servers to your subnets. By default, instances in the private subnet cannot access the internet; however, you can enable internet access through a Network Address Translation (NAT) device in the public subnet. You might want to set up a NAT device to support periodic updates and patches from the internet for your database server. You'll migrate your Elastic IP addresses to a VPC, and create a load balancer in your public subnet to load balance the traffic between your web servers.
To migrate your web application to a VPC, you can follow these steps:

- **Create a VPC**: In this case, you can use the VPC wizard in the Amazon VPC console to create your VPC and subnets. The second wizard configuration creates a VPC with one private and one public subnet, and launches and configures a NAT device in your public subnet for you. For more information, see VPC with public and private subnets (NAT) in the Amazon VPC User Guide.

- **Configure your security groups**: In your EC2-Classic environment, you have one security group for your web servers, and another security group for your database server. You can use the Amazon EC2 console to copy the rules from each security group into new security groups for your VPC. For more information, see Security groups (p. 1043).

  **Tip**
  Create the security groups that are referenced by other security groups first.

- **Create AMIs and launch new instances**: Create an AMI from one of your web servers, and a second AMI from your database server. Then, launch replacement web servers into your public subnet, and launch your replacement database server into your private subnet. For more information, see Create an AMI (p. 1045).

- **Configure your NAT device**: If you are using a NAT instance, you must create a security group for it that allows HTTP and HTTPS traffic from your private subnet. For more information, see NAT instances. If you are using a NAT gateway, traffic from your private subnet is automatically allowed.

- **Configure your database**: When you created an AMI from your database server in EC2-Classic, all of the configuration information that was stored in that instance was copied to the AMI. You might have to connect to your new database server and update the configuration details. For example, if you configured your database to grant full read, write, and modification permissions to your web servers in EC2-Classic, you need to update the configuration files to grant the same permissions to your new VPC web servers instead.

- **Configure your web servers**: Your web servers will have the same configuration settings as your instances in EC2-Classic. For example, if you configured your web servers to use the database in EC2-Classic, update your web servers’ configuration settings to point to your new database instance.
By default, instances launched into a nondefault subnet are not assigned a public IP address, unless you specify otherwise at launch. Your new database server might not have a public IP address. In this case, you can update your web servers' configuration file to use your new database server's private DNS name. Instances in the same VPC can communicate with each other via private IP address.

- **Migrate your Elastic IP addresses**: Disassociate your Elastic IP addresses from your web servers in EC2-Classic, and then migrate them to a VPC. After you've migrated them, you can associate them with your new web servers in your VPC. For more information, see [Migrate an Elastic IP Address from EC2-Classic](p. 1027).

- **Create a new load balancer**: To continue using Elastic Load Balancing to load balance the traffic to your instances, make sure you understand the various ways to configure your load balancer in VPC. For more information, see the [Elastic Load Balancing User Guide](#).

- **Update your DNS records**: After you've set up your load balancer in your public subnet, verify that your `www.garden.example.com` domain points to your new load balancer. To do this, update your DNS records and your alias record set in Route 53. For more information about using Route 53, see [Getting Started with Route 53](#).

- **Shut down your EC2-Classic resources**: After you've verified that your web application is working from within the VPC architecture, you can shut down your EC2-Classic resources to stop incurring charges for them.
Security in Amazon EC2

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Amazon EC2, see AWS Services in Scope by Compliance Program.

- **Security in the cloud** – Your responsibility includes the following areas:
  - Controlling network access to your instances, for example, through configuring your VPC and security groups. For more information, see Controlling network traffic (p. 1052).
  - Managing the credentials used to connect to your instances.
  - Managing the guest operating system and software deployed to the guest operating system, including updates and security patches. For more information, see Update management in Amazon EC2 (p. 1155).
  - Configuring the IAM roles that are attached to the instance and the permissions associated with those roles. For more information, see IAM roles for Amazon EC2 (p. 1114).

This documentation helps you understand how to apply the shared responsibility model when using Amazon EC2. It shows you how to configure Amazon EC2 to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon EC2 resources.

For security best practices for Amazon EC2 running Windows Server, see Security and Network under Best practices for Windows on Amazon EC2 (p. 18).

Contents

- Infrastructure security in Amazon EC2 (p. 1051)
- Amazon EC2 and interface VPC endpoints (p. 1054)
- Resilience in Amazon EC2 (p. 1055)
- Data protection in Amazon EC2 (p. 1056)
- Identity and access management for Amazon EC2 (p. 1058)
- Amazon EC2 key pairs and Windows instances (p. 1127)
- Amazon EC2 security groups for Windows instances (p. 1135)
- Configuration management in Amazon EC2 (p. 1155)
- Update management in Amazon EC2 (p. 1155)
- Change management in Amazon EC2 (p. 1156)
- Compliance validation for Amazon EC2 (p. 1156)
- Audit and accountability in Amazon EC2 (p. 1157)

Infrastructure security in Amazon EC2

As a managed service, Amazon EC2 is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.
You use AWS published API calls to access Amazon EC2 through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Network isolation

A virtual private cloud (VPC) is a virtual network in your own logically isolated area in the AWS Cloud. Use separate VPCs to isolate infrastructure by workload or organizational entity.

A subnet is a range of IP addresses in a VPC. When you launch an instance, you launch it into a subnet in your VPC. Use subnets to isolate the tiers of your application (for example, web, application, and database) within a single VPC. Use private subnets for your instances if they should not be accessed directly from the internet.

To call the Amazon EC2 API from your VPC without sending traffic over the public internet, use AWS PrivateLink.

Isolation on physical hosts

Different EC2 instances on the same physical host are isolated from each other as though they are on separate physical hosts. The hypervisor isolates CPU and memory, and the instances are provided virtualized disks instead of access to the raw disk devices.

When you stop or terminate an instance, the memory allocated to it is scrubbed (set to zero) by the hypervisor before it is allocated to a new instance, and every block of storage is reset. This ensures that your data is not unintentionally exposed to another instance.

Network MAC addresses are dynamically assigned to instances by the AWS network infrastructure. IP addresses are either dynamically assigned to instances by the AWS network infrastructure, or assigned by an EC2 administrator through authenticated API requests. The AWS network allows instances to send traffic only from the MAC and IP addresses assigned to them. Otherwise, the traffic is dropped.

By default, an instance cannot receive traffic that is not specifically addressed to it. If you need to run network address translation (NAT), routing, or firewall services on your instance, you can disable source/destination checking for the network interface.

Controlling network traffic

Consider the following options for controlling network traffic to your EC2 instances:

- Restrict access to your instances using security groups (p. 1135). Configure Amazon EC2 instance security groups to permit the minimum required network traffic for the Amazon EC2 instance and to allow access only from defined, expected, and approved locations. For example, if an Amazon EC2 instance is an IIS web server, configure its security groups to permit only inbound HTTP/HTTPS, Windows management traffic, and minimal outbound connections.

- Leverage security groups as the primary mechanism for controlling network access to Amazon EC2 instances. When necessary, use network ACLs sparingly to provide stateless, coarse-grain network control. Security groups are more versatile than network ACLs due to their ability to perform stateful packet filtering and create rules that reference other security groups. However, network ACLs can be effective as a secondary control for denying a specific subset of traffic or providing high-level subnet guard rails. Also, because network ACLs apply to an entire subnet, they can be used as defense-in-depth in case an instance is ever launched unintentionally without a correct security group.
• Centrally manage Windows Firewall settings with Group Policy Objects (GPO) to further enhance network controls. Customers often use the Windows Firewall for further visibility into network traffic and to complement security group filters, creating advanced rules to block specific applications from accessing the network or to filter traffic from a subset IP addresses. For example, the Windows Firewall can limit access to the EC2 metadata service IP address to specific users or applications. Alternatively, a public-facing service might use security groups to restrict traffic to specific ports and the Windows Firewall to maintain a list of explicitly blocked IP addresses.

• When managing Windows instances, limit access to a few well-defined centralized management servers or bastion hosts to reduce the environment’s attack surface. Also, use secure administration protocols like RDP encapsulation over SSL/TLS. The Remote Desktop Gateway Quick Start provides best practices for deploying remote desktop gateway, including configuring RDP to use SSL/TLS.

• Use Active Directory or AWS Directory Service to tightly and centrally control and monitor interactive user and group access to Windows instances, and avoid local user permissions. Also avoid using Domain Administrators and instead create more granular, application-specific role-based accounts. Just Enough Administration (JEA) allows changes to Windows instances to be managed without interactive or administrator access. In addition, JEA enables organizations to lock down administrative access to the subset of Windows PowerShell commands required for instance administration. For additional information, see the section on "Managing OS-level Access to Amazon EC2" in the AWS Security Best Practices whitepaper.

• Systems Administrators should use Windows accounts with limited access to perform daily activities, and only elevate access when necessary to perform specific configuration changes. Additionally, only access Windows instances directly when absolutely necessary. Instead, leverage central configuration management systems such as EC2 Run Command, Systems Center Configuration Manager (SCCM), Windows PowerShell DSC, or Amazon EC2 Systems Manager (SSM) to push changes to Windows servers.

• Configure Amazon VPC subnet route tables with the minimal required network routes. For example, place only Amazon EC2 instances that require direct Internet access into subnets with routes to an Internet Gateway, and place only Amazon EC2 instances that need direct access to internal networks into subnets with routes to a virtual private gateway.

• Consider using additional security groups or ENIs to control and audit Amazon EC2 instance management traffic separately from regular application traffic. This approach allows customers to implement special IAM policies for change control, making it easier to audit changes to security group rules or automated rule-verification scripts. Multiple ENIs also provide additional options for controlling network traffic including the ability to create host-based routing policies or leverage different VPC subnet routing rules based on an ENI's assigned subnet.

• Use AWS Virtual Private Network or AWS Direct Connect to establish private connections from your remote networks to your VPCs. For more information, see Network-to-Amazon VPC Connectivity Options.

• Use VPC Flow Logs to monitor the traffic that reaches your instances.

• Use AWS Security Hub to check for unintended network accessibility from your instances.

• Use AWS Systems Manager Session Manager to access your instances remotely instead of opening inbound RDP ports.

• Use AWS Systems Manager Run Command to automate common administrative tasks instead of opening inbound RDP ports.

• Many of the Windows OS roles and Microsoft business applications also provide enhanced functionality such as IP Address Range restrictions within IIS, TCP/IP filtering policies in Microsoft SQL Server, and connection filter policies in Microsoft Exchange. Network restriction functionality within the application layer can provide additional layers of defense for critical business application servers.

In addition to restricting network access to each Amazon EC2 instance, Amazon VPC supports implementing additional network security controls like in-line gateways, proxy servers, and various network monitoring options.

For more information, see the AWS Security Best Practices whitepaper.
Amazon EC2 and interface VPC endpoints

You can improve the security posture of your VPC by configuring Amazon EC2 to use an interface VPC endpoint. Interface endpoints are powered by AWS PrivateLink, a technology that enables you to privately access Amazon EC2 APIs by restricting all network traffic between your VPC and Amazon EC2 to the Amazon network. With interface endpoints, you also don't need an internet gateway, a NAT device, or a virtual private gateway.

You are not required to configure AWS PrivateLink, but it's recommended. For more information about AWS PrivateLink and VPC endpoints, see Interface VPC Endpoints (AWS PrivateLink).

Topics
- Create an interface VPC endpoint (p. 1054)
- Create an interface VPC endpoint policy (p. 1054)

Create an interface VPC endpoint

Create an endpoint for Amazon EC2 using the following service name:

- com.amazonaws.region.ec2 — Creates an endpoint for the Amazon EC2 API actions.

For more information, see Creating an Interface Endpoint in the Amazon VPC User Guide.

Create an interface VPC endpoint policy

You can attach a policy to your VPC endpoint to control access to the Amazon EC2 API. The policy specifies:

- The principal that can perform actions.
- The actions that can be performed.
- The resource on which the actions can be performed.

**Important**

When a non-default policy is applied to an interface VPC endpoint for Amazon EC2, certain failed API requests, such as those failing from RequestLimitExceeded, might not be logged to AWS CloudTrail or Amazon CloudWatch.

For more information, see Controlling Access to Services with VPC Endpoints in the Amazon VPC User Guide.

The following example shows a VPC endpoint policy that denies permission to create unencrypted volumes or to launch instances with unencrypted volumes. The example policy also grants permission to perform all other Amazon EC2 actions.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "ec2:*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```
Resilience in Amazon EC2

The AWS global infrastructure is built around AWS Regions and Availability Zones. Regions provide multiple physically separated and isolated Availability Zones, which are connected through low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

If you need to replicate your data or applications over greater geographic distances, use AWS Local Zones. An AWS Local Zone is an extension of an AWS Region in geographic proximity to your users. Local Zones have their own connections to the internet and support AWS Direct Connect. Like all AWS Regions, AWS Local Zones are completely isolated from other AWS Zones.

If you need to replicate your data or applications in an AWS Local Zone, AWS recommends that you use one of the following zones as the failover zone:

- Another Local Zone
- An Availability Zone in the Region that is not the parent zone. You can use the describe-availability-zones command to view the parent zone.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

In addition to the AWS global infrastructure, Amazon EC2 offers the following features to support your data resiliency:

- Copying AMIs across Regions
- Copying EBS snapshots across Regions
- Automating EBS-backed AMIs using Amazon Data Lifecycle Manager
• Automating EBS snapshots using Amazon Data Lifecycle Manager
• Maintaining the health and availability of your fleet using Amazon EC2 Auto Scaling
• Distributing incoming traffic across multiple instances in a single Availability Zone or multiple Availability Zones using Elastic Load Balancing

Data protection in Amazon EC2

The AWS shared responsibility model applies to data protection in Amazon Elastic Compute Cloud. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the AWS Security Blog.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

• Use multi-factor authentication (MFA) with each account.
• Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
• Set up API and user activity logging with AWS CloudTrail.
• Use AWS encryption solutions, along with all default security controls within AWS services.
• Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
• If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a Name field. This includes when you work with Amazon EC2 or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Encryption at rest

Amazon EBS encryption is an encryption solution for your EBS volumes and snapshots. It uses AWS KMS keys. For more information, see Amazon EBS encryption (p. 1327).

Customers can also use Microsoft EFS and NTFS permissions for folder- and file-level encryption.

The data on NVMe instance store volumes is encrypted using an XTS-AES-256 cipher implemented on a hardware module on the instance. The encryption keys are generated using the hardware module and are unique to each NVMe instance storage device. All encryption keys are destroyed when the instance is stopped or terminated and cannot be recovered. You cannot disable this encryption and you cannot provide your own encryption key.

The data on HDD instance store volumes on H1, D3, and D3en instances is encrypted using XTS-AES-256 and one-time keys.
Encryption in transit

Encryption at the physical layer

All data flowing across AWS Regions over the AWS global network is automatically encrypted at the physical layer before it leaves AWS secured facilities. All traffic between AZs is encrypted. Additional layers of encryption, including those listed in this section, may provide additional protections.

Encryption provided by Amazon VPC and Transit Gateway cross-Region peering

All cross-Region traffic that uses Amazon VPC and Transit Gateway peering is automatically bulk-encrypted when it exits a Region. An additional layer of encryption is automatically provided at the physical layer for all cross-Region traffic, as previously noted in this section.

Encryption between instances

AWS provides secure and private connectivity between EC2 instances of all types. In addition, some instance types use the offload capabilities of the underlying Nitro System hardware to automatically encrypt in-transit traffic between instances, using AEAD algorithms with 256-bit encryption. There is no impact on network performance. To support this additional in-transit traffic encryption between instances, the following requirements must be met:

- The instances use the following instance types:
  - General purpose: M5dn | M5n | M5zn | M6i
  - Compute optimized: C5a | C5ad | C5n
  - Memory optimized: R5dn | R5n | high memory (u-*), virtualized only
  - Storage optimized: D3 | D3en | I3en
  - Accelerated computing: G4ad | G4dn | P3dn
- The instances are in the same Region.
- The instances are in the same VPC or peered VPCs, and the traffic does not pass through a virtual network device or service, such as a load balancer or a transit gateway.

An additional layer of encryption is automatically provided at the physical layer for all traffic before it leaves AWS secured facilities, as previously noted in this section.

To view the instance types that encrypt in-transit traffic between instances using the AWS CLI

Use the following `describe-instance-types` command.

```
aws ec2 describe-instance-types \
  --filters Name=network-info.encryption-in-transit-supported,Values=true \
  --query "InstanceTypes[*].[InstanceType]" --output text
```

Encryption to and from AWS Outposts

An Outpost creates special network connections called service links to its AWS home Region and, optionally, private connectivity to a VPC subnet that you specify. All traffic over those connection is fully encrypted. For more information, see Connectivity through service links and Encryption in transit in the AWS Outposts User Guide.

Remote access encryption

RDP provides a secure communications channel for remote access to your Windows instances, whether directly or through EC2 Instance Connect. Remote access to your instances using AWS Systems Manager Session Manager or the Run Command is encrypted using TLS 1.2, and requests to create a connection are signed using SigV4 and authenticated and authorized by AWS Identity and Access Management.
It is your responsibility to use an encryption protocol, such as Transport Layer Security (TLS), to encrypt sensitive data in transit between clients and your Amazon EC2 instances.

Make sure to allow only encrypted connections between EC2 instances and the AWS API endpoints or other sensitive remote network services. You can enforce this through an outbound security group or Windows Firewall rules.

Identity and access management for Amazon EC2

Your security credentials identify you to services in AWS and grant you unlimited use of your AWS resources, such as your Amazon EC2 resources. You can use features of Amazon EC2 and AWS Identity and Access Management (IAM) to allow other users, services, and applications to use your Amazon EC2 resources without sharing your security credentials. You can use IAM to control how other users use resources in your AWS account, and you can use security groups to control access to your Amazon EC2 instances. You can choose to allow full use or limited use of your Amazon EC2 resources.

Contents

- Network access to your instance (p. 1058)
- Amazon EC2 permission attributes (p. 1058)
- IAM and Amazon EC2 (p. 1058)
- IAM policies for Amazon EC2 (p. 1060)
- AWS managed policies for Amazon Elastic Compute Cloud (p. 1113)
- IAM roles for Amazon EC2 (p. 1114)
- Authorize inbound traffic for your Windows instances (p. 1123)

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 Authorize inbound traffic for your Windows instances (p. 1123).

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 Make an AMI public (p. 104).

Each Amazon EBS snapshot has a createVolumePermission attribute that controls which AWS accounts can use the snapshot. For more information, see Share an Amazon EBS snapshot (p. 1234).

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?

Create an IAM group and users

To create an IAM group

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose Groups and then choose Create New Group.
3. For Group Name, enter a name for your group, and then choose Next Step.
4. On the Attach Policy page, select an AWS managed policy and then choose Next Step. For example, for Amazon EC2, one of the following AWS managed policies might meet your needs:
   • PowerUserAccess
   • ReadOnlyAccess
   • AmazonEC2FullAccess
   • AmazonEC2ReadOnlyAccess
5. Choose Create Group.

Your new group is listed under Group Name.

To create an IAM user, add the user to your group, and create a password for the user

1. In the navigation pane, choose Users, Add user.
2. For User name, enter a user name.
3. For Access type, select both Programmatic access and AWS Management Console access.
4. For Console password, choose one of the following:
   • Autogenerated password. Each user gets a randomly generated password that meets the current password policy in effect (if any). You can view or download the passwords when you get to the Final page.
   • Custom password. Each user is assigned the password that you enter in the box.
5. Choose Next: Permissions.
6. On the Set permissions page, choose Add user to group. Select the check box next to the group that you created earlier and choose Next: Review.
7. Choose Create user.
8. To view the users' access keys (access key IDs and secret access keys), choose Show next to each password and secret access key to see. To save the access keys, choose Download .csv and then save the file to a safe location.

**Important**
You cannot retrieve the secret access key after you complete this step; if you misplace it you must create a new one.

9. Choose Close.

10. Give each user his or her credentials (access keys and password); this enables them to use services based on the permissions you specified for the IAM group.

## Related topics

For more information about IAM, see the following:

- IAM policies for Amazon EC2 (p. 1060)
- IAM roles for Amazon EC2 (p. 1114)
- 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.

<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understand the basic structure of a policy</td>
<td>Policy syntax (p. 1061)</td>
</tr>
<tr>
<td>Define actions in your policy</td>
<td>Actions for Amazon EC2 (p. 1062)</td>
</tr>
<tr>
<td>Define specific resources in your policy</td>
<td>Amazon Resource Names (ARNs) for Amazon EC2 (p. 1062)</td>
</tr>
<tr>
<td>Apply conditions to the use of the resources</td>
<td>Condition keys for Amazon EC2 (p. 1063)</td>
</tr>
<tr>
<td>Work with the available resource-level permissions for Amazon EC2</td>
<td>Actions, resources, and condition keys for Amazon EC2</td>
</tr>
</tbody>
</table>
Policy structure

The following topics explain the structure of an IAM policy.

Contents

- Policy syntax (p. 1061)
- Actions for Amazon EC2 (p. 1062)
- Supported resource-level permissions for Amazon EC2 API actions (p. 1062)
- Amazon Resource Names (ARNs) for Amazon EC2 (p. 1062)
- Condition keys for Amazon EC2 (p. 1063)
- Check that users have the required permissions (p. 1064)

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. 1062).
- **Resource**: The resource that's affected by the action. Some Amazon EC2 API actions allow you to include specific resources in your policy that can be created or modified by the action. You specify a resource using an Amazon Resource Name (ARN) or using the wildcard (*) to indicate that the...
statement applies to all resources. For more information, see Supported resource-level permissions for Amazon EC2 API actions (p. 1062).

- **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. 1063).

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. 1067).

**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 defined by Amazon EC2 in the Service Authorization Reference.

**Supported resource-level permissions for Amazon EC2 API actions**

**Resource-level permissions** refers to the ability to specify which resources users are allowed to perform actions on. Amazon EC2 has partial support for resource-level permissions. This means that for certain Amazon EC2 actions, you can control when users are allowed to use those actions based on conditions that have to be fulfilled, or specific resources that users are allowed to use. For example, you can grant users permissions to launch instances, but only of a specific type, and only using a specific AMI.

To specify a resource in an IAM policy statement, use its Amazon Resource Name (ARN). For more information about specifying the ARN value, see Amazon Resource Names (ARNs) for Amazon EC2 (p. 1062). If an API action does not support individual ARNs, you must use a wildcard (*) to specify that all resources can be affected by the action.

To see tables that identify which Amazon EC2 API actions support resource-level permissions, and the ARNs and condition keys that you can use in a policy, see Actions, resources, and condition keys for Amazon EC2.

Keep in mind that you can apply tag-based resource-level permissions in the IAM policies you use for Amazon EC2 API actions. This gives you better control over which resources a user can create, modify, or use. For more information, see Grant permission to tag resources during creation (p. 1065).

**Amazon Resource Names (ARNs) for Amazon EC2**

Each IAM policy statement applies to the resources that you specify using their ARNs.

An ARN has the following general syntax:
The service (for example, ec2).

The Region for the resource (for example, us-east-1).

The AWS account ID, with no hyphens (for example, 123456789012).

The type of resource (for example, instance).

A path that identifies the resource. You can use the * wildcard in your paths.

For example, you can indicate a specific instance (i-1234567890abcdef0) in your statement using its ARN as follows.

"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"

You can specify all instances that belong to a specific account by using the * wildcard as follows.

"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"

You can also specify all Amazon EC2 resources that belong to a specific account by using the * wildcard as follows.

"Resource": "arn:aws:ec2:us-east-1:123456789012:*"

To specify all resources, or if a specific API action does not support ARNs, use the * wildcard in the Resource element as follows.

"Resource": "*"

Many Amazon EC2 API actions involve multiple resources. For example, AttachVolume attaches an Amazon EBS volume to an instance, so an IAM user must have permissions to use the volume and the instance. To specify multiple resources in a single statement, separate their ARNs with commas, as follows.

"Resource": ["arn1", "arn2"]

For a list of ARNs for Amazon EC2 resources, see Resource types defined by Amazon EC2.

**Condition keys for Amazon EC2**

In a policy statement, you can optionally specify conditions that control when it is in effect. Each condition contains one or more key-value pairs. Condition keys are not case-sensitive. We've defined AWS-wide condition keys, plus additional service-specific condition keys.

For a list of service-specific condition keys for Amazon EC2, see Condition keys for Amazon EC2. Amazon EC2 also implements the AWS-wide condition keys. For more information, see Information available in all requests in the IAM User Guide.
To use a condition key in your IAM policy, use the `Condition` statement. For example, the following policy grants users permission to add and remove inbound and outbound rules for any security group. It uses the `ec2:Vpc` condition key to specify that these actions can only be performed on security groups in a specific VPC.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress"
      ],
      "Condition": {
        "StringEquals": {
        }
      }
    }
  ]
}
```

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 IAM policy elements: Variables and tags in the IAM User Guide.

**Important**

Many condition keys are specific to a resource, and some API actions use multiple resources. If you write a policy with a condition key, use the `Resource` element of the statement to specify the resource to which the condition key applies. If not, the policy may prevent users from performing the action at all, because the condition check fails for the resources to which the condition key does not apply. If you do not want to specify a resource, or if you've written the `Action` element of your policy to include multiple API actions, then you must use the `...IfExists` condition type to ensure that the condition key is ignored for resources that do not use it. For more information, see `...IfExists` Conditions in the IAM User Guide.

All Amazon EC2 actions support the `aws:RequestedRegion` and `ec2:Region` condition keys. For more information, see Example: Restrict access to a specific Region (p. 1068).

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 Amazon EC2: Attach or detach volumes to an EC2 instance and Example: Allow a specific instance to view resources in other AWS services (p. 1100). The `ec2:SourceInstanceARN` key cannot be used as a variable to populate the ARN for the `Resource` element in a statement.

For example policy statements for Amazon EC2, see Example policies for working with the AWS CLI or an AWS SDK (p. 1067).

**Check 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*.

### Grant permission to tag resources during creation

Some resource-creating Amazon EC2 API actions enable you to specify tags when you create the resource. You can use resource tags to implement attribute-based control (ABAC). For more information, see [Tag your resources](#) and [Control access to EC2 resources using resource tags](#).

To enable users to tag resources on creation, they must have permissions to use the action that creates the resource, such as `ec2:RunInstances` or `ec2:CreateVolume`. If tags are specified in the resource-creating action, Amazon performs additional authorization on the `ec2:CreateTags` action to verify if users have permissions to create tags. Therefore, users must also have explicit permissions to use the `ec2:CreateTags` action.

In the IAM policy definition for the `ec2:CreateTags` action, use the `Condition` element with the `ec2:CreateAction` condition key to give tagging permissions to the action that creates the resource.

The following example demonstrates a policy that allows users to launch instances and apply any tags to instances and volumes during launch. Users are not permitted to tag any existing resources (they cannot call the `ec2:CreateTags` action directly).

```json
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account:*/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction" : "RunInstances"
        }
      }
    }
  ]
}
```
Similarly, the following policy allows users to create volumes and apply any tags to the volumes during volume creation. Users are not permitted to tag any existing resources (they cannot call the `ec2:CreateTags` action directly).

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateVolume"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account:*/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateVolume"
        }
      }
    }
  ]
}
```

The `ec2:CreateTags` action is only evaluated if tags are applied during the resource-creating action. Therefore, a user that has permissions to create a resource (assuming there are no tagging conditions) does not require permissions to use the `ec2:CreateTags` action if no tags are specified in the request. However, if the user attempts to create a resource with tags, the request fails if the user does not have permissions to use the `ec2:CreateTags` action.

The `ec2:CreateTags` action is also evaluated if tags are provided in a launch template. For an example policy, see Tags in a launch template (p. 1088).

**Control access to specific tags**

You can use additional conditions in the `Condition` element of your IAM policies to control the tag keys and values that can be applied to resources.

The following condition keys can be used with the examples in the preceding section:

- `aws:RequestTag`: To indicate that a particular tag key or tag key and value must be present in a request. Other tags can also be specified in the request.
  - Use with the `StringEquals` condition operator to enforce a specific tag key and value combination, for example, to enforce the tag `cost-center=cc123`:

  ```
  "StringEquals": { "aws:RequestTag/cost-center": "cc123" }
  ```

  - Use with the `StringLike` condition operator to enforce a specific tag key in the request; for example, to enforce the tag key `purpose`:

  ```
  "StringLike": { "aws:RequestTag/purpose": "*" }
  ```

- `aws:TagKeys`: To enforce the tag keys that are used in the request.
  - Use with the `ForAllValues` modifier to enforce specific tag keys if they are provided in the request (if tags are specified in the request, only specific tag keys are allowed; no other tags are allowed). For example, the tag keys `environment` or `cost-center` are allowed:
ForAllValues:StringEquals": { "aws:TagKeys": ["environment","cost-center"] }

- Use with the ForAnyValue modifier to enforce the presence of at least one of the specified tag keys in the request. For example, at least one of the tag keys environment or webserver must be present in the request:

"ForAnyValue:StringEquals": { "aws:TagKeys": ["environment","webserver"] }

These condition keys can be applied to resource-creating actions that support tagging, as well as the `ec2:CreateTags` and `ec2:DeleteTags` actions. To learn whether an Amazon EC2 API action supports tagging, see Actions, resources, and condition keys for Amazon EC2.

To force users to specify tags when they create a resource, you must use the `aws:RequestTag` condition key or the `aws:TagKeys` condition key with the ForAnyValue modifier on the resource-creating action. The `ec2:CreateTags` action is not evaluated if a user does not specify tags for the resource-creating action.

For conditions, the condition key is not case-sensitive and the condition value is case-sensitive. Therefore, to enforce the case-sensitivity of a tag key, use the `aws:TagKeys` condition key, where the tag key is specified as a value in the condition.

For example IAM policies, see Example policies for working with the AWS CLI or an AWS SDK (p. 1067). For more information about multi-value conditions, see Creating a Condition That Tests Multiple Key Values in the IAM User Guide.

Control access to EC2 resources using resource tags

When you create an IAM policy that grants IAM users permission to use EC2 resources, you can include tag information in the Condition element of the policy to control access based on tags. This is known as attribute-based access control (ABAC). ABAC provides better control over which resources a user can modify, use, or delete. For more information, see What is ABAC for AWS?

For example, you can create a policy that allows users to terminate an instance, but denies the action if the instance has the tag `environment=production`. To do this, you use the `ec2:ResourceTag` condition key to allow or deny access to the resource based on the tags that are attached to the resource.

"StringEquals": { "ec2:ResourceTag/environment": "production" }

To learn whether an Amazon EC2 API action supports controlling access using the `ec2:ResourceTag` condition key, see Actions, resources, and condition keys for Amazon EC2. Note that the Describe actions do not support resource-level permissions, so you must specify them in a separate statement without conditions.

For example IAM policies, see Example policies for working with the AWS CLI or an AWS SDK (p. 1067).

If you allow or deny users access to resources based on tags, you must consider explicitly denying users the ability to add those tags to or remove them from the same resources. Otherwise, it's possible for a user to circumvent your restrictions and gain access to a resource by modifying its tags.

Example policies for working with the AWS CLI or an AWS SDK

The following examples show policy statements that you could use to control the permissions that IAM users have to Amazon EC2. These policies are designed for requests that are made with the AWS CLI or an AWS SDK. For example policies for working in the Amazon EC2 console, see Example policies for working in the Amazon EC2 console (p. 1104). For examples of IAM policies specific to Amazon VPC, see Identity and Access Management for Amazon VPC.
Examples

- Example: Read-only access (p. 1068)
- Example: Restrict access to a specific Region (p. 1068)
- Work with instances (p. 1069)
- Work with volumes (p. 1070)
- Work with snapshots (p. 1073)
- Launch instances (RunInstances) (p. 1080)
- Work with Spot Instances (p. 1092)
- Example: Work with Reserved Instances (p. 1096)
- Example: Tag resources (p. 1097)
- Example: Work with IAM roles (p. 1099)
- Example: Work with route tables (p. 1100)
- Example: Allow a specific instance to view resources in other AWS services (p. 1100)
- Example: Work with launch templates (p. 1101)
- Work with instance metadata (p. 1102)

Example: Read-only access

The following policy grants users permissions to use all Amazon EC2 API actions whose names begin with `Describe`. The `Resource` element uses a wildcard to indicate that users can specify all resources with these API actions. The * wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see Actions, resources, and condition keys for Amazon EC2.

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: Restrict access to a specific Region

The following policy denies users permission to use all Amazon EC2 API actions unless the Region is Europe (Frankfurt). It uses the global condition key `aws:RequestedRegion`, which is supported by all Amazon EC2 API actions.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "ec2:*",
      "Resource": "***",
      "Condition": {
        "StringNotEquals": {
          "aws:RequestedRegion": "eu-central-1"
        }
      }
    }
  ]
}
```
Alternatively, you can use the condition key `ec2:Region`, which is specific to Amazon EC2 and is supported by all Amazon EC2 API actions.

```json
{ "Version": "2012-10-17",
 "Statement": [
 { "Effect": "Deny",
   "Action": "ec2:*",
   "Resource": "*",
   "Condition": {
     "StringNotEquals": {
       "ec2:Region": "eu-central-1"
     }
   }
 ]
}
```

## Work with instances

### Examples

- Example: Describe, launch, stop, start, and terminate all instances (p. 1069)
- Example: Describe all instances, and stop, start, and terminate only particular instances (p. 1070)

#### Example: Describe, launch, stop, start, and terminate all instances

The following policy grants users permissions to use the API actions specified in the `Action` element. The `Resource` element uses a `*` wildcard to indicate that users can specify all resources with these API actions. The `*` wildcard is also necessary in cases where the API action does not support resource-level permissions. For more information about which ARNs you can use with which Amazon EC2 API actions, see Actions, resources, and condition keys for Amazon EC2.

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.

```json
{ "Version": "2012-10-17",
 "Statement": [
 { "Effect": "Allow",
   "Resource": "*"
 }
]
```
Example: Describe all instances, and stop, start, and terminate only particular instances

The following policy allows users to describe all instances, to start and stop only instances i-1234567890abcdef0 and i-0598c7d356eba48d7, and to terminate only instances in the US East (N. Virginia) Region (us-east-1) with the resource tag "purpose=test".

The first statement uses a * wildcard for the Resource element to indicate that users can specify all resources with the action; in this case, they can list all instances. The * wildcard is also necessary in cases where the API action does not support resource-level permissions (in this case, ec2:DescribeInstances). For more information about which ARNs you can use with which Amazon EC2 API actions, see Actions, resources, and condition keys for Amazon EC2.

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.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": ["ec2:StopInstances", "ec2:StartInstances"],
    },
    {
      "Effect": "Allow",
      "Action": "ec2:TerminateInstances",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/purpose": "test"
        }
      }
    }
  ]
}
```

Work with volumes

Examples

- Example: Attach and detach volumes (p. 1071)
- Example: Create a volume (p. 1071)
- Example: Create a volume with tags (p. 1072)
Example: Attach and detach 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.

```json
{
  "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"
      ],
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/volume_user": "${aws:username}"
        }
      }
    }
  ]
}
```

Example: Create 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.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateVolume"
      ],
      "Condition": {
        "NumericLessThan": {
          "ec2:VolumeSize": "20"
        }
      }
    }
  ]
}
```
Amazon Elastic Compute Cloud
User Guide for Windows Instances
IAM policies

Example: Create 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 Grant permission to tag resources during creation (p. 1065).

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowCreateTaggedVolumes",
         "Effect": "Allow",
         "Action": "ec2:CreateVolume",
         "Condition": {
            "StringEquals": {
               "aws:RequestTag/costcenter": "115",
               "aws:RequestTag/stack": "prod"
            },
            "ForAllValues:StringEquals": {
               "aws:TagKeys": ["costcenter","stack"]
            }
         }
      },
      {
         "Effect": "Allow",
         "Action": ["ec2:CreateTags"],
         "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.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      { "Effect": "Allow", 
         "Action": ["ec2:CreateTags"],
         "Condition": {
            "StringEquals": {
               "ec2:CreateAction" : "CreateVolume"
            }
         }
      }
   ]
}
```
IAM policies

"Action": "ec2:CreateVolume",
"Resource": "*"
},
{
"Effect": "Allow",
"Action": [ "ec2:CreateTags" ],
"Condition": {
"StringEquals": {
"aws:RequestTag/purpose": "test",
"ec2:CreateAction" : "CreateVolume"
},
"ForAllValues:StringEquals": {
"aws:TagKeys": "purpose"
}
}
]}

Work with snapshots

The following are example policies for both CreateSnapshot (point-in-time snapshot of an EBS volume) and CreateSnapshots (multi-volume snapshots).

Examples

- Example: Create a snapshot (p. 1073)
- Example: Create snapshots (p. 1074)
- Example: Create a snapshot with tags (p. 1074)
- Example: Create snapshots with tags (p. 1075)
- Example: Copying snapshots (p. 1079)
- Example: Modify permission settings for snapshots (p. 1080)

Example: Create a snapshot

The following policy allows customers to use the CreateSnapshot API action. The customer can create snapshots only if the volume is encrypted and only if the volume size is less than 20 GiB.

```
{
    "Version": "2012-10-17",
    "Statement": [ {
        "Effect": "Allow",
        "Action": "ec2:CreateSnapshot",
    }, {
        "Effect": "Allow",
        "Action": "ec2:CreateSnapshot",
        "Condition": {
            "NumericLessThan": {
                "ec2:VolumeSize": "20"
            },
            "Bool": {
                "ec2:Encrypted": "true"
            }
        }
    ]
}
```
Example: Create snapshots

The following policy allows customers to use the `CreateSnapshots` API action. The customer can create snapshots only if all of the volumes on the instance are type GP2.

```json
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect":"Allow",
      "Action":"ec2:CreateSnapshots",
      "Resource":[
        "arn:aws:ec2:us-east-1::snapshot/*",
        "arn:aws:ec2:*:*:instance/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":"ec2:CreateSnapshots",
      "Resource":"arn:aws:ec2:us-east-1:*:volume/*",
      "Condition":{
        "StringLikeIfExists":{
          "ec2:VolumeType":"gp2"
        }
      }
    }
  ]
}
```

Example: Create a snapshot with tags

The following policy includes the `aws:RequestTag` condition key that requires the customer to apply the tags `costcenter=115` and `stack=prod` to any new snapshot. The `aws:TagKeys` condition key uses the `ForAllValues` modifier to indicate that only the keys `costcenter` and `stack` can be specified in the request. The request fails if either of these conditions is not met.

For resource-creating actions that apply tags, customers must also have permissions to use the `CreateTags` action. The third statement uses the `ec2:CreateAction` condition key to allow customers to create tags only in the context of `CreateSnapshot`. Customers cannot tag existing volumes or any other resources. For more information, see Grant permission to tag resources during creation (p. 1065).

```json
{
  "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":{
          "ec2:VolumeType":"
        }
      }
    }
  ]
}
```
Example: Create snapshots with tags

The following policy includes the `aws:RequestTag` condition key that requires the customer to apply the tags `costcenter=115` and `stack=prod` to any new snapshot. The `aws:TagKeys` condition key uses the `ForAllValues` modifier to indicate that only the keys `costcenter` and `stack` can be specified in the request. The request fails if either of these conditions is not met.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CreateSnapshots",
      "Resource": [
        "arn:aws:ec2:us-east-1::snapshot/*",
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ec2:*:*:volume/*"
      ],
    },
    {
      "Sid": "AllowCreateTaggedSnapshots",
      "Effect": "Allow",
      "Action": "ec2:CreateSnapshots",
      "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/costcenter": "115",
          "aws:RequestTag/stack": "prod"
        },
        "ForAllValues: StringEquals": {
          "aws:TagKeys": [
            "costcenter",
            "stack"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/costcenter": "115",
          "aws:RequestTag/stack": "prod"
        },
        "ForAllValues: StringEquals": {
          "aws:TagKeys": [
            "costcenter",
            "stack"
          ]
        }
      }
    }
  ]
}
```
The following policy allows customers to create a snapshot without having to specify tags. The CreateTags action is evaluated only if tags are specified in the CreateSnapshot or CreateSnapshots request. If a tag is specified, the tag must be purpose=test. No other tags are allowed in the request.

```json
{
  "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 snapshots to be created only if the source volume is tagged with 
**User:** `username` for the customer, and the snapshot itself is tagged with **Environment:** `Dev` and 
**User:** `username`. The customer can add additional tags to the snapshot.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshot",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/User": "${aws:username}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshot",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Environment": "Dev",
                    "aws:RequestTag/User": "${aws:username}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*"
        }
    ]
}
```

The following policy for CreateSnapshots allows snapshots to be created only if the source 
volume is tagged with **User:** `username` for the customer, and the snapshot itself is tagged with 
**Environment:** `Dev` and **User:** `username`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshots",
            "Resource": "arn:aws:ec2:us-east-1:*:instance/*",
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshots",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/User": "${aws:username}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*"
        }
    ]
}
```
IAM policies

```
"Action":"ec2:CreateSnapshots",
"Resource":"arn:aws:ec2:us-east-1::snapshot/*",
"Condition":{
  "StringEquals":{
    "aws:RequestTag/Environment":"Dev",
    "aws:RequestTag/User":"${aws:username}"
  }
},
{
  "Effect":"Allow",
  "Action":"ec2:CreateTags",
  "Resource":"arn:aws:ec2:us-east-1::snapshot/*"
}
```

The following policy allows deletion of a snapshot only if the snapshot is tagged with User:username for the customer.

```
{
  "Version":"2012-10-17",
  "Statement": [ 
    {
      "Effect":"Allow",
      "Action":"ec2:DeleteSnapshot",
      "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
      "Condition":{
        "StringEquals":{
          "ec2:ResourceTag/User":"${aws:username}"
        }
      }
    },
    {
      "Effect":"Deny",
      "Action":"ec2:CreateSnapshot",
      "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
      "Condition":{
        "ForAnyValue:StringEquals":{
          "aws:TagKeys":"stack"
        }
      }
    }
  ]
}
```

The following policy allows a customer to create a snapshot but denies the action if the snapshot being created has a tag key value=stack.

```
{
  "Version":"2012-10-17",
  "Statement": [ 
    {
      "Effect":"Allow",
      "Action": [ "ec2:CreateSnapshot", "ec2:CreateTags" ],
      "Resource":"*
    },
    {
      "Effect":"Deny",
      "Action":"ec2:CreateSnapshot",
      "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
      "Condition":{
        "ForAnyValue:StringEquals":{
          "aws:TagKeys":"stack"
        }
      }
    }
  ]
}
```
The following policy allows a customer to create snapshots but denies the action if the snapshots being created have a tag key value=stack.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshots",
                "ec2:CreateTags"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": "ec2:CreateSnapshots",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": "stack"
                }
            }
        }
    ]
}
```

The following policy allows you to combine multiple actions into a single policy. You can only create a snapshot (in the context of `CreateSnapshots`) when the snapshot is created in Region `us-east-1`. You can only create snapshots (in the context of `CreateSnapshots`) when the snapshots are being created in the Region `us-east-1` and when the instance type is `t2*`.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshots",
                "ec2:CreateSnapshot",
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2::*:instance/*",
                "arn:aws:ec2::*:snapshot/*",
                "arn:aws:ec2::*:volume/*"
            ],
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "ec2:Region": "us-east-1"
                },
                "StringLikeIfExists": {
                    "ec2:InstanceType": ["t2.*"]
                }
            }
        }
    ]
}
```

Example: Copying snapshots

Resource-level permissions specified for the `CopySnapshot` action apply to the new snapshot only. They cannot be specified for the source snapshot.
The following example policy allows principals to copy snapshots only if the new snapshot is created with tag key of `purpose` and a tag value of `production` (purpose=production).

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCopySnapshotWithTags",
      "Effect": "Allow",
      "Action": "ec2:CopySnapshot",
      "Resource": "arn:aws:ec2:*:123456789012:snapshot/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/purpose": "production"
        }
      }
    }
  ]
}
```

**Example: Modify 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.

```json
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect":"Allow",
      "Action":"ec2: ModifySnapshotAttribute",
      "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
      "Condition":{
        "StringEquals":{
          "ec2:ResourceTag/user-name":"${aws:username}"
        }
      }
    }
  ]
}
```

**Launch instances (RunInstances)**

The `RunInstances` API action launches one or more On-Demand Instances or one or more Spot Instances. `RunInstances` requires an AMI and creates an instance. Users can specify a key pair and security group in the request. Launching into a VPC requires a subnet, and creates a network interface. Launching from an Amazon EBS-backed AMI creates a volume. Therefore, the user must have permissions to use these Amazon EC2 resources. You can create a policy statement that requires users to specify an optional parameter on `RunInstances`, or restricts users to particular values for a parameter.

For more information about the resource-level permissions that are required to launch an instance, see Actions, resources, and condition keys for Amazon EC2.

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 Work with instances (p. 1069).

**Resources**

1080
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:ami/ami-9e1670f7",
                "arn:aws:ec2:region:ami/ami-45cf5c3c",
                "arn:aws:ec2:account:instance/*",
                "arn:aws:ec2:account:volume/*",
                "arn:aws:ec2:account:security-group/*",
                "arn:aws:ec2:account:subnet/*",
                "arn:aws:ec2: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:account:instance/*",
                "arn:aws:ec2:account:volume/*",
                "arn:aws:ec2:account:security-group/*",
                "arn:aws:ec2:account:subnet/*",
                "arn:aws:ec2:account:network-interface/*"
            ]
        }
    ]
}
```
IAM policies

"arn:aws:ec2:region:account:subnet/**",
"arn:aws:ec2:region:account:volume/**,
"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.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region:account:instance/**
      ],
      "Condition": {
        "StringEquals": {
          "ec2:InstanceType": ["t2.micro", "t2.small"]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region::image/ami-**",
        "arn:aws:ec2:region:account:subnet/**,
        "arn:aws:ec2:region:account: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.

```json
{
  "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"]
        }
      }
    }
  ]
}
```
IAM policies

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::image/ami-*",
                "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/**
            ]
        }
    ]
}
```

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": [
            ],
            "Condition": {
                "ArnNotEquals": {
                }
            }
        }
    ]
}
```
IAM policies

{  
  "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:::*::volume/*",  
        "arn:aws:ec2:::*::key-pair/*",  
        "arn:aws:ec2:::*::security-group/*"  
      ]  
    }  
  ]  
}

Tags

Tag instances on creation

The following policy allows users to launch instances and tag the instances during creation. For resource-creating actions that apply tags, users must have permissions to use the CreateTags action. The second statement uses the ec2:CreateAction condition key to allow users to create tags only in the context of RunInstances, and only for instances. Users cannot tag existing resources, and users cannot tag volumes using the RunInstances request.
For more information, see Grant permission to tag resources during creation (p. 1065).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "RunInstances"
        }
      }
    }
  ]
}
```

**Tag instances and volumes on creation with specific tags**

The following policy includes the `aws:RequestTag` condition key that requires users to tag any instances and volumes that are created by `RunInstances` with the tags `environment=production` and `purpose=webserver`. The `aws:TagKeys` condition key uses the `ForAllValues` modifier to indicate that only the keys `environment` and `purpose` are allowed in the request (no other tags can be specified). If no tags are specified in the request, the request fails.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:region::image/*",
        "arn:aws:ec2:region:account:subnet/*",
        "arn:aws:ec2:region:account: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"
        }
      }
    }
  ]
}
```
Tag instances and volumes on creation with at least one specific tag

The following policy uses the ForAnyValue modifier on the aws:TagKeys condition to indicate that at least one tag must be specified in the request, and it must contain the key environment or webserver. The tag must be applied to both instances and volumes. Any tag values can be specified in the request.

```json
{
    "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:security-group/*",
                "arn:aws:ec2:region:account:key-pair/*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": ["environment","webserver"]
                }
            }
        },
        {
            "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"
        }
    ]
}
```
If instances are tagged on creation, they must be tagged with a specific tag

In the following policy, users do not have to specify tags in the request, but if they do, the tag must be `purpose=test`. No other tags are allowed. Users can apply the tags to any taggable resource in the `RunInstances` request.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["ec2:RunInstances"],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": ["ec2:CreateTags"],
      "Resource": "arn:aws:ec2:region:account:*/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/purpose": "test",
          "ec2:CreateAction": "RunInstances"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": "purpose"
        }
      }
    }
  ]
}
```

To disallow anyone called tag on create for `RunInstances`

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRun",
      "Effect": "Allow",
      "Action": ["ec2:RunInstances"],
      "Resource": ["arn:aws:ec2:us-east-1::image/**",
                    "arn:aws:ec2:us-east-1::subnet/**",
                    "arn:aws:ec2:us-east-1::network-interface/**",
                    "arn:aws:ec2:us-east-1::security-group/**",
                    "arn:aws:ec2:us-east-1::key-pair/**",
```

1087
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.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowRun",
         "Effect": "Allow",
         "Action": ["ec2:RunInstances"],
         "Resource": [
            "arn:aws:ec2:us-east-1:image/*",
            "arn:aws:ec2:us-east-1:subnet/*",
            "arn:aws:ec2:us-east-1:network-interface/*",
            "arn:aws:ec2:us-east-1:security-group/*",
            "arn:aws:ec2:us-east-1:key-pair/*",
            "arn:aws:ec2:us-east-1:volume/*",
            "arn:aws:ec2:us-east-1:instance/*",
            "arn:aws:ec2:us-east-1:spot-instances-request/*"
         ]
      },
      {
         "Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": "ec2:RunInstances",
         "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*",
         "Condition": {
            "StringEquals": {
               "aws:RequestTag/environment": "production"
            }
         }
      }
   ]
}
```
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.
"Action": "ec2:RunInstances",
"Resource": [
    "arn:aws:ec2::*:image/ami-*",
    "arn:aws:ec2::*:account:network-interface/*",
    "arn:aws:ec2::*:account:instance/*",
    "arn:aws:ec2::*:account:subnet/*",
    "arn:aws:ec2::*:account:volume/*",
    "arn:aws:ec2::*:account:key-pair/*",
    "arn:aws:ec2::*:account:security-group/*"
]}

Launch templates

In the following example, users can launch instances, but only if they use a specific launch template (lt-09477bcd97b0d310e). Users can override any parameters in the launch template by specifying the parameters in the RunInstances action.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/lt-09477bcd97b0d310e"
                }
            }
        }
    ]
}
```

In this example, users can launch instances only if they use a launch template. The policy uses the ec2:IsLaunchTemplateResource condition key to prevent users from overriding any pre-existing ARNs in the launch template.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/**"
                },
                "Bool": {
                    "ec2:IsLaunchTemplateResource": "true"
                }
            }
        }
    ]
}
```

The following example policy allows users 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
IAM policies 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",
      "Condition": {
        "ArnLike": {
          "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/**"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "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",
      "Condition": {
        "ArnLike": {
          "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/**"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Purpose": "Webservers"
        }
      }
    }
  ]
}
```
Work with Spot Instances

You can use the RunInstances action to create Spot Instance requests, and tag the Spot Instance requests on create. The resource to specify for RunInstances is `spot-instances-request`.

The `spot-instances-request` resource is evaluated in the IAM policy as follows:

- If you don't tag a Spot Instance request on create, Amazon EC2 does not evaluate the `spot-instances-request` resource in the RunInstances statement.
- If you tag a Spot Instance request on create, Amazon EC2 evaluates the `spot-instances-request` resource in the RunInstances statement.

Therefore, for the `spot-instances-request` resource, the following rules apply to the IAM policy:

- If you use RunInstances to create a Spot Instance request and you don't intend to tag the Spot Instance request on create, you don't need to explicitly allow the `spot-instances-request` resource; the call will succeed.
- If you use RunInstances to create a Spot Instance request and intend to tag the Spot Instance request on create, you must include the `spot-instances-request` resource in the RunInstances allow statement, otherwise the call will fail.
- If you use RunInstances to create a Spot Instance request and intend to tag the Spot Instance request on create, you must specify the `spot-instances-request` resource or * wildcard in the CreateTags allow statement, otherwise the call will fail.

You can request Spot Instances using RunInstances or RequestSpotInstances. The following example IAM policies apply only when requesting Spot Instances using RunInstances.

Example: Request Spot Instances using RunInstances

The following policy allows users to request Spot Instances by using the RunInstances action. The `spot-instances-request` resource, which is created by RunInstances, requests Spot Instances.

Note

To use RunInstances to create Spot Instance requests, you can omit `spot-instances-request` from the Resource list if you do not intend to tag the Spot Instance requests on create. This is because Amazon EC2 does not evaluate the `spot-instances-request` resource in the RunInstances statement if the Spot Instance request is not tagged on create.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": [ "ec2:RunInstances"
            ],
```
Warning

**NOT SUPPORTED – Example: Deny users permission to request Spot Instances using RunInstances**

The following policy is not supported for the `spot-instances-request` resource. The following policy is meant to give users the permission to launch On-Demand Instances, but deny users the permission to request Spot Instances. The `spot-instances-request` resource, which is created by RunInstances, is the resource that requests Spot Instances. The second statement is meant to deny the RunInstances action for the `spot-instances-request` resource. However, this condition is not supported because Amazon EC2 does not evaluate the `spot-instances-request` resource in the RunInstances statement if the Spot Instance request is not tagged on create.

```json
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Sid": "AllowRun",
        "Effect": "Allow",
        "Action": ["ec2:RunInstances"
    },
    {
        "Sid": "DenySpotInstancesRequests - NOT SUPPORTED - DO NOT USE!",
        "Effect": "Deny",
        "Action": ["ec2:RunInstances",
        "Resource": "arn:aws:ec2:us-east-1:*:spot-instances-request/*"
    }
    ]
}
```

**Example: Tag Spot Instance requests on create**

The following policy allows users to tag all resources that are created during instance launch. The first statement allows RunInstances to create the listed resources. The `spot-instances-request` resource, which is created by RunInstances, is the resource that requests Spot Instances. The second statement provides a * wildcard to allow all resources to be tagged when they are created at instance launch.

**Note**

If you tag a Spot Instance request on create, Amazon EC2 evaluates the `spot-instances-request` resource in the RunInstances statement. Therefore, you must explicitly allow the `spot-instances-request` resource for the RunInstances action, otherwise the call will fail.
"Statement": [  
  {  
    "Sid": "AllowRun",  
    "Effect": "Allow",  
    "Action": [  
      "ec2:RunInstances"
    ],  
    "Resource": [  
      "arn:aws:ec2:us-east-1::image/**",  
      "arn:aws:ec2:us-east-1::*:subnet/**",  
      "arn:aws:ec2:us-east-1::*:network-interface/**",  
      "arn:aws:ec2:us-east-1::*:security-group/**",  
      "arn:aws:ec2:us-east-1::*:key-pair/**",  
      "arn:aws:ec2:us-east-1::*:volume/**",  
      "arn:aws:ec2:us-east-1::*:instance/**",  
      "arn:aws:ec2:us-east-1::*:spot-instances-request/**"
    ]  
  },  
  {  
    "Sid": "TagResources",  
    "Effect": "Allow",  
    "Action": "ec2:CreateTags",  
    "Resource": "*"
  }
]  
}

Example: Deny tag on create for Spot Instance requests

The following policy denies users the permission to tag the resources that are created during instance launch.

The first statement allows RunInstances to create the listed resources. The spot-instances-request resource, which is created by RunInstances, is the resource that requests Spot Instances. The second statement provides a * wildcard to deny all resources being tagged when they are created at instance launch. If spot-instances-request or any other resource is tagged on create, the RunInstances call will fail.

{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AllowRun",  
      "Effect": "Allow",  
      "Action": [  
        "ec2:RunInstances"
      ],  
      "Resource": [  
        "arn:aws:ec2:us-east-1::image/**",  
        "arn:aws:ec2:us-east-1::*:subnet/**",  
        "arn:aws:ec2:us-east-1::*:network-interface/**",  
        "arn:aws:ec2:us-east-1::*:security-group/**",  
        "arn:aws:ec2:us-east-1::*:key-pair/**",  
        "arn:aws:ec2:us-east-1::*:volume/**",  
        "arn:aws:ec2:us-east-1::*:instance/**",  
        "arn:aws:ec2:us-east-1::*:spot-instances-request/**"
      ]  
    },  
    {  
      "Sid": "DenyTagResources",  
      "Effect": "Deny",  
      "Action": "ec2:CreateTags",  
      "Resource": "*"
    }
  ]  
}
Warning

NOT SUPPORTED – Example: Allow creating a Spot Instance request only if it is assigned a specific tag

The following policy is not supported for the spot-instances-request resource.
The following policy is meant to grant RunInstances the permission to create a Spot Instance request only if the request is tagged with a specific tag.
The first statement allows RunInstances to create the listed resources.
The second statement is meant to grant users the permission to create a Spot Instance request only if the request has the tag environment=production. If this condition is applied to other resources created by RunInstances, specifying no tags results in an Unauthenticated error.
However, if no tags are specified for the Spot Instance request, Amazon EC2 does not evaluate the spot-instances-request resource in the RunInstances statement, which results in non-tagged Spot Instance requests being created by RunInstances.
Note that specifying another tag other than environment=production results in an Unauthenticated error, because if a user tags a Spot Instance request, Amazon EC2 evaluates the spot-instances-request resource in the RunInstances statement.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRun",
            "Effect": "Allow",
            "Action": ["ec2:RunInstances"],
            "Resource": ["arn:aws:ec2:us-east-1::image/*",
                          "arn:aws:ec2:us-east-1::*:subnet/*",
                          "arn:aws:ec2:us-east-1::*:network-interface/*",
                          "arn:aws:ec2:us-east-1::*:security-group/*",
                          "arn:aws:ec2:us-east-1::*:key-pair/*",
                          "arn:aws:ec2:us-east-1::*:volume/*",
                          "arn:aws:ec2:us-east-1::*:instance/*"
                        ]
        },
        {
            "Sid": "RequestSpotInstancesOnlyIfTagIs_environment=production - NOT SUPPORTED - DO NOT USE!",
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1::*:spot-instances-request/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                }
            }
        },
        {
            "Sid": "TagResources",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

Example: Deny creating a Spot Instance request if it is assigned a specific tag
The following policy denies RunInstances the permission to create a Spot Instance request if the request is tagged with environment=production.

The first statement allows RunInstances to create the listed resources.

The second statement denies users the permission to create a Spot Instance request if the request has the tag environment=production. Specifying environment=production as a tag results in an Unauthenticated error. Specifying other tags or specifying no tags will result in the creation of a Spot Instance request.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRun",
      "Effect": "Allow",
      "Action": ["ec2:RunInstances"],
    },
    {
      "Sid": "DenySpotInstancesRequests",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:us-east-1::*:spot-instances-request/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": "production"
        }
      }
    },
    {
      "Sid": "TagResources",
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": "*"
    }
  ]
}
```

**Example: Work 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.
Example: Tag 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.
The following policy allows users to tag any taggable resource that already has a tag with a key of `owner` and a value of the IAM username. In addition, users must specify a tag with a key of `anycompany:environment-type` and a value of either `test` or `prod` in the request. Users can specify additional tags in the request.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ec2:CreateTags"],
            "Resource": "arn:aws:ec2:region:account:*/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/anycompany:environment-type": ["test","prod"],
                    "ec2:ResourceTag/owner": "${aws:username}"}
            }
        }
    ]
}
```

You can create an IAM policy that allows users to delete specific tags for a resource. For example, the following policy allows users to delete tags for a volume if the tag keys specified in the request are `environment` or `cost-center`. Any value can be specified for the tag but the tag key must match either of the specified keys.

**Note**

If you delete a resource, all tags associated with the resource are also deleted. Users do not need permissions to use the `ec2:DeleteTags` action to delete a resource that has tags; they only need permissions to perform the deleting action.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ec2:DeleteTags"],
            "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.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ec2:DeleteTags"],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/anycompany:environment-type": ["test","prod"],
                    "ec2:ResourceTag/owner": "${aws:username}"}
            }
        }
    ]
}
```
Example: Work 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.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AssociateIamInstanceProfile",
        "ec2:ReplaceIamInstanceProfileAssociation",
        "ec2:DisassociateIamInstanceProfile"
      ],
      "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. 1115).
Example: Work 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": [
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:Vpc": "arn:aws:ec2:region:account:vpc/vpc-ec43eb89"
                }
            }
        }
    ]
}
```

Example: Allow 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-09345221264b0dd6. 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.
Example: Work 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.

```json
{
    "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.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteLaunchTemplateVersions"
            ],
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Purpose": "Testing"
                }
            }
        }
    ]
}
```
Work with instance metadata

The following policies ensure that users can only retrieve instance metadata (p. 579) using Instance Metadata Service Version 2 (IMDSv2). You can combine the following four policies into one policy with four statements. When combined as one policy, you can use the policy as a service control policy (SCP). It can work equally well as a **deny** policy that you apply to an existing IAM policy (taking away and limiting existing permission), or as an SCP that is applied globally across an account, an organizational unit (OU), or an entire organization.

**Note**
The following RunInstances metadata options policies must be used in conjunction with a policy that gives the principal permissions to launch an instance with RunInstances. If the principal does not also have RunInstances permissions, it will not be able to launch an instance. For more information, see the policies in **Work with instances (p. 1069)** and **Launch instances (RunInstances) (p. 1080)**.

**Important**
If you use Auto Scaling groups and you need to require the use of IMDSv2 on all new instances, your Auto Scaling groups must use **launch templates**.

When an Auto Scaling group uses a launch template, the ec2:RunInstances permissions of the IAM principal are checked when a new Auto Scaling group is created. They are also checked when an existing Auto Scaling group is updated to use a new launch template or a new version of a launch template.

Restrictions on the use of IMDSv1 on IAM principals for RunInstances are only checked when an Auto Scaling group that is using a launch template, is created or updated. For an Auto Scaling group that is configured to use the Latest or Default launch template, the permissions are not checked when a new version of the launch template is created. For permissions to be checked, you must configure the Auto Scaling group to use a **specific version** of the launch template.

**To enforce the use of IMDSv2 on instances launched by Auto Scaling groups, the following additional steps are required:**

1. Disable the use of launch configurations for all accounts in your organization by using either service control policies (SCPs) or IAM permissions boundaries for new principals that are created. For existing IAM principals with Auto Scaling group permissions, update their associated policies with this condition key. To disable the use of launch configurations, create or modify the relevant SCP, permissions boundary, or IAM policy with the "autoscaling:LaunchConfigurationName" condition key with the value specified as null.

2. For new launch templates, configure the instance metadata options in the launch template. For existing launch templates, create a new version of the launch template and configure the instance metadata options in the new version.

3. In the policy that gives any principal the permission to use a launch template, restrict association of $latest and $default by specifying "autoscaling:LaunchTemplateVersionSpecified": "true". By restricting the use to a specific version of a launch template, you can ensure that new instances will be launched using the version in which the instance metadata options are configured. For more information, see **LaunchTemplateSpecification** in the *Amazon EC2 Auto Scaling API Reference*, specifically the Version parameter.

4. For an Auto Scaling group that uses a launch configuration, replace the launch configuration with a launch template. For more information, see **Replacing a Launch Configuration with a Launch Template** in the *Amazon EC2 Auto Scaling User Guide.*
5. For an Auto Scaling group that uses a launch template, make sure that it uses a new launch template with the instance metadata options configured, or uses a new version of the current launch template with the instance metadata options configured. For more information, see update-auto-scaling-group in the AWS CLI Command Reference.

Examples

- Require the use of IMDSv2 (p. 1103)
- Specify maximum hop limit (p. 1103)
- Limit who can modify the instance metadata options (p. 1104)
- Require role credentials to be retrieved from IMDSv2 (p. 1104)

Require the use of IMDSv2

The following policy specifies that you can't call the RunInstances API unless the instance is also opted in to require the use of IMDSv2 (indicated by "ec2:MetadataHttpTokens": "required"). If you do not specify that the instance requires IMDSv2, you get an UnauthorizedOperation error when you call the RunInstances API.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "RequireImdsV2",
         "Effect": "Deny",
         "Action": "ec2:RunInstances",
         "Resource": "arn:aws:ec2:*:*:instance/*",
         "Condition": {
            "StringNotEquals": {
               "ec2:MetadataHttpTokens": "required"
            }
         }
      }
   ]
}
```

Specify maximum hop limit

The following policy specifies that you can't call the RunInstances API unless you also specify a hop limit, and the hop limit can't be more than 3. If you fail to do that, you get an UnauthorizedOperation error when you call the RunInstances API.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "MaxImdsHopLimit",
         "Effect": "Deny",
         "Action": "ec2:RunInstances",
         "Resource": "arn:aws:ec2:*:*:instance/*",
         "Condition": {
            "NumericGreaterThan": {
               "ec2:MetadataHttpPutResponseHopLimit": "3"
            }
         }
      }
   ]
}
```
Limit who can modify the instance metadata options

The following policy removes the ability for the general population of administrators to modify instance metadata options, and permits only users with the role ec2-imds-admins to make changes. If any principal other than the ec2-imds-admins role tries to call the ModifyInstanceMetadataOptions API, it will get a UnauthorizedOperation error. This statement could be used to control the use of the ModifyInstanceMetadataOptions API; there are currently no fine-grained access controls (conditions) for the ModifyInstanceMetadataOptions API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowOnlyImdsAdminsToModifySettings",
      "Effect": "Deny",
      "Action": "ec2:ModifyInstanceMetadataOptions",
      "Resource": "*",
      "Condition": {
        "StringNotLike": {
          "aws:PrincipalARN": "arn:aws:iam::*:role/ec2-imds-admins"
        }
      }
    }
  ]
}
```

Require role credentials to be retrieved from IMDSv2

The following policy specifies that if this policy is applied to a role, and the role is assumed by the EC2 service and the resulting credentials are used to sign a request, then the request must be signed by EC2 role credentials retrieved from IMDSv2. Otherwise, all of its API calls will get an UnauthorizedOperation error. This statement/policy can be applied generally because, if the request is not signed by EC2 role credentials, it has no effect.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RequireAllEc2RolesToUseV2",
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "NumericLessThan": {
          "ec2:RoleDelivery": "2.0"
        }
      }
    }
  ]
}
```

Example policies for working in the Amazon EC2 console

You can use IAM policies to grant users permissions to view and work with specific resources in the Amazon EC2 console. You can use the example policies in the previous section; however, they are designed for requests that are made with the AWS CLI or an AWS SDK. The console uses additional API actions for its features, so these policies may not work as expected. For example, a user that has permission to use only the DescribeVolumes API action will encounter errors when trying to view volumes in the console. This section demonstrates policies that enable users to work with specific parts of the console.
Tip
To help you work out which API actions are required to perform tasks in the console, you can use a service such as AWS CloudTrail. For more information, see the AWS CloudTrail User Guide. If your policy does not grant permission to create or modify a specific resource, the console displays an encoded message with diagnostic information. You can decode the message using the DecodeAuthorizationMessage API action for AWS STS, or the decode-authorization-message command in the AWS CLI.

Examples
- Example: Read-only access (p. 1105)
- Example: Use the EC2 launch wizard (p. 1106)
- Example: Work with volumes (p. 1109)
- Example: Work with security groups (p. 1110)
- Example: Work with Elastic IP addresses (p. 1112)
- Example: Work with Reserved Instances (p. 1112)

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. 1068). Users cannot perform any actions on those resources or create new resources, unless another statement grants them permission to do so.

View instances, AMIs, and snapshots
Alternatively, you can provide read-only access to a subset of resources. To do this, replace the * wildcard in the ec2:Describe API action with specific ec2:Describe actions for each resource. The following policy allows users to view all instances, AMIs, and snapshots in the Amazon EC2 console. The ec2:DescribeTags action allows users to view public AMIs. The console requires the tagging information to display public AMIs; however, you can remove this action to allow users to view only private AMIs.

```
{
    "Version": "2012-10-17",
    "Statement": 
        [{
            "Effect": "Allow",
            "Action": 
                ["ec2:DescribeInstances",
                 "ec2:DescribeImages",
                 "ec2:DescribeTags",
                 "ec2:DescribeSnapshots"],
            "Resource": "*
        }]
}
```

Note
The Amazon EC2 ec2:Describe* API actions do not support resource-level permissions, so you cannot control which individual resources users can view in the console. Therefore, the * wildcard is necessary in the Resource element of the above statement. For more information about which ARNs you can use with which Amazon EC2 API actions, see Actions, resources, and condition keys for Amazon EC2.

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.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "ec2:DescribeInstances",
            "cloudwatch:DescribeAlarms",
            "cloudwatch:GetMetricStatistics"
         ],
         "Resource": "*
      }
   ]
}
```

**Example: Use 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:DescribeInstanceTypes: To view and select an instance type.
- ec2:DescribeVpcs: To view the available network options.
- ec2:DescribeSubnets: To view all available subnets for the chosen VPC.
- ec2:DescribeSecurityGroups or ec2:CreateSecurityGroup: To view and select an existing security group, or to create a new one.
- ec2:DescribeKeyPairs or ec2:CreateKeyPair: To select an existing key pair, or to create a new one.
- ec2:AuthorizeSecurityGroupIngress: To add inbound rules.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "ec2:DescribeInstances",
            "ec2:DescribeImages",
            "ec2:DescribeInstanceTypes",
            "ec2:DescribeKeyPairs",
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups",
            "ec2:CreateSecurityGroup",
            "ec2:AuthorizeSecurityGroupIngress",
            "ec2:CreateKeyPair"
         ],
      }
   ]
}
```
You can add API actions to your policy to provide more options for users, for example:

- **ec2:DescribeAvailabilityZones**: To view and select a specific Availability Zone.
- **ec2:DescribeNetworkInterfaces**: To view and select existing network interfaces for the selected subnet.
- To add outbound rules to VPC security groups, users must be granted permission to use the `ec2:AuthorizeSecurityGroupEgress` API action. To modify or delete existing rules, users must be granted permission to use the relevant `ec2:RevokeSecurityGroup*` API action.
- **ec2:CreateTags**: To tag the resources that are created by `RunInstances`. For more information, see Grant permission to tag resources during creation (p. 1065). If users do not have permission to use this action and they attempt to apply tags on the tagging page of the launch wizard, the launch fails.

  **Important**  
  Be careful about granting users permission to use the `ec2:CreateTags` action, because doing so limits your ability to use the `ec2:ResourceTag` condition key to restrict their use of other resources. If you grant users permission to use the `ec2:CreateTags` action, they can change a resource's tag in order to bypass those restrictions. For more information, see Control access to EC2 resources using resource tags (p. 1067).

- To use Systems Manager parameters when selecting an AMI, you must add `ssm:DescribeParameters` and `ssm:GetParameters` to your policy. `ssm:DescribeParameters` grants your IAM users the permission to view and select Systems Manager parameters. `ssm:GetParameters` grants your IAM users the permission to get the values of the Systems Manager parameters. You can also restrict access to specific Systems Manager parameters. For more information, see Restrict access to specific Systems Manager parameters later in this section.

Currently, the Amazon EC2 `Describe*` API actions do not support resource-level permissions, so you cannot restrict which individual resources users can view in the launch wizard. However, you can apply resource-level permissions on the `ec2:RunInstances` API action to restrict which resources users can use to launch an instance. The launch fails if users select options that they are not authorized to use.

**Restrict access to a specific instance type, subnet, and Region**

The following policy allows users to launch `t2.micro` instances using AMIs owned by Amazon, and only into a specific subnet (`subnet-1a2b3c4d`). Users can only launch in the sa-east-1 Region. If users select a different Region, or select a different instance type, AMI, or subnet in the launch wizard, the launch fails.

The first statement grants users permission to view the options in the launch wizard or to create new ones, as explained in the example above. The second statement grants users permission to use the network interface, volume, key pair, security group, and subnet resources for the `ec2:RunInstances` action, which are required to launch an instance into a VPC. For more information about using the `ec2:RunInstances` action, see Launch instances (RunInstances) (p. 1080). The third and fourth statements grant users permission to use the instance and AMI resources respectively, but only if the instance is a `t2.micro` instance, and only if the AMI is owned by Amazon.

```json
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": "*"
  }]
}
```
"Effect": "Allow",
"Action": [
  "ec2:DescribeInstances",
  "ec2:DescribeImages",
  "ec2:DescribeInstanceTypes",
  "ec2:DescribeKeyPairs",
  "ec2:CreateKeyPair",
  "ec2:DescribeVpcs",
  "ec2:DescribeSubnets",
  "ec2:DescribeSecurityGroups",
  "ec2:CreateSecurityGroup",
  "ec2:AuthorizeSecurityGroupIngress"
],
"Resource": "*",
},
{
  "Effect": "Allow",
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:sa-east-1:111122223333:key-pair/*",
    "arn:aws:ec2:sa-east-1:111122223333:subnet/subnet-1a2b3c4d"
  ],
},
{
  "Effect": "Allow",
  "Action": "ec2:RunInstances",
  "Resource": [
  ],
  "Condition": {
    "StringEquals": {
      "ec2:InstanceType": "t2.micro"
    }
  }
},
{
  "Effect": "Allow",
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:sa-east-1::image/ami-*"
  ],
  "Condition": {
    "StringEquals": {
      "ec2:Owner": "amazon"
    }
  }
}]

Restrict access to specific Systems Manager parameters

The following policy grants access to use Systems Manager parameters with a specific name.

The first statement grants users the permission to view Systems Manager parameters when selecting an AMI in the launch wizard. The second statement grants users the permission to only use parameters that are named prod-*. 

{
  "Version": "2012-10-17",
  "Statement": [{

Example: Work 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).
Example: Work with security groups

View security groups and add and remove rules

The following policy grants users permission to view security groups in the Amazon EC2 console, to add and remove inbound and outbound rules, and to list and modify rule descriptions for existing security groups that have the tag `Department=Test`.

In the first statement, the `ec2:DescribeTags` action allows users to view tags in the console, which makes it easier for users to identify the security groups that they are allowed to modify.

```json
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeSecurityGroups",
            "ec2:DescribeSecurityGroupRules",
            "ec2:DescribeTags"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:AuthorizeSecurityGroupIngress",
            "ec2:RevokeSecurityGroupIngress",
            "ec2:AuthorizeSecurityGroupEgress",
            "ec2:RevokeSecurityGroupEgress",
            "ec2:ModifySecurityGroupRules",
            "ec2:UpdateSecurityGroupRuleDescriptionsIngress",
            "ec2:UpdateSecurityGroupRuleDescriptionsEgress"
        ],
        "Resource": [
        ],
        "Condition": {
            "StringEquals": {
                "ec2:ResourceTag/Department": "Test"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:ModifySecurityGroupRules"
        ],
        "Resource": [
            "arn:aws:ec2:region:111122223333:security-group-rule/*"
        ]
    }
}
```

Work 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:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSecurityGroup"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSecurityGroupIngress",
                "ec2:CreateSecurityGroupEgress"
            ],
            "Resource": "*"
        }
    ]
}
```
• ec2:CreateSecurityGroup: To create a new security group.
• ec2:DescribeVpcs: To view a list of existing VPCs in the VPC list.

With these permissions, users can create a new security group successfully, but they cannot add any rules to it. To work with rules in the Create Security Group dialog box, you can add the following API actions to your policy:

• ec2:AuthorizeSecurityGroupIngress: To add inbound rules.
• ec2:AuthorizeSecurityGroupEgress: To add outbound rules to VPC security groups.
• ec2:RevokeSecurityGroupIngress: To modify or delete existing inbound rules. This is useful to allow users to use the Copy to new feature in the console. This feature opens the Create Security Group dialog box and populates it with the same rules as the security group that was selected.
• ec2:RevokeSecurityGroupEgress: To modify or delete outbound rules for VPC security groups. This is useful to allow users to modify or delete the default outbound rule that allows all outbound traffic.
• ec2:DeleteSecurityGroup: To cater for when invalid rules cannot be saved. The console first creates the security group, and then adds the specified rules. If the rules are invalid, the action fails, and the console attempts to delete the security group. The user remains in the Create Security Group dialog box so that they can correct the invalid rule and try to create the security group again. This API action is not required, but if a user is not granted permission to use it and attempts to create a security group with invalid rules, the security group is created without any rules, and the user must add them afterward.
• ec2:UpdateSecurityGroupRuleDescriptionsIngress: To add or update descriptions of ingress (inbound) security group rules.
• ec2:UpdateSecurityGroupRuleDescriptionsEgress: To add or update descriptions of egress (outbound) security group rules.
• ec2:ModifySecurityGroupRules: To modify security group rules.
• ec2:DescribeSecurityGroupRules: To list security group 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.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSecurityGroups",
        "ec2:CreateSecurityGroup",
        "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteSecurityGroup",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:AuthorizeSecurityGroupEgress"
      ],
```
Example: Work 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.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "ec2:DescribeAddresses",
            "ec2:AllocateAddress",
            "ec2:DescribeInstances",
            "ec2:AssociateAddress"
         ],
         "Resource": "*"
      }
   ]
}
```

Example: Work 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.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
```

1112
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.

**AWS managed policies for Amazon Elastic Compute Cloud**

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to create IAM customer managed policies that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see AWS managed policies in the [IAM User Guide](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_policies.html).

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the `ReadOnlyAccess` AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see AWS managed policies for job functions in the [IAM User Guide](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_policies.html).

**AWS managed policy: AmazonEC2FullAccess**

You can attach the `AmazonEC2FullAccess` policy to your IAM identities. This policy grants permissions that allow full access to Amazon EC2.

To view the permissions for this policy, see AmazonEC2FullAccess in the AWS Management Console.

**AWS managed policy: AmazonEC2ReadOnlyAccess**

You can attach the `AmazonEC2ReadOnlyAccess` policy to your IAM identities. This policy grants permissions that allow read-only access to Amazon EC2.

To view the permissions for this policy, see AmazonEC2ReadOnlyAccess in the AWS Management Console.
AWS managed policy: AWSEC2FleetServiceRolePolicy

This policy is attached to the service-linked role named AWSServiceRoleForEC2Fleet to allow EC2 Fleet to request, launch, terminate, and tag instances on your behalf. For more information, see Service-linked role for EC2 Fleet (p. 693).

AWS managed policy: AWSEC2SpotFleetServiceRolePolicy

This policy is attached to the service-linked role named AWSServiceRoleForEC2SpotFleet to allow Spot Fleet to launch and manage instances on your behalf. For more information, see Service-linked role for Spot Fleet (p. 723).

AWS managed policy: AWSEC2SpotServiceRolePolicy

This policy is attached to the service-linked role named AWSServiceRoleForEC2Spot to allow Amazon EC2 to launch and manage Spot Instances on your behalf. For more information, see Service-linked role for Spot Instance requests (p. 294).

IAM roles for Amazon EC2

Applications must sign their API requests with AWS credentials. Therefore, if you are an application developer, you need a strategy for managing credentials for your applications that run on EC2 instances. For example, you can securely distribute your AWS credentials to the instances, enabling the applications on those instances to use your credentials to sign requests, while protecting your credentials from other users. However, it's challenging to securely distribute credentials to each instance, especially those that AWS creates on your behalf, such as Spot Instances or instances in Auto Scaling groups. You must also be able to update the credentials on each instance when you rotate your AWS credentials.

We designed IAM roles so that your applications can securely make API requests from your instances, without requiring you to manage the security credentials that the applications use. Instead of creating and distributing your AWS credentials, you can delegate permission to make API requests using IAM roles as follows:

1. Create an IAM role.
2. Define which accounts or AWS services can assume the role.
3. Define which API actions and resources the application can use after assuming the role.
4. Specify the role when you launch your instance, or attach the role to an existing instance.
5. Have the application retrieve a set of temporary credentials and use them.

For example, you can use IAM roles to grant permissions to applications running on your instances that need to use a bucket in Amazon S3. You can specify permissions for IAM roles by creating a policy in JSON format. These are similar to the policies that you create for IAM users. If you change a role, the change is propagated to all instances.

You can only attach one IAM role to an instance, but you can attach the same role to many 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. 1062) and the following example: Example: Work with IAM roles (p. 1099).

Contents

- Instance profiles (p. 1115)
Amazon Elastic Compute Cloud
User Guide for Windows Instances
IAM roles

• Retrieve security credentials from instance metadata (p. 1115)
• Grant an IAM user permission to pass an IAM role to an instance (p. 1116)
• Work with IAM roles (p. 1117)

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.

Retrieve security credentials from instance metadata
An application on the instance retrieves the security credentials provided by the role from the instance
metadata item iam/security-credentials/role-name. The application is granted the permissions
for the actions and resources that you've deﬁned for the role through the security credentials associated
with the role. These security credentials are temporary and we rotate them automatically. We make new
credentials available at least ﬁve minutes before the expiration of the old credentials.

Warning

If you use services that use instance metadata with IAM roles, ensure that you don't expose your
credentials when the services make HTTP calls on your behalf. The types of services that could
expose your credentials include HTTP proxies, HTML/CSS validator services, and XML processors
that support XML inclusion.
The following command retrieves the security credentials for an IAM role named s3access.
IMDSv2
PS C:\> $token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-ttl-seconds" =


IMDSv1

The following is example output.
{

"Code" : "Success",

1115


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. 579). For information about the instance metadata IP address, see Retrieve instance metadata (p. 586).

Grant 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 use the following API actions:

- iam:PassRole
- ec2:AssociateIamInstanceProfile
- ec2:ReplaceIamInstanceProfileAssociation

For example, the following IAM policy grants users permission to launch instances with an IAM role, or to attach or replace an IAM role for an existing instance using the AWS CLI.

**Note**
This policy grants IAM users access to all of 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.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances",
        "ec2:AssociateIamInstanceProfile",
        "ec2:ReplaceIamInstanceProfileAssociation"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": ["iam:PassRole",
      "Resource": "*"
      ]
    }
  ]
}
```

To grant users permission to launch instances with an IAM role, or to attach or replace an IAM role for an existing instance using the Amazon EC2 console, you must grant them permission to use iam:ListInstanceProfiles, iam:PassRole, ec2:AssociateIamInstanceProfile, and
Work 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
- Create an IAM role (p. 1117)
- Launch an instance with an IAM role (p. 1119)
- Attach an IAM role to an instance (p. 1120)
- Replace an IAM role (p. 1121)
- Detach an IAM role (p. 1122)
- Generate a policy for your IAM role based on access activity (p. 1123)

Create an IAM role

You must create an IAM role before you can launch an instance with that role or attach it to an instance.

To create an IAM role using the IAM console
1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose Roles, Create role.
3. On the Select role type page, choose EC2 and the EC2 use case. Choose Next: Permissions.
4. On the Attach permissions policy page, select an AWS managed policy that grants your instances access to the resources that they need.
5. On the Review page, enter a name for the role and choose Create role.

Alternatively, you can use the AWS CLI to create an IAM role. The following example creates an IAM role with a policy that allows the role to use an Amazon S3 bucket.

To create an IAM role and instance profile (AWS CLI)
1. Create the following trust policy and save it in a text file named ec2-role-trust-policy.json.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "ec2.amazonaws.com"},
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Create the s3access role and specify the trust policy that you created using the create-role command.

```bash
aws iam create-role --role-name s3access --assume-role-policy-document file://ec2-role-trust-policy.json
```

```json
{
  "Role": {
    "AssumeRolePolicyDocument": {
```
3. Create an access policy and save it in a text file named ec2-role-access-policy.json. For example, this policy grants administrative permissions for Amazon S3 to applications running on the instance.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["s3:*"],
            "Resource": ["*"]
        }
    ]
}
```

4. Attach the access policy to the role using the put-role-policy command.

```bash
aws iam put-role-policy --role-name s3access --policy-name S3-Permissions --policy-document file://ec2-role-access-policy.json
```

5. Create an instance profile named s3access-profile using the create-instance-profile command.

```bash
aws iam create-instance-profile --instance-profile-name s3access-profile
```

6. Add the s3access role to the s3access-profile instance profile.

```bash
aws iam add-role-to-instance-profile --instance-profile-name s3access-profile --role-name s3access
```

Alternatively, you can use the following AWS Tools for Windows PowerShell commands:

- New-IAMRole
Amazon Elastic Compute Cloud
User Guide for Windows Instances
IAM roles

• Register-IAMRolePolicy
• New-IAMInstanceProﬁle

Launch 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 might take several seconds for the permissions to propagate. If
your ﬁrst attempt to launch an instance with a role fails, wait a few seconds before trying again.
For more information, see Troubleshooting Working with Roles in the IAM User Guide.

To launch an instance with an IAM role (console)
1.

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

2.

On the dashboard, choose Launch instance.

3.

Select an AMI and instance type and then choose Next: Conﬁgure Instance Details.

4.

On the Conﬁgure Instance Details page, for IAM role, select the IAM role that you created.

Note

The IAM role list displays the name of the instance proﬁle that you created when you
created your IAM role. If you created your IAM role using the console, the instance proﬁle
was created for you and given the same name as the role. If you created your IAM role using
the AWS CLI, API, or an AWS SDK, you may have named your instance proﬁle diﬀerently.
5.

Conﬁgure any other details, then follow the instructions through the rest of the wizard, or choose
Review and Launch to accept default settings and go directly to the Review Instance Launch page.

6.

Review your settings, then choose Launch to choose a key pair and launch your instance.

7.

If you are using the Amazon EC2 API actions in your application, retrieve the AWS security
credentials made available on the instance and use them to sign the requests. The AWS SDK does
this for you.
IMDSv2

PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token}

IMDSv1
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.
1119


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.

```
```

### Attach an IAM role to an instance

To attach an IAM role to an instance that has no role, the instance can be in the stopped or running state.

#### New console

**To attach an IAM role to an instance**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance, choose Actions, Security, Modify IAM role.
4. Select the IAM role to attach to your instance, and choose Save.

#### Old console

**To attach an IAM role to an instance**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance, choose Actions, Instance Settings, Attach/Replace IAM role.
4. Select the IAM role to attach to your instance, and choose Apply.

### To attach an IAM role to an instance (AWS CLI)

1. If required, describe your instances to get the ID of the instance to which to attach the role.

```
aws ec2 describe-instances
```

2. Use the `associate-iam-instance-profile` command to attach the IAM role to the instance by specifying the instance profile. You can use the Amazon Resource Name (ARN) of the instance profile, or you can use its name.

```
aws ec2 associate-iam-instance-profile \
   --instance-id i-1234567890abcdef0 \
   --iam-instance-profile Name="TestRole-1"
```
"IamInstanceProfileAssociation": { 
  "InstanceId": "i-1234567890abcdef0",
  "State": "associating",
  "AssociationId": "iip-assoc-0d8529a48294120",
  "IamInstanceProfile": { 
    "Id": "AIPAJLNLDX3AMYZNYYAY",
    "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-1"
  }
}
}

Alternatively, use the following Tools for Windows PowerShell commands:

- Get-EC2Instance
- Register-EC2IamInstanceProfile

**Replace an IAM role**

To replace the IAM role on an instance that already has an attached IAM role, the instance must be in the running state. You can do this if you want to change the IAM role for an instance without detaching the existing one first. For example, you can do this to ensure that API actions performed by applications running on the instance are not interrupted.

**New console**

To replace an IAM role for an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions, Security, Modify IAM role**.
4. Select the IAM role to attach to your instance, and choose **Save**.

**Old console**

To replace an IAM role for an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions, Instance Settings, Attach/Replace IAM role**.
4. Select the IAM role to attach to your instance, and choose **Apply**.

**To replace an IAM role for an instance (AWS CLI)**

1. If required, describe your IAM instance profile associations to get the association ID for the IAM instance profile to replace.

   ```bash
   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.

   ```bash
   aws ec2 replace-iam-instance-profile-association
   ```
Alternatively, use the following Tools for Windows PowerShell commands:

- Get-EC2IamInstanceProfileAssociation
- Set-EC2IamInstanceProfileAssociation

**Detach an IAM role**

You can detach an IAM role from a running or stopped instance.

**New console**

To detach an IAM role from an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions**, **Security**, **Modify IAM role**.
4. For **IAM role**, choose **No IAM Role**. Choose **Save**.
5. In the confirmation dialog box, enter **Detach**, and then choose **Detach**.

**Old console**

To detach an IAM role from an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance, choose **Actions**, **Instance Settings**, **Attach/Replace IAM role**.
4. For **IAM role**, choose **No Role**. Choose **Apply**.
5. In the confirmation dialog box, choose **Yes, Detach**.

**To detach an IAM role from an instance (AWS CLI)**

1. If required, use `describe-iam-instance-profile-associations` to describe your IAM instance profile associations and get the association ID for the IAM instance profile to detach.

```bash
aws ec2 describe-iam-instance-profile-associations
```
2. Use the `disassociate-iam-instance-profile` command to detach the IAM instance profile using its association ID.

```bash
aws ec2 disassociate-iam-instance-profile --association-id iip-assoc-0044d817db6c0a4ba
```

Alternatively, use the following Tools for Windows PowerShell commands:

- `Get-EC2IamInstanceProfileAssociation`
- `Unregister-EC2IamInstanceProfile`

### Generate a policy for your IAM role based on access activity

When you first create an IAM role for your applications, you might sometimes grant permissions beyond what is required. Before launching your application in your production environment, you can generate an IAM policy that is based on the access activity for an IAM role. IAM Access Analyzer reviews your AWS CloudTrail logs and generates a policy template that contains the permissions that have been used by the role in your specified date range. You can use the template to create a managed policy with fine-grained permissions and then attach it to the IAM role. That way, you grant only the permissions that the role needs to interact with AWS resources for your specific use case. This helps you adhere to the best practice of granting least privilege. To learn more, see Generate policies based on access activity in the **IAM User Guide**.

### Authorize 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. 1139) and Custom security groups (p. 1140). 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.

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. 1469).

Add 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 to enable you to connect to your Windows instance from your IP address using RDP.

New console

To add a rule to a security group for inbound RDP traffic over IPv4 (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select your instance and, in bottom half of the screen, choose the Security tab. Security groups lists the security groups that are associated with the instance. Inbound rules displays a list of the inbound rules that are in effect for the instance.
4. For the security group to which you'll add the new rule, choose the security group ID link to open the security group.
6. On the Edit inbound rules page, do the following:
   a. Choose Add rule.
   b. For Type, choose RDP.
   c. For Source, choose My IP to automatically populate the field with the public IPv4 address of your local computer.
      Alternatively, for Source, choose Custom and enter the public IPv4 address of your computer or network in CIDR notation. For example, if your IPv4 address is 203.0.113.25,
**Amazon Elastic Compute Cloud**

**User Guide for Windows Instances**

**Network access**

Enter 203.0.113.25/32 to list this single IPv4 address in CIDR notation. If your company allocates addresses from a range, enter the entire range, such as 203.0.113.0/24.

For information about finding your IP address, see Before you start (p. 1124).

d. Choose **Save rules**.

**Old console**

**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. 1124).
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.

**New console**

**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 **Instances**.
3. Select your instance and, in bottom half of the screen, choose the **Security** tab. **Security groups** lists the security groups that are associated with the instance. **Inbound rules** displays a list of the inbound rules that are in effect for the instance.
4. For the security group to which you'll add the new rule, choose the security group ID link to open the security group.
5. On the **Inbound rules** tab, choose **Edit inbound rules**.
6. On the **Edit inbound rules** page, do the following:
   a. Choose **Add rule**.
   b. For **Type**, choose **RDP**.
   c. For **Source**, choose **Custom** and enter the IPv6 address of your computer in CIDR notation. For example, if your IPv6 address is 2001:db8:1234:1a00:9691:9503:25ad:1761, enter 2001:db8:1234:1a00:9691:9503:25ad:1761/128 to list the single IPv6 address in CIDR notation. If your company allocates addresses from a range, enter the entire range, such as 2001:db8:1234:1a00::/64.
   d. Choose **Save rules**.

1125
Old console

**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/](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 [Access 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)
     ```bash
     aws ec2 describe-instance-attribute --instance-id instance_id --attribute groupSet
     ```
   - **Get-EC2InstanceAttribute** (AWS Tools for Windows PowerShell)
     ```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)
     ```bash
     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)
     ```powershell
     PS C:\> $ip1 = @{ IpProtocol="tcp"; FromPort="3389"; ToPort="3389"; IpRanges="cidr_ip_range" };
     PS C:\> Grant-EC2SecurityGroupIngress -GroupId security_group_id -IpPermission @($ip1)
     ```

1126
Assign a security group to an instance

You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group.

After you launch an instance, you can change its security groups. For more information, see Changing an instance's security groups in the Amazon VPC User Guide.

Amazon EC2 key pairs and Windows instances

A key pair, consisting of a public key and a private key, is a set of security credentials that you use to prove your identity when connecting to an Amazon EC2 instance. Amazon EC2 stores the public key on your instance, and you store the private key. For Windows instances, the private key is required to decrypt the administrator password. You then use the decrypted password to connect to your instance. Anyone who possesses your private key can connect to your instances, so it’s important that you store your private key in a secure place.

When you launch an instance, you are prompted for a key pair (p. 397). If you plan to connect to the instance using RDP, you must specify a key pair. You can choose an existing key pair or create a new one. With Windows instances, you use the private key to obtain the administrator password and then log in using RDP. For more information about connecting to your instance, see Connect to your Windows instance (p. 413). For more information about key pairs and Linux instances, see Amazon EC2 key pairs and Linux instances in the Amazon EC2 User Guide for Linux Instances.

Because Amazon EC2 doesn't keep a copy of your private key, there is no way to recover a private key if you lose it. However, there can still be a way to connect to instances for which you've lost the private key. For more information, see Connect to your Windows instance if you lose your private key (p. 1134).

You can use Amazon EC2 to create your key pairs. You can also use a third-party tool to create your key pairs, and then import the public keys to Amazon EC2.

The keys that Amazon EC2 uses are ED25519 or 2048-bit SSH-2 RSA keys.

You can have up to 5,000 key pairs per Region.

Contents

- Create a key pair using Amazon EC2 (p. 1127)
- Create a key pair using a third-party tool and import the public key to Amazon EC2 (p. 1129)
- Tag a public key (p. 1130)
- Retrieve the public key from the private key (p. 1132)
- Retrieve the public key through instance metadata (p. 1132)
- Identify the key pair that was specified at launch (p. 1133)
- Verify your key pair’s fingerprint (p. 1133)
- Delete your key pair (p. 1134)
- Connect to your Windows instance if you lose your private key (p. 1134)

Create a key pair using Amazon EC2

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

To create your key pair

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under Network & Security, choose Key Pairs.
3. Choose Create key pair.
4. For Name, enter a descriptive name for the key pair. Amazon EC2 associates the public key with the name that you specify as the key name. A key name can include up to 255 ASCII characters. It can’t include leading or trailing spaces.
5. For Key pair type, choose either RSA or ED25519. Note that ED25519 keys are not supported for Windows instances, EC2 Instance Connect, or EC2 Serial Console.
6. For Private key file format, choose the format in which to save the private key. To save the private key in a format that can be used with OpenSSH, choose pem. To save the private key in a format that can be used with PuTTY, choose ppk.

If you chose ED25519 in the previous step, the Private key file format options do not appear, and the private key format defaults to pem.
7. To add a tag to the public key, choose Add tag, and enter the key and value for the tag. Repeat for each tag.
8. Choose Create key pair.
9. The private key file is automatically downloaded by your browser. The base file name is the name that you specified as the name of your key pair, and the file name extension is determined by the file format that you chose. Save the private key file in a safe place.

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

AWS CLI

To create your key pair

• Use the create-key-pair command as follows to generate the key pair and to save the private key to a .pem file.

For --key-name, specify a name for the public key. The name can be up to 255 ASCII characters.

For --key-type, specify either rsa or ed25519. If you do not include the --key-type parameter, an rsa key is created by default. Note that ED25519 keys are not supported for Windows instances, EC2 Instance Connect, and EC2 Serial Console.

--query "KeyMaterial" prints the private key material to the output.

--output text > my-key-pair.pem saves the private key material in a file with the .pem extension. The private key can have a name that’s different from the public key name, but for ease of use, use the same name.

aws ec2 create-key-pair  \
   --key-name my-key-pair  \
   --key-type rsa  \
   --query "KeyMaterial"  \
   --output text > my-key-pair.pem
Create a key pair using a third-party tool and import the public key to Amazon EC2

To create your key pair

Use the `New-EC2KeyPair` AWS Tools for Windows PowerShell command as follows to generate the key and save it to a `.pem` file.

For `-KeyName`, specify a name for the public key. The name can be up to 255 ASCII characters.

For `-KeyType`, specify either `rsa` or `ed25519`. If you do not include the `-KeyType` parameter, an `rsa` key is created by default. Note that `ed25519` keys are not supported for Windows instances, EC2 Instance Connect, and EC2 Serial Console.

`KeyMaterial` prints the private key material to the output.

```
Out-File -Encoding ascii -FilePath C:\path\my-key-pair.pem
```

Create a key pair using a third-party tool and import the public key to Amazon EC2

Instead of using Amazon EC2 to create your key pair, you can create an RSA or ED25519 key pair by using a third-party tool, and then import the public key to Amazon EC2.

Requirements for key pairs

- Supported types: RSA and ED25519. Amazon EC2 does not accept DSA keys.
  - Note that ED25519 keys are not supported for Windows instances, EC2 Instance Connect, and EC2 Serial Console.
- Supported formats:
  - OpenSSH public key format
  - SSH private key file format must be PEM
  - (RSA only) Base64 encoded DER format
  - (RSA only) SSH public key file format as specified in [RFC 4716](https://tools.ietf.org/html/rfc4716)
- 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. For example, you can use `ssh-keygen` (a tool provided with the standard OpenSSH installation). Alternatively, Java, Ruby, Python, and many other programming languages provide standard libraries that you can use to create an RSA or ED25519 key pair.

   **Important**
   The private key must be in the PEM format. For example, use `ssh-keygen -m PEM` to generate the OpenSSH key in the PEM format.

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 local file that has the .pem extension. For example, C:\keys\my-key-pair.pem. The file name extension for this file is important because only .pem files can be selected when connecting to your Windows instance from the EC2 console.

   **Important**

   Save the private key file in a safe place. You'll need to provide the name of your public key when you launch an instance, and the corresponding private key each time you connect to the instance.

After you have created the key pair, use one of the following methods to import your public key to 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, choose **Key Pairs**.
3. Choose **Import key pair**.
4. For **Name**, enter a descriptive name for the public key. The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

   **Note**

   When you connect to your instance from the EC2 console, the console suggests this name for the name of your private key file.

5. Either choose **Browse** to navigate to and select your public key, or paste the contents of your public key into the **Public key contents** field.
6. Choose **Import key pair**.
7. Verify that the public key that you imported appears in the list of key pairs.

**AWS CLI**

**To import the public key**

Use the **import-key-pair** AWS CLI command.

**To verify that the key pair was imported successfully**

Use the **describe-key-pairs** AWS CLI command.

**PowerShell**

**To import the public key**

Use the **Import-EC2KeyPair** AWS Tools for Windows PowerShell command.

**To verify that the key pair was imported successfully**

Use the **Get-EC2KeyPair** AWS Tools for Windows PowerShell command.

**Tag a public key**

To help categorize and manage the public keys that you've either created using Amazon EC2 or imported to Amazon EC2, you can tag them with custom metadata. For more information about how tags work, see Tag your Amazon EC2 resources (p. 1450).

You can view, add, and delete tags using one of the following methods.
Tag a public key

**Console**

**To view, add, or delete a tag for an existing public key**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Key Pairs**.
3. Select a public key, and then choose **Actions, Manage tags**.
4. The **Manage tags** page displays any tags that are assigned to the public key.
   - To add a tag, choose **Add tag**, and then enter the tag key and value. You can add up to 50 tags per key. For more information, see Tag restrictions (p. 1454).
   - To delete a tag, choose **Remove** next to the tag to delete.
5. Choose **Save**.

**AWS CLI**

**To view public key tags**

Use the **describe-tags** AWS CLI command. In the following example, you describe the tags for all of your public keys.

C:\> aws ec2 describe-tags --filters "Name=resource-type,Values=key-pair"

```json
{
  "Tags": [
    {
      "Key": "Environment",
      "ResourceId": "key-0123456789EXAMPLE",
      "ResourceType": "key-pair",
      "Value": "Production"
    },
    {
      "Key": "Environment",
      "ResourceId": "key-9876543210EXAMPLE",
      "ResourceType": "key-pair",
      "Value": "Production"
    }
  ]
}
```

**To describe the tags for a specific public key**

Use the **describe-key-pairs** AWS CLI command.

C:\> aws ec2 describe-key-pairs --key-pair-ids key-0123456789EXAMPLE

```json
{
  "KeyPairs": [
    {
      "KeyName": "MyKeyPair",
      "KeyPairId": "key-0123456789EXAMPLE",
      "Tags": [
        {
          "Key": "Environment",
          "Value": "Production"
        }
      ]
    }
  ]
}
```
To tag an existing public key

Use the `create-tags` AWS CLI command. In the following example, the existing key is tagged with Key=Cost-Center and Value=CC-123.

```
C:\> aws ec2 create-tags --resources key-0123456789EXAMPLE --tags Key=Cost-Center,Value=CC-123
```

To delete a tag from a public key

Use the `delete-tags` AWS CLI command. For examples, see `Examples` in the `AWS CLI Command Reference`.

PowerShell

To view public key tags

Use the `Get-EC2Tag` command.

To describe the tags for a specific public key

Use the `Get-EC2KeyPair` command.

To tag an existing public key

Use the `New-EC2Tag` command.

To delete a tag from a public key

Use the `Remove-EC2Tag` command.

Retrieve the public key from the private key

On your local Windows computer, you can use PuTTYgen to get the public key for your key pair.

Start PuTTYgen and choose `Load`. Select the `.ppk` or `.pem` private key file. PuTTYgen displays the public key under `Public key for pasting into OpenSSH authorized_keys file`. You can also view the public key by choosing `Save public key`, specifying a name for the file, saving the file, and then opening the file.

Retrieve the public key through instance metadata

The public key that you specified when you launched an instance is also available through the instance metadata. To view the public key that you specified when launching the instance, use the following command from your instance.

```
```

The following is an example output.

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQClKsfkNkuSevGj3eYheC53pjqF3maAhDFcV8BS7O6V
h2zItxChh+PnDSUaw+WNQn/m2phTk/a/g08je2oWbkM4yxyb/wB96xb1FveSPJu0p/d6RJhJ0I0iBXR
lsLnB1tntckIJ7PbtxUMXlvtwv3ryDUil8MTfjYtwB+QhTXUM0zce5Pj25/185e3tjnV3iAoG/cQk+0FzZ
```
Identify the key pair that was specified at launch

When you launch an instance, you are prompted for a key pair (p. 397). If you plan to connect to the instance using RDP, you must specify a key pair.

To identify the key pair that was specified at launch

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances, and then select your instance.
3. On the Details tab, under Instance details, the Key pair name field displays the name of the key pair that you specified when you launched the instance. The value of the Key pair name does not change even if you change the public key on the instance, or add key pairs.

Verify your key pair's fingerprint

On the Key Pairs page in the Amazon EC2 console, the Fingerprint column displays the fingerprints generated from your key pairs. AWS calculates the fingerprint differently depending on whether the key pair was generated by AWS or a third-party tool. If you created the key pair using AWS, the fingerprint is calculated using an SHA-1 hash function. If you created the key pair with a third-party tool and uploaded the public key to AWS, or if you generated a new public key from an existing AWS-created private key and uploaded it to AWS, the fingerprint is calculated using an MD5 hash function.

You can use the SSH2 fingerprint that's displayed on the Key Pairs page to verify that the private key you have on your local machine matches the public key stored in AWS. From the computer where you downloaded the private key file, generate an SSH2 fingerprint from the private key file. The output should match the fingerprint that's displayed in the console.

If you're using a Windows local machine, you can run the following commands using the Windows Subsystem for Linux (WSL). Install the WSL and a Linux distribution using the instructions in the Windows 10 Installation Guide. The example in the instructions installs the Ubuntu distribution of Linux, but you can install any distribution. You are prompted to restart your computer for the changes to take effect.

If you created your key pair using AWS, you can use the OpenSSL tools to generate a fingerprint as shown in the following example.

```
$ openssl pkcs8 -in path_to_private_key -inform PEM -outform DER -topk8 -nocrypt | openssl sha1 -c
```

If you created a key pair using a third-party tool and uploaded the public key to AWS, you can use the OpenSSL tools to generate the fingerprint as shown in the following example.

```
$ openssl rsa -in path_to_private_key -pubout -outform DER | openssl md5 -c
```

If you created an OpenSSH key pair using OpenSSH 7.8 or later and uploaded the public key to AWS, you can use ssh-keygen to generate the fingerprint as shown in the following examples.
Delete your key pair

When you delete a key pair using the following methods, you are only deleting the public key that you saved in Amazon EC2 when you created (p. 1127) or imported (p. 1129) the key pair. Deleting a key pair doesn't delete the public key from any instances that were previously launched using that key pair. It also doesn't delete the private key on your local computer. You can continue to connect to instances that you launched using a key pair that is subsequently deleted, as long as you still have the private key (.pem) file.

If you're using an Auto Scaling group (for example, in an Elastic Beanstalk environment), ensure that the key pair you're deleting is not specified in an associated launch template or launch configuration. If Amazon EC2 Auto Scaling detects an unhealthy instance, it launches a replacement instance. However, the instance launch fails if the key pair cannot be found. For more information, see Launch templates in the Amazon EC2 Auto Scaling User Guide.

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

Console

To delete your key pair

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Key Pairs.
3. Select the key pair to delete and choose Delete.
4. In the confirmation field, enter Delete and then choose Delete.

AWS CLI

To delete your key pair

Use the delete-key-pair AWS CLI command.

PowerShell

To delete your key pair

Use the Remove-EC2KeyPair AWS Tools for Windows PowerShell command.

Connect 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 must reset the password or create a new instance. For more information, see [Reset a lost or expired Windows administrator password](p. 1484). For steps to reset the password using an Systems Manager document, see [Walkthrough: Reset passwords and SSH keys on EC2 instances](in the AWS Systems Manager User Guide).

### Amazon EC2 security groups for Windows instances

A security group acts as a virtual firewall for your EC2 instances to control incoming and outgoing traffic. Inbound rules control the incoming traffic to your instance, and outbound rules control the outgoing traffic from your instance. When you launch an instance, you can specify one or more security groups. If you don’t specify a security group, Amazon EC2 uses the default security group. You can add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time. New and modified rules are automatically applied to all instances that are associated with the security group. When Amazon EC2 decides whether to allow traffic to reach an instance, it evaluates all of the rules from all of the security groups that are associated with the instance.

When you launch an instance in a VPC, you must specify a security group that's created for that VPC. After you launch an instance, you can change its security groups. Security groups are associated with network interfaces. Changing an instance's security groups changes the security groups associated with the primary network interface (eth0). For more information, see [Changing an instance's security groups](in the Amazon VPC User Guide). You can also change the security groups associated with any other network interface. For more information, see [Modify network interface attributes](p. 953).

Security is a shared responsibility between AWS and you. For more information, see [Security in Amazon EC2](p. 1051). AWS provides security groups as one of the tools for securing your instances, and you need to configure them to meet your security needs. If you have requirements that aren't fully met by security groups, you can maintain your own firewall on any of your instances in addition to using security groups.

To allow traffic to a Linux instance, see [Amazon EC2 security groups for Linux instances](in the Amazon EC2 User Guide for Linux Instances).

### Contents

- Security group rules (p. 1136)
- Security group connection tracking (p. 1137)
  - Untracked connections (p. 1138)
  - Example (p. 1138)
  - Throttling (p. 1139)
- Default and custom security groups (p. 1139)
  - Default security groups (p. 1139)
  - Custom security groups (p. 1140)
- Work with security groups (p. 1140)
  - Create a security group (p. 1141)
  - Copy a security group (p. 1142)
  - View your security groups (p. 1142)
  - Add rules to a security group (p. 1143)
  - Update security group rules (p. 1145)
  - Delete rules from a security group (p. 1147)
Security group rules

The rules of a security group control the inbound traffic that's allowed to reach the instances that are associated with the security group. The rules also control the outbound traffic that's allowed to leave them.

The following are the characteristics of security group rules:

- By default, security groups allow all outbound traffic. Note that Amazon EC2 blocks traffic on port 25 by default. For more information, see Restriction on email sent using port 25 (p. 1464).
- Security group rules are always permissive; you can't create rules that deny access.
- Security group rules enable you to filter traffic based on protocols and port numbers.
- Security groups are stateful—if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. For VPC security groups, this also means that responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules. For more information, see Security group connection tracking (p. 1137).
- You can add and remove rules at any time. Your changes are automatically applied to the instances that are associated with the security group.

The effect of some rule changes can depend on how the traffic is tracked. For more information, see Security group connection tracking (p. 1137).

- When you associate multiple security groups with an instance, the rules from each security group are effectively aggregated to create one set of rules. Amazon EC2 uses this set of rules to determine whether to allow access.

You can assign multiple security groups to an instance. Therefore, an instance can have hundreds of rules that apply. This might cause problems when you access the instance. We recommend that you condense your rules as much as possible.

When you create a rule, you can specify the following:

- **Name**: The name for the security group (for example, my-security-group).

  A name can be up to 255 characters in length. Allowed characters are a-z, A-Z, 0-9, spaces, and _,-:/ !@#$%^&*(). When the name contains trailing spaces, we trim the spaces when we save the name. For example, if you enter "Test Security Group " for the name, we store it as "Test Security Group".

- **Protocol**: The protocol to allow. The most common protocols are 6 (TCP), 17 (UDP), and 1 (ICMP).
Connection tracking

- **Port range**: For TCP, UDP, or a custom protocol, the range of ports to allow. You can specify a single port number (for example, 22), or range of port numbers (for example, 7000-8000).

- **ICMP type and code**: For ICMP, the ICMP type and code.

- **Source or destination**: The source (inbound rules) or destination (outbound rules) for the traffic. Specify one of these options:
  - An individual IPv4 address. You must use the /32 prefix length; for example, 203.0.113.1/32.
  - An individual IPv6 address. You must use the /128 prefix length; for example, 2001:db8:1234:1a00::123/128.
  - A range of IPv4 addresses, in CIDR block notation; for example, 203.0.113.0/24.
  - A range of IPv6 addresses, in CIDR block notation; for example, 2001:db8:1234:1a00::/64.
  - A prefix list ID, for example, pl-1234abc1234abc123. For more information, see Prefix lists in the Amazon VPC User Guide.
  - Another security group. This allows instances that are associated with the specified security group to access instances associated with this security group. Choosing this option does not add rules from the source security group to this security group. You can specify one of the following security groups:
    - The current security group
    - A different security group for the same VPC
    - A different security group for a peer VPC in a VPC peering connection

- **(Optional) Description**: You can add a description for the rule, which can help you identify it later. A description can be up to 255 characters in length. Allowed characters are a-z, A-Z, 0-9, spaces, and _-:/@#$\&*.

When you create a security group rule, AWS assigns a unique ID to the rule. You can use the ID of a rule when you use the API or CLI to modify or delete the rule.

When you specify a security group as the source or destination for a rule, the rule affects all instances that are associated with the security group. Incoming traffic is allowed based on the private IP addresses of the instances that are associated with the source security group (and not the public IP or Elastic IP addresses). For more information about IP addresses, see Amazon EC2 instance IP addressing (p. 893). If your security group rule references a security group in a peer VPC, and the referenced security group or VPC peering connection is deleted, the rule is marked as stale. For more information, see Working with Stale Security Group Rules in the Amazon VPC Peering Guide.

If there is more than one rule for a specific port, Amazon EC2 applies the most permissive rule. For example, if you have a rule that allows access to TCP port 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.

When you add, update, or remove rules, the changes are automatically applied to all instances associated with the security group.

**Security group 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. With this approach, security groups are stateful. This means that responses to inbound traffic are allowed to flow out of the instance regardless of outbound security group rules, and vice versa.

As an example, suppose that you initiate an ICMP ping command to your instances 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.
For protocols other than TCP, UDP, or ICMP, only the IP address and protocol number is tracked. If your instance sends traffic to another host (host B), and host B initiates the same type of traffic to your instance in a separate request within 600 seconds of the original request or response, your instance accepts it regardless of inbound security group rules. Your instance accepts it because it's regarded as response traffic.

To ensure that traffic is immediately interrupted when you remove a security group rule, or to ensure that all inbound traffic is subject to firewall rules, you can use a network ACL for your subnet. Network ACLs are stateless and therefore do not automatically allow response traffic. For more information, see Network ACLs in the Amazon VPC User Guide.

Untracked connections

Not all flows of traffic are tracked. If a security group rule permits TCP or UDP flows for all traffic (0.0.0.0/0 or ::/0) and there is a corresponding rule in the other direction that permits all response traffic (0.0.0.0/0 or ::/0) for all ports (0-65535), then that flow of traffic is not tracked. The response traffic is therefore allowed to flow based on the inbound or outbound rule that permits the response traffic, and not on tracking information.

An untracked flow of traffic is immediately interrupted if the rule that enables the flow is removed or modified. For example, if you have an open (0.0.0.0/0) outbound rule, and you remove a rule that allows all (0.0.0.0/0) inbound SSH (TCP port 22) traffic to the instance (or modify it such that the connection would no longer be permitted), your existing SSH connections to the instance are immediately dropped. The connection was not previously being tracked, so the change will break the connection. On the other hand, if you have a narrower inbound rule that initially allows the SSH connection (meaning that the connection was tracked), but change that rule to no longer allow new connections from the address of the current SSH client, the existing connection will not be broken by changing the rule.

Example

In the following example, the security group has specific inbound rules for TCP and ICMP traffic, and outbound rules that allow all outbound IPv4 and IPv6 traffic.

<table>
<thead>
<tr>
<th>Inbound rules</th>
<th>Source IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol type</td>
<td>Port number</td>
</tr>
<tr>
<td>TCP</td>
<td>22 (SSH)</td>
</tr>
<tr>
<td>TCP</td>
<td>80 (HTTP)</td>
</tr>
<tr>
<td>TCP</td>
<td>80 (HTTP)</td>
</tr>
<tr>
<td>ICMP</td>
<td>All</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outbound rules</th>
<th>Destination IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol type</td>
<td>Port number</td>
</tr>
<tr>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>All</td>
<td>All</td>
</tr>
</tbody>
</table>

- TCP traffic on port 22 (SSH) to and from the instance is tracked, because the inbound rule allows traffic from 203.0.113.1/32 only, and not all IP addresses (0.0.0.0/0).
- TCP traffic on port 80 (HTTP) to and from the instance is not tracked, because both the inbound and outbound rules allow all traffic (0.0.0.0/0 or ::/0).
- ICMP traffic is always tracked, regardless of rules.
If you remove the outbound rule from the security group, all traffic to and from the instance is tracked, including traffic on port 80 (HTTP).

**Throttling**

Amazon EC2 defines the maximum number of connections that can be tracked per instance. After the maximum is reached, any packets that are sent or received are dropped because a new connection cannot be established. When this happens, applications that send and receive packets cannot communicate properly.

To determine whether packets were dropped because the network traffic for your instance exceeded the maximum number of connections that can be tracked, use the `conntrack_allowance_exceeded` network performance metric. For more information, see Monitor network performance for your EC2 instance (p. 973).

Connections made through a Network Load Balancer are automatically tracked, even if the security group configuration does not require tracking. If you exceed the maximum number of connections that can be tracked per instance, we recommend that you scale either the number of instances registered with the load balancer or the size of the instances registered with the load balancer.

**Default and custom 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. If you don’t want your instances to use the default security group, you can create your own custom security groups and specify them when you launch your instances.

**Topics**

- Default security groups (p. 1139)
- Custom security groups (p. 1140)

**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 table describes the default rules for a default security group.

<table>
<thead>
<tr>
<th>Inbound rule</th>
<th>Source</th>
<th>Protocol</th>
<th>Port range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The security group ID (its own resource ID)</td>
<td>All</td>
<td>All</td>
<td>Allows inbound traffic from network interfaces and instances that are assigned to the same security group.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outbound rules</th>
<th>Destination</th>
<th>Protocol</th>
<th>Port range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0/0</td>
<td>All</td>
<td>All</td>
<td>Allows all outbound IPv4 traffic.</td>
<td></td>
</tr>
</tbody>
</table>
You can add or remove inbound and outbound rules for any default security group.

You can't delete a default security group. If you try to delete a default security group, you see the following error: `Client.CannotDelete: the specified group: "sg-51530134" name: "default" cannot be deleted by a user.`

**Custom security groups**

If you don't want your instances to use the default security group, you can create your own security groups and specify them when you launch your instances. You can create multiple security groups to reflect the different roles that your instances play; for example, a web server or a database server.

When you create a security group, you must provide it with a name and a description. Security group names and descriptions can be up to 255 characters in length, and are limited to the following characters:

a-z, A-Z, 0-9, spaces, and _-:/(),@!#$

A security group name cannot start with `sg-`. A security group name must be unique for the VPC.

The following are the default rules for a security group that you create:

- Allows no inbound traffic
- Allows all outbound traffic

After you've created a security group, you can change its inbound rules to reflect the type of inbound traffic that you want to reach the associated instances. You can also change its outbound rules.

For more information about the rules you can add to a security group, see Security group rules for different use cases (p. 1149).

**Work with security groups**

You can assign a security group to an instance when you launch the instance. When you add or remove rules, those changes are automatically applied to all instances to which you've assigned the security group. For more information, see Assign a security group to an instance (p. 1148).

After you launch an instance, you can change its security groups. For more information, see Change an instance's security group (p. 1149).

You can create, view, update, and delete security groups and security group rules using the Amazon EC2 console and the command line tools.

**Tasks**

- Create a security group (p. 1141)
- Copy a security group (p. 1142)
- View your security groups (p. 1142)
- Add rules to a security group (p. 1143)
- Update security group rules (p. 1145)
Create a security group

Although you can use the default security group for your instances, you might want to create your own groups to reflect the different roles that instances play in your system.

By default, new security groups start with only an outbound rule that allows all traffic to leave the instances. You must add rules to enable any inbound traffic or to restrict the outbound traffic.

A security group can be used only in the VPC for which it is created.

New console

To create a security group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Security Groups.
3. Choose Create security group.
4. In the Basic details section, do the following.
   a. Enter a descriptive name and brief description for the security group. They can't be edited after the security group is created. The name and description can be up to 255 characters long. The valid characters are a–z, A–Z, 0–9, spaces, and . _ : / # @ \ [ ] += & ; { } ! $ *.
   b. For VPC, choose the VPC.
5. You can add security group rules now, or you can add them later. For more information, see Add rules to a security group (p. 1143).
6. You can add tags now, or you can add them later. To add a tag, choose Add new tag and enter the tag key and value.
7. Choose Create security group.

Old console

To create a security group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Security Groups.
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 Add rules to a security group (p. 1143).

Command line

To create a security group

Use one of the following commands:
Copy a security group

You can create a new security group by creating a copy of an existing one. When you copy a security group, the copy is created with the same inbound and outbound rules as the original security group. If the original security group is in a VPC, the copy is created in the same VPC unless you specify a different one.

The copy receives a new unique security group ID and you must give it a name. You can also add a description.

You can't copy a security group from one Region to another Region.

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

New console

To copy a security group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Security Groups.
3. Select the security group to copy and choose Actions, Copy to new security group.
4. Specify a name and optional description, and change the VPC and security group rules if needed.
5. Choose Create.

Old console

To copy a security group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Security Groups.
3. Select the security group you want to copy, choose Actions, Copy to new.
4. The Create Security Group dialog opens, and is populated with the rules from the existing security group. Specify a name and description for your new security group. For VPC, choose the ID of the VPC. When you are done, choose Create.

View your security groups

You can view information about your security groups using one of the following methods.

New console

To view your security groups

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Security Groups.
3. Your security groups are listed. To view the details for a specific security group, including its inbound and outbound rules, choose its ID in the Security group ID column.
Old console

To view your security groups

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Security Groups.
3. (Optional) Select VPC ID from the filter list, then choose the ID of the VPC.
4. Select a security group. General information is displayed on the Description tab, inbound rules on the Inbound tab, outbound rules on the Outbound tab, and tags on the Tags tab.

Command line

To view your security groups

Use one of the following commands.

- describe-security-groups (AWS CLI)
- describe-security-group-rules (AWS CLI)

Add rules to a security group

When you add a rule to a security group, the new rule is automatically applied to any instances that are associated with the security group. There might be a short delay before the rule is applied. For more information, see Security group rules for different use cases (p. 1149) and Security group rules (p. 1136).

New console

To add an inbound rule to a security group

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Security Groups.
3. Select the security group, and choose Actions, Edit inbound rules.
4. For each rule, choose Add rule and do the following.
   a. For Type, choose the type of protocol to allow.
      - For TCP or UDP, you must enter the port range to allow.
      - For custom ICMP, you must choose the ICMP type name from Protocol, and, if applicable, the code name from Port range.
      - For any other type, the protocol and port range are configured automatically.
   b. For Source, do one of the following to allow traffic.
      - Choose Custom and then enter an IP address in CIDR notation, a CIDR block, another security group, or a prefix list.
      - Choose Anywhere to allow all traffic for the specified protocol to reach your instance. This option automatically adds the 0.0.0.0/0 IPv4 CIDR block as the source. This is acceptable for a short time in a test environment, but it's unsafe in production environments. In production, authorize only a specific IP address or range of addresses to access your instances.

      If your security group is in a VPC that's enabled for IPv6, this option automatically adds a rule for the ::/0 IPv6 CIDR block.
Choose **My IP** to allow inbound traffic from only your local computer's public IPv4 address.

c. For **Description**, optionally specify a brief description for the rule.

5. Choose **Preview changes, Save rules**.

**To add an outbound rule to a security group**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Security Groups**.
3. Select the security group, and choose **Actions, Edit outbound rules**.
4. For each rule, choose **Add rule** and do the following.
   a. For **Type**, choose the type of protocol to allow.
      - For TCP or UDP, you must enter the port range to allow.
      - For custom ICMP, you must choose the ICMP type name from **Protocol**, and, if applicable, the code name from **Port range**.
      - For any other type, the protocol and port range are configured automatically.
   b. For **Destination**, do one of the following.
      - Choose **Custom** and then enter an IP address in CIDR notation, a CIDR block, another security group, or a prefix list for which to allow outbound traffic.
      - Choose **Anywhere** to allow outbound traffic to all IP addresses. This option automatically adds the 0.0.0.0/0 IPv4 CIDR block as the destination.
      - If your security group is in a VPC that's enabled for IPv6, this option automatically adds a rule for the ::/0 IPv6 CIDR block.
      - Choose **My IP** to allow outbound traffic only to your local computer's public IPv4 address.
   c. (Optional) For **Description**, specify a brief description for the rule.

5. Choose **Preview changes, Confirm**.

**Old console**

**To add rules to a security group**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Security Groups** and select the security group.
3. On the **Inbound** tab, choose **Edit**.
4. In the dialog, choose **Add Rule** and do the following:
   - For **Type**, select the protocol.
   - If you select a custom TCP or UDP protocol, specify the port range in **Port Range**.
   - If you select a custom ICMP protocol, choose the ICMP type name from **Protocol**, and, if applicable, the code name from **Port Range**.
   - For **Source**, choose one of the following:
     - **Custom**: in the provided field, you must specify an IP address in CIDR notation, a CIDR block, or another security group.
     - **Anywhere**: automatically adds the 0.0.0.0/0 IPv4 CIDR block. This option enables all traffic of the specified type to reach your instance. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, authorize only a specific IP address or range of addresses to access your instance.
If your security group is in a VPC that's enabled for IPv6, the Anywhere option creates two rules—one for IPv4 traffic (0.0.0.0/0) and one for IPv6 traffic (::/0).

- My IP: automatically adds the public IPv4 address of your local computer.
- For Description, you can optionally specify a description for the rule.

For more information about the types of rules that you can add, see Security group rules for different use cases (p. 1149).

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.

    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.

Command line

To add rules to a security group

Use one of the following commands.

- authorize-security-group-ingress (AWS CLI)
- Grant-EC2SecurityGroupIngress (AWS Tools for Windows PowerShell)

To add one or more egress rules to a security group

Use one of the following commands.

- authorize-security-group-egress (AWS CLI)
- Grant-EC2SecurityGroupEgress (AWS Tools for Windows PowerShell)

Update security group rules

You can update a security group rule using one of the following methods. The updated rule is automatically applied to any instances that are associated with the security group.

New console

When you modify the protocol, port range, or source or destination of an existing security group rule using the console, the console deletes the existing rule and adds a new one for you.
To update a security group rule

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Security Groups.
3. Select the security group.
4. Choose Actions, Edit inbound rules to update a rule for inbound traffic or Actions, Edit outbound rules to update a rule for outbound traffic.
5. Update the rule as required.
6. Choose Preview changes, Confirm.

To tag a security group rule

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Security Groups.
3. Select the security group.
4. On the Inbound rules or Outbound rules tab, select the check box for the rule and then choose Manage tags.
5. The Manage tags page displays any tags that are assigned to the rule. To add a tag, choose Add tag and enter the tag key and value. To delete a tag, choose Remove next to the tag that you want to delete.
6. Choose Save changes.

Old console

When you modify the protocol, port range, or source or destination of an existing security group rule using the console, the console deletes the existing rule and adds a new one for you.

To update a security group rule

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Security Groups.
3. Select the security group to update, and choose the Inbound tab to update a rule for inbound traffic or the Outbound tab to update a rule for outbound traffic.
4. Choose Edit.
5. Modify the rule entry as required and choose Save.

Command line

You cannot modify the protocol, port range, or source or destination of an existing rule using the Amazon EC2 API or a command line tools. Instead, you must delete the existing rule and add a new rule. You can, however, update the description of an existing rule.

To update a rule

Use one the following command.

- modify-security-group-rules (AWS CLI)

To update the description for an existing inbound rule

Use one of the following commands.
Work with security groups

- **update-security-group-rule-descriptions-ingress** (AWS CLI)
- **Update-EC2SecurityGroupRuleIngressDescription** (AWS Tools for Windows PowerShell)

To update the description for an existing outbound rule

Use one of the following commands.

- **update-security-group-rule-descriptions-egress** (AWS CLI)
- **Update-EC2SecurityGroupRuleEgressDescription** (AWS Tools for Windows PowerShell)

To tag a security group rule

Use one of the following commands.

- **create-tags** (AWS CLI)
- **New-EC2Tag** (AWS Tools for Windows PowerShell)

Delete rules from a security group

When you delete a rule from a security group, the change is automatically applied to any instances associated with the security group.

You can delete rules from a security group using one of the following methods.

New console

To delete a security group rule

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group to update, choose **Actions**, and then choose **Edit inbound rules** to remove an inbound rule or **Edit outbound rules** to remove an outbound rule.
4. Choose the **Delete** button to the right of the rule to delete.
5. Choose **Preview changes**, **Confirm**.

Old console

To delete a security group rule

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Security Groups**.
3. Select a security group.
4. On the **Inbound** tab (for inbound rules) or **Outbound** tab (for outbound rules), choose **Edit**. Choose **Delete** (a cross icon) next to each rule to delete.
5. Choose **Save**.

Command line

To remove one or more ingress rules from a security group

Use one of the following commands.

- **revoke-security-group-ingress** (AWS CLI)
• Revoke-EC2SecurityGroupIngress (AWS Tools for Windows PowerShell)

To remove one or more egress rules from a security group

Use one of the following commands.
• revoke-security-group-egress (AWS CLI)
• Revoke-EC2SecurityGroupEgress (AWS Tools for Windows PowerShell)

Delete a security group

You can't delete a security group that is associated with an instance. You can't delete the default security group. You can't delete a security group that is referenced by a rule in another security group in the same VPC. If your security group is referenced by one of its own rules, you must delete the rule before you can delete the security group.

New console

To delete a security group
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Security Groups.
3. Select the security group to delete and choose Actions, Delete security group, Delete.

Old console

To delete a security group
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Security Groups.
4. Choose Yes, Delete.

Command line

To delete a security group

Use one of the following commands.
• delete-security-group (AWS CLI)
• Remove-EC2SecurityGroup (AWS Tools for Windows PowerShell)

Assign a security group to an instance

You can assign one or more security groups to an instance when you launch the instance. You can also specify one or more security groups in a launch template. The security groups will be assigned to all instances that are launched using the launch template.

• To assign a security group to an instance when you launch the instance, see Step 6: Configure Security Group (p. 397).
• To specify a security group in a launch template, see Step 6 of Create a new launch template using parameters you define (p. 399).
Change an instance's security group

After you launch an instance, you can change its security groups by adding or removing security groups. You can change the security groups when the instance is in the running or stopped state.

New console

To change the security groups for an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select your instance, and then choose Actions, Security, Change security groups.
4. For Associated security groups, select a security group from the list and choose Add security group.
   To remove an already associated security group, choose Remove for that security group.
5. Choose Save.

Old console

To change the security groups for an instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select your instance, and then choose Actions, Networking, Change Security Groups.
4. To add one or more security groups, select its check box.
   To remove an already associated security group, clear its check box.

Command line

To change the security groups for an instance using the command line

Use one of the following commands.

- modify-instance-attribute (AWS CLI)
- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

Security group rules for different use cases

You can create a security group and add rules that reflect the role of the instance that's associated with the security group. For example, an instance that's configured as a web server needs security group rules that allow inbound HTTP and HTTPS access. Likewise, a database instance needs rules that allow access for the type of database, such as access over port 3306 for MySQL.

The following are examples of the kinds of rules that you can add to security groups for specific kinds of access.

Examples

- Web server rules (p. 1150)
- Database server rules (p. 1150)
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.

<table>
<thead>
<tr>
<th>Protocol type</th>
<th>Protocol number</th>
<th>Port</th>
<th>Source IP</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>6</td>
<td>80 (HTTP)</td>
<td>0.0.0.0/0</td>
<td>Allows inbound HTTP access from any IPv4 address</td>
</tr>
<tr>
<td>TCP</td>
<td>6</td>
<td>443 (HTTPS)</td>
<td>0.0.0.0/0</td>
<td>Allows inbound HTTPS access from any IPv4 address</td>
</tr>
<tr>
<td>TCP</td>
<td>6</td>
<td>80 (HTTP)</td>
<td>::/0</td>
<td>Allows inbound HTTP access from any IPv6 address</td>
</tr>
<tr>
<td>TCP</td>
<td>6</td>
<td>443 (HTTPS)</td>
<td>::/0</td>
<td>Allows inbound HTTPS access from any IPv6 address</td>
</tr>
</tbody>
</table>

Database server rules

The following inbound rules are examples of rules you might add for database access, depending on what type of database you’re running on your instance. For more information about Amazon RDS instances, see the Amazon RDS User Guide.

For the source IP, specify one of the following:

- A specific IP address or range of IP addresses (in CIDR block notation) in your local network
- A security group ID for a group of instances that access the database

<table>
<thead>
<tr>
<th>Protocol type</th>
<th>Protocol number</th>
<th>Port</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>6</td>
<td>1433 (MS SQL)</td>
<td>The default port to access a Microsoft SQL Server database, for example, on an Amazon RDS instance</td>
</tr>
</tbody>
</table>
## Security group rules for different use cases

<table>
<thead>
<tr>
<th>Protocol type</th>
<th>Protocol number</th>
<th>Port</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>6</td>
<td>3306 (MySQL/Aurora)</td>
<td>The default port to access a MySQL or Aurora database, for example, on an Amazon RDS instance</td>
</tr>
<tr>
<td>TCP</td>
<td>6</td>
<td>5439 (Redshift)</td>
<td>The default port to access an Amazon Redshift cluster database.</td>
</tr>
<tr>
<td>TCP</td>
<td>6</td>
<td>5432 (PostgreSQL)</td>
<td>The default port to access a PostgreSQL database, for example, on an Amazon RDS instance</td>
</tr>
<tr>
<td>TCP</td>
<td>6</td>
<td>1521 (Oracle)</td>
<td>The default port to access an Oracle database, for example, on an Amazon RDS instance</td>
</tr>
</tbody>
</table>

You can optionally restrict outbound traffic from your database servers. For example, you might want to allow access to the internet for software updates, but restrict all other kinds of traffic. You must first remove the default outbound rule that allows all outbound traffic.

<table>
<thead>
<tr>
<th>Protocol type</th>
<th>Protocol number</th>
<th>Port</th>
<th>Destination IP</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>6</td>
<td>80 (HTTP)</td>
<td>0.0.0.0/0</td>
<td>Allows outbound HTTP access to any IPv4 address</td>
</tr>
<tr>
<td>TCP</td>
<td>6</td>
<td>443 (HTTPS)</td>
<td>0.0.0.0/0</td>
<td>Allows outbound HTTPS access to any IPv4 address</td>
</tr>
<tr>
<td>TCP</td>
<td>6</td>
<td>80 (HTTP)</td>
<td>::/0</td>
<td>(IPv6-enabled VPC only) Allows outbound HTTP access to any IPv6 address</td>
</tr>
<tr>
<td>TCP</td>
<td>6</td>
<td>443 (HTTPS)</td>
<td>::/0</td>
<td>(IPv6-enabled VPC only) Allows outbound HTTPS access to any IPv6 address</td>
</tr>
</tbody>
</table>

### 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).
Security group rules for different use cases

<table>
<thead>
<tr>
<th>Protocol type</th>
<th>Protocol number</th>
<th>Port</th>
<th>Source IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>6</td>
<td>22 (SSH)</td>
<td>The public IPv4 address of your computer, or a range of IP addresses (in CIDR block notation) in your local network. If your VPC is enabled for IPv6 and your instance has an IPv6 address, you can enter an IPv6 address or range.</td>
</tr>
<tr>
<td>TCP</td>
<td>6</td>
<td>3389 (RDP)</td>
<td>The public IPv4 address of your computer, or a range of IP addresses (in CIDR block notation) in your local network. If your VPC is enabled for IPv6 and your instance has an IPv6 address, you can enter an IPv6 address or range.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Protocol type</th>
<th>Protocol number</th>
<th>Ports</th>
<th>Source IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1 (All)</td>
<td>-1 (All)</td>
<td>-1 (All)</td>
<td>The ID of the security group</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Protocol type</th>
<th>Protocol number</th>
<th>ICMP type</th>
<th>ICMP code</th>
<th>Source IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMP</td>
<td>1</td>
<td>8 (Echo)</td>
<td>N/A</td>
<td>The public IPv4 address of your computer, or a range of IPv4 addresses (in CIDR block notation) in your local network</td>
</tr>
</tbody>
</table>
To use the `ping6` command to ping the IPv6 address for your instance, you must add the following inbound ICMPv6 rule.

<table>
<thead>
<tr>
<th>Protocol type</th>
<th>Protocol number</th>
<th>ICMP type</th>
<th>ICMP code</th>
<th>Source IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMPv6</td>
<td>58</td>
<td>128 (Echo)</td>
<td>0</td>
<td>The IPv6 address of your computer, or a range of IPv6 addresses (in CIDR block notation) in your local network</td>
</tr>
</tbody>
</table>

**DNS server rules**

If you've set up your EC2 instance as a DNS server, you must ensure that TCP and UDP traffic can reach your DNS server over port 53.

For the source IP, specify one of the following:

- An IP address or range of IP addresses (in CIDR block notation) in a network
- The ID of a security group for the set of instances in your network that require access to the DNS server

<table>
<thead>
<tr>
<th>Protocol type</th>
<th>Protocol number</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>6</td>
<td>53</td>
</tr>
<tr>
<td>UDP</td>
<td>17</td>
<td>53</td>
</tr>
</tbody>
</table>

**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.
Security group rules for different use cases

<table>
<thead>
<tr>
<th>Protocol type</th>
<th>Protocol number</th>
<th>Ports</th>
<th>Source IP</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>6</td>
<td>22 (SSH)</td>
<td>The IP address range of your local computer, or the range of IP addresses (in CIDR block notation) for your network.</td>
<td>Allows inbound SSH access from your local computer.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Inbound</th>
<th>Protocol type</th>
<th>Protocol number</th>
<th>Port</th>
<th>Source IP</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TCP</td>
<td>6</td>
<td>The listener port</td>
<td>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</td>
<td>Allow inbound traffic on the load balancer listener port.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outbound</th>
<th>Protocol type</th>
<th>Protocol number</th>
<th>Port</th>
<th>Destination IP</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>6</td>
<td>The instance listener port</td>
<td>The ID of the instance security group</td>
<td>Allow outbound traffic to instances on the instance listener port.</td>
<td></td>
</tr>
<tr>
<td>TCP</td>
<td>6</td>
<td>The health check port</td>
<td>The ID of the instance security group</td>
<td>Allow outbound traffic to instances on the health check port.</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Protocol type</th>
<th>Protocol number</th>
<th>Port</th>
<th>Source IP</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>6</td>
<td>The instance listener port</td>
<td>The ID of the load balancer security group</td>
<td>Allow traffic from the load balancer on the instance listener port.</td>
</tr>
</tbody>
</table>
TCP | 6 | The health check port | The ID of the load balancer security group | Allow traffic from the load balancer on the health check port.

For more information, see Configure security groups for your Classic Load Balancer in the User Guide for Classic Load Balancers, and Security groups for your Application Load Balancer in the User Guide for Application Load Balancers.

### VPC peering rules

You can update the inbound or outbound rules for your VPC security groups to reference security groups in the peered VPC. Doing so allows traffic to flow to and from instances that are associated with the referenced security group in the peered VPC. For more information about how to configure security groups for VPC peering, see Updating your security groups to reference peer VPC groups.

### Configuration management in Amazon EC2

Amazon Machine Images (AMIs) provide an initial configuration for an Amazon EC2 instance, which includes the Windows OS and optional customer-specific customizations, such as applications and security controls. Create an AMI catalog containing customized security configuration baselines to ensure all Windows instances are launched with standard security controls. Security baselines can be baked into an AMI, bootstrapped dynamically when an EC2 instance is launched, or packaged as a product for uniform distribution through AWS Service Catalog portfolios. For more information on securing an AMI, see Best Practices for Building an AMI.

Each Amazon EC2 instance should adhere to organizational security standards. Do not install any Windows roles and features that are not required, and do install software to protect against malicious code (antivirus, antimalware, exploit mitigation), monitor host-integrity, and perform intrusion detection. Configure security software to monitor and maintain OS security settings, protect the integrity of critical OS files, and alert on deviations from the security baseline. Consider implementing recommended security configuration benchmarks published by Microsoft, the Center for Internet Security (CIS), or the National Institute of Standards and Technology (NIST). Consider using other Microsoft tools for particular application servers, such as the Best Practice Analyzer for SQL Server.

AWS customers can also run Amazon Inspector assessments to improve the security and compliance of applications deployed on Amazon EC2 instances. Amazon Inspector automatically assesses applications for vulnerabilities or deviations from best practices and includes a knowledge base of hundreds of rules mapped to common security compliance standards (for example, PCI DSS) and vulnerability definitions. Examples of built-in rules include checking if remote root login is enabled, or if vulnerable software versions are installed. These rules are regularly updated by AWS security researchers.

### Update management in Amazon EC2

We recommend that you regularly patch, update, and secure the operating system and applications on your EC2 instances. You can use AWS Systems Manager Patch Manager to automate the process of installing security-related updates for both the operating system and applications. Alternatively, you can use any automatic update services or recommended processes for installing updates that are provided by the application vendor.

You should configure Windows Update on your Amazon EC2 instances running Windows Server. By default, you will not receive Windows updates on AWS-provided AMIs. For a list of the latest Amazon EC2 AMIs running Windows Server, see Details About AWS Windows AMI Versions.
Change management in Amazon EC2

After initial security baselines are applied to Amazon EC2 instances at launch, control ongoing Amazon EC2 changes to maintain the security of your virtual machines. Establish a change management process to authorize and incorporate changes to AWS resources (such as security groups, route tables, and network ACLs) as well as to OS and application configurations (such as Windows or application patching, software upgrades, or configuration file updates).

AWS provides several tools to help manage changes to AWS resources, including AWS CloudTrail, AWS Config, AWS CloudFormation, and AWS Elastic Beanstalk, AWS OpsWorks, and management packs for Systems Center Operations Manager and System Center Virtual Machine Manager. Note that Microsoft releases Windows patches every Tuesday (sometimes even daily) and AWS updates all AWS-managed Windows AMIs within five days after Microsoft releases a patch. Therefore it is important to continually patch all baseline AMIs, update AWS CloudFormation templates and Auto Scaling group configurations with the latest AMI IDs, and implement tools to automate running instance patch management.

Microsoft provides several options for managing Windows OS and application changes. SCCM, for example, provides full lifecycle coverage of environment modifications. Select tools that address business requirements and control how changes will affect application SLAs, capacity, security, and disaster recovery procedures. Avoid manual changes and instead leverage automated configuration management software or command line tools such as the EC2 Run Command or Windows PowerShell to implement scripted, repeatable change processes. To assist with this requirement, use bastion hosts with enhanced logging for all interactions with your Windows instances to ensure that all events and tasks are automatically recorded.

Compliance validation for Amazon EC2

Third-party auditors assess the security and compliance of AWS services as part of multiple AWS compliance programs, such as SOC, PCI, FedRAMP, and HIPAA.

To learn whether Amazon Elastic Compute Cloud or other AWS services are in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- **Security and Compliance Quick Start Guides** – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- **Architecting for HIPAA Security and Compliance Whitepaper** – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.

**Note**
Not all services are compliant with HIPAA.

- **AWS Compliance Resources** – This collection of workbooks and guides might apply to your industry and location.
- **Evaluating Resources with Rules** in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
Audit and accountability in Amazon EC2

AWS CloudTrail, AWS Config, and AWS Config Rules provide audit and change tracking features for auditing AWS resource changes. Configure Windows event logs to send local log files to a centralized log management system to preserve log data for security and operational behavior analysis. Microsoft System Center Operations Manager (SCOM) aggregates information about Microsoft applications deployed to Windows instances and applies preconfigured and custom rulesets based on application roles and services. System Center Management Packs build on SCOM to provide application-specific monitoring and configuration guidance. These Management Packs support Windows Server Active Directory, SharePoint Server 2013, Exchange Server 2013, Lync Server 2013, SQL Server 2014, and many more servers and technologies. The AWS Management Pack for Microsoft System Center Operations Manager (SCOM) and the Systems Manager for Microsoft System Center Virtual Machine Manager (SCVMM) integrate with Microsoft Systems Center to help you monitor and manage your on-premises and AWS environments together.

In addition to Microsoft systems management tools, customers can use Amazon CloudWatch to monitor instance CPU utilization, disk performance, network I/O, and perform host and instance status checks. The EC2Config and EC2Launch services provide access to additional, advanced features for Windows instances. For example, they can export Windows system, security, application, and Internet Information Services (IIS) logs to CloudWatch Logs which can then be integrated with Amazon CloudWatch metrics and alarms. Customers can also create scripts that export Windows performance counters to Amazon CloudWatch custom metrics.
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. 1159)
- Amazon EC2 instance store (p. 1392)
- Use Amazon S3 with Amazon EC2 (p. 1405)

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. 1327).

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. 1159).

**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, hibernate, or terminate an instance, any data on instance store volumes is lost. For more information, see Amazon EC2 instance store (p. 1392).

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 Use Amazon S3 with Amazon EC2 (p. 1405).

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 mappings (p. 1413).

You can also attach EBS volumes to a running instance. For more information, see Attach an Amazon EBS volume to an instance (p. 1186).

Storage pricing

For information about storage pricing, open AWS Pricing, scroll down to Services Pricing, choose Storage, and then choose the storage option to open that storage option's pricing page. For information about estimating the cost of storage, see the AWS Pricing Calculator.

Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances. EBS volumes behave like raw, unformatted block devices. You can mount these volumes as devices on your instances. EBS volumes that are attached to an instance are exposed as storage volumes that persist independently from the life of the instance. You can create a file system on top of these volumes, or use them in any way you would use a block device (such as a hard drive). You can dynamically change the configuration of a volume attached to an instance.

We recommend Amazon EBS for data that must be quickly accessible and requires long-term persistence. EBS volumes are particularly well-suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage. Amazon EBS is well suited to both database-style applications that rely on random reads and writes, and to throughput-intensive applications that perform long, continuous reads and writes.

With Amazon EBS, you pay only for what you use. For more information about Amazon EBS pricing, see the Projecting Costs section of the Amazon Elastic Block Store page.

Contents

- Features of Amazon EBS (p. 1160)
- Amazon EBS volumes (p. 1160)
- Amazon EBS snapshots (p. 1207)
- Amazon Data Lifecycle Manager (p. 1272)
- Amazon EBS data services (p. 1315)
- Amazon EBS and NVMe on Windows instances (p. 1343)
- Amazon EBS–optimized instances (p. 1344)
Features of Amazon EBS

- You create an EBS volume in a specific Availability Zone, and then attach it to an instance in that same Availability Zone. To make a volume available outside of the Availability Zone, you can create a snapshot and restore that snapshot to a new volume anywhere in that Region. You can copy snapshots to other Regions and then restore them to new volumes there, making it easier to leverage multiple AWS Regions for geographical expansion, data center migration, and disaster recovery.

- Amazon EBS provides the following volume types: General Purpose SSD, Provisioned IOPS SSD, Throughput Optimized HDD, and Cold HDD. For more information, see EBS volume types (p. 1163).

The following is a summary of performance and use cases for each volume type.

- General Purpose SSD volumes (gp2 and gp3) balance price and performance for a wide variety of transactional workloads. These volumes are ideal for use cases such as boot volumes, medium-size single instance databases, and development and test environments.

- Provisioned IOPS SSD volumes (io1 and io2) are designed to meet the needs of I/O-intensive workloads that are sensitive to storage performance and consistency. They provide a consistent IOPS rate that you specify when you create the volume. This enables you to predictably scale to tens of thousands of IOPS per instance. Additionally, io2 volumes provide the highest levels of volume durability.

- Throughput Optimized HDD volumes (st1) provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. These volumes are ideal for large, sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing.

- Cold HDD volumes (sc1) provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. These volumes are ideal for large, sequential, cold-data workloads. If you require infrequent access to your data and are looking to save costs, these volumes provides inexpensive block storage.

- You can create your EBS volumes as encrypted volumes, in order to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. When you create an encrypted EBS volume and attach it to a supported instance type, data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted. The encryption occurs on the servers that host EC2 instances, providing encryption of data-in-transit from EC2 instances to EBS storage. For more information, see Amazon EBS encryption (p. 1327).

- 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. 1207).

- 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. 1361).

Amazon EBS volumes

An Amazon EBS volume is a durable, block-level storage device that you can attach to your instances. After you attach a volume to an instance, you can use it as you would use a physical hard drive. EBS
EBS volumes are flexible. For current-generation volumes attached to current-generation instance types, you can dynamically increase size, modify the provisioned IOPS capacity, and change volume type on live production volumes.

You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. You can also use them for throughput-intensive applications that perform continuous disk scans. EBS volumes persist independently from the running life of an EC2 instance.

You can attach multiple EBS volumes to a single instance. The volume and instance must be in the same Availability Zone.

Amazon EBS provides the following volume types: General Purpose SSD (gp2 and gp3), Provisioned IOPS SSD (io1 and io2), Throughput Optimized HDD (st1), Cold HDD (sc1), and Magnetic (standard). 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. 1163).

Your account has a limit on the number of EBS volumes that you can use, and the total storage available to you. For more information about these limits, and how to request an increase in your limits, see Amazon EC2 service quotas (p. 1463).

For more information about pricing, see Amazon EBS Pricing.

Benefits of using EBS volumes

EBS volumes provide benefits that are not provided by instance store volumes.

Data availability

When you create an EBS volume, it is automatically replicated within its Availability Zone to prevent data loss due to failure of any single hardware component. You can attach an EBS volume to any EC2 instance in the same Availability Zone. After you attach a volume, it appears as a native block device similar to a hard drive or other physical device. At that point, the instance can interact with the volume just as it would with a local drive. You can connect to the instance and format the EBS volume with a file system, such as NTFS, and then install applications.

If you attach multiple volumes to a device that you have named, you can stripe data across the volumes for increased I/O and throughput performance.

You can get monitoring data for your EBS volumes, including root device volumes for EBS-backed instances, at no additional charge. For more information about monitoring metrics, see Amazon EBS monitoring.
CloudWatch metrics for Amazon EBS (p. 1375). For information about tracking the status of your volumes, see Amazon CloudWatch Events for Amazon EBS (p. 1381).

Data persistence

An EBS volume is off-instance storage that can persist independently from the life of an instance. You continue to pay for the volume usage as long as the data persists.

EBS volumes that are attached to a running instance can automatically detach from the instance with their data intact when the instance is terminated if you uncheck the Delete on Termination check box when you configure EBS volumes for your instance on the EC2 console. The volume can then be reattached to a new instance, enabling quick recovery. If the check box for Delete on Termination is checked, the volume(s) will delete upon termination of the EC2 instance. If you are using an EBS-backed instance, you can stop and restart that instance without affecting the data stored in the attached volume. The volume remains attached throughout the stop-start cycle. This enables you to process and store the data on your volume indefinitely, only using the processing and storage resources when required. The data persists on the volume until the volume is deleted explicitly. The physical block storage used by deleted EBS volumes is overwritten with zeroes before it is allocated to another account. If you are dealing with sensitive data, you should consider encrypting your data manually or storing the data on a volume protected by Amazon EBS encryption. For more information, see Amazon EBS encryption (p. 1327).

By default, the root EBS volume that is created and attached to an instance at launch is deleted when that instance is terminated. You can modify this behavior by changing the value of the flag DeleteOnTermination to false when you launch the instance. This modified value causes the volume to persist even after the instance is terminated, and enables you to attach the volume to another instance.

By default, additional EBS volumes that are created and attached to an instance at launch are not deleted when that instance is terminated. You can modify this behavior by changing the value of the flag DeleteOnTermination to true when you launch the instance. This modified value causes the volumes to be deleted when the instance is terminated.

Data encryption

For simplified data encryption, you can create encrypted EBS volumes with the Amazon EBS encryption feature. All EBS volume types support encryption. You can use encrypted EBS volumes to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications. Amazon EBS encryption uses 256-bit Advanced Encryption Standard algorithms (AES-256) and an Amazon-managed key infrastructure. The encryption occurs on the server that hosts the EC2 instance, providing encryption of data-in-transit from the EC2 instance to Amazon EBS storage. For more information, see Amazon EBS encryption (p. 1327).

Amazon EBS encryption uses AWS Key Management Service (AWS KMS) master keys when creating encrypted volumes and any snapshots created from your encrypted volumes. The first time you create an encrypted EBS volume in a region, a default master key is created for you automatically. This key is used for Amazon EBS encryption unless you select a customer master key (CMK) that you created separately using AWS KMS. Creating your own CMK gives you more flexibility, including the ability to create, rotate, disable, define access controls, and audit the encryption keys used to protect your data. For more information, see the AWS Key Management Service Developer Guide.

Snapshots

Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon S3, where it is stored redundantly in multiple Availability Zones. The volume does not need to be attached to a running instance in order to take a snapshot. As you continue to write data to a volume, you can periodically create a snapshot of the volume to use as a baseline for
new volumes. These snapshots can be used to create multiple new EBS volumes or move volumes across Availability Zones. Snapshots of encrypted EBS volumes are automatically encrypted.

When you create a new volume from a snapshot, it's an exact copy of the original volume at the time the snapshot was taken. EBS volumes that are created from encrypted snapshots are automatically encrypted. By optionally specifying a different Availability Zone, you can use this functionality to create a duplicate volume in that zone. The snapshots can be shared with specific AWS accounts or made public. When you create snapshots, you incur charges in Amazon S3 based on the volume's total size. For a successive snapshot of the volume, you are only charged for any additional data beyond the volume's original size.

Snapshots are incremental backups, meaning that only the blocks on the volume that have changed after your most recent snapshot are saved. If you have a volume with 100 GiB of data, but only 5 GiB of data have changed since your last snapshot, only the 5 GiB of modified data is written to Amazon S3. Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot.

To help categorize and manage your volumes and snapshots, you can tag them with metadata of your choice. For more information, see Tag your Amazon EC2 resources (p. 1450).

To back up your volumes automatically, you can use Amazon Data Lifecycle Manager (p. 1272) or AWS Backup.

**Flexibility**

EBS volumes support live configuration changes while in production. You can modify volume type, volume size, and IOPS capacity without service interruptions. For more information, see Amazon EBS Elastic Volumes (p. 1315).

**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 volumes types fall into these categories:

- **Solid state drives (SSD) (p. 1163)** — Optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS.
- **Hard disk drives (HDD) (p. 1165)** — Optimized for large streaming workloads where the dominant performance attribute is throughput.
- **Previous generation (p. 1165)** — Hard disk drives that can be used for workloads with small datasets where data is accessed infrequently and performance is not of primary importance. We recommend that you consider a current generation volume type instead.

There are several factors that can affect the performance of EBS volumes, such as instance configuration, I/O characteristics, and workload demand. To fully use the IOPS provisioned on an EBS volume, use EBS-optimized instances (p. 1344). For more information about getting the most out of your EBS volumes, see Amazon EBS volume performance on Windows instances (p. 1361).

For more information about pricing, see Amazon EBS Pricing.

**Solid state drives (SSD)**

The SSD-backed volumes provided by Amazon EBS fall into these categories:

- **General Purpose SSD** — Provides a balance of price and performance. We recommend these volumes for most workloads.
- Provisioned IOPS SSD — Provides high performance for mission-critical, low-latency, or high-throughput workloads.

The following is a summary of the use cases and characteristics of SSD-backed volumes. For information about the maximum IOPS and throughput per instance, see Amazon EBS–optimized instances (p. 1344).

<table>
<thead>
<tr>
<th></th>
<th>General Purpose SSD</th>
<th>Provisioned IOPS SSD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Volume type</strong></td>
<td>gp3</td>
<td>gp2</td>
</tr>
<tr>
<td></td>
<td>io2 Block Express ‡</td>
<td>io2</td>
</tr>
<tr>
<td></td>
<td>io1</td>
<td></td>
</tr>
<tr>
<td><strong>Durability</strong></td>
<td>99.8% - 99.9%</td>
<td>99.999%</td>
</tr>
<tr>
<td></td>
<td>durability (0.1%</td>
<td>durability (0.001%</td>
</tr>
<tr>
<td></td>
<td>- 0.2% annual</td>
<td>annual failure rate)</td>
</tr>
<tr>
<td></td>
<td>failure rate)</td>
<td>failure rate)</td>
</tr>
<tr>
<td><strong>Use cases</strong></td>
<td>Low-latency interactive apps</td>
<td>Workloads that require:</td>
</tr>
<tr>
<td></td>
<td>Development and test environments</td>
<td>• Sub-millisecond latency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Sustained IOPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>performance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• More than 64,000 IOPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or 1,000 MiB/s of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>throughput</td>
</tr>
<tr>
<td><strong>Volume size</strong></td>
<td>1 GiB - 16 TiB</td>
<td>4 GiB - 64 TiB</td>
</tr>
<tr>
<td></td>
<td>4 GiB - 16 TiB</td>
<td></td>
</tr>
<tr>
<td><strong>Max IOPS per volume</strong></td>
<td>16,000</td>
<td>256,000</td>
</tr>
<tr>
<td>(16 KiB I/O)</td>
<td></td>
<td>64,000 †</td>
</tr>
<tr>
<td><strong>Max throughput per volume</strong></td>
<td>1,000 MiB/s</td>
<td>250 MiB/s *</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4,000 MiB/s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,000 MiB/s †</td>
</tr>
<tr>
<td><strong>Amazon EBS Multi-attach</strong></td>
<td>Not supported</td>
<td>Supported</td>
</tr>
<tr>
<td><strong>Boot volume</strong></td>
<td>Supported</td>
<td></td>
</tr>
</tbody>
</table>

* The throughput limit is between 128 MiB/s and 250 MiB/s, depending on the volume size. Volumes smaller than or equal to 170 GiB deliver a maximum throughput of 128 MiB/s. Volumes larger than 170 GiB but smaller than 334 GiB deliver a maximum throughput of 250 MiB/s if burst credits are available.
Volumes larger than or equal to 334 GiB deliver 250 MiB/s regardless of burst credits. gp2 volumes that were created before December 3, 2018 and that have not been modified since creation might not reach full performance unless you modify the volume (p. 1315).

† Maximum IOPS and throughput are guaranteed only on Instances built on the Nitro System (p. 146) provisioned with more than 32,000 IOPS. Other instances guarantee up to 32,000 IOPS and 500 MiB/s. io1 volumes that were created before December 6, 2017 and that have not been modified since creation might not reach full performance unless you modify the volume (p. 1315).

‡ io2 Block Express volumes are supported with R5b instances only. io2 volumes attached to an R5b instance during or after launch automatically run on Block Express. For more information, see io2 Block Express volumes (p. 1172).

Hard disk drives (HDD)

The HDD-backed volumes provided by Amazon EBS fall into these categories:

- Throughput Optimized HDD — A low-cost HDD designed for frequently accessed, throughput-intensive workloads.
- Cold HDD — The lowest-cost HDD design for less frequently accessed workloads.

The following is a summary of the use cases and characteristics of HDD-backed volumes. For information about the maximum IOPS and throughput per instance, see Amazon EBS–optimized instances (p. 1344).

<table>
<thead>
<tr>
<th></th>
<th>Throughput Optimized HDD</th>
<th>Cold HDD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Volume type</strong></td>
<td>st1</td>
<td>sc1</td>
</tr>
<tr>
<td><strong>Durability</strong></td>
<td>99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)</td>
<td>99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)</td>
</tr>
<tr>
<td><strong>Use cases</strong></td>
<td>• Big data</td>
<td>• Throughput-oriented storage for data that is infrequently accessed</td>
</tr>
<tr>
<td></td>
<td>• Data warehouses</td>
<td>• Scenarios where the lowest storage cost is important</td>
</tr>
<tr>
<td></td>
<td>• Log processing</td>
<td></td>
</tr>
<tr>
<td><strong>Volume size</strong></td>
<td>125 GiB - 16 TiB</td>
<td>125 GiB - 16 TiB</td>
</tr>
<tr>
<td><strong>Max IOPS per volume (1 MiB I/O)</strong></td>
<td>500</td>
<td>250</td>
</tr>
<tr>
<td><strong>Max throughput per volume</strong></td>
<td>500 MiB/s</td>
<td>250 MiB/s</td>
</tr>
<tr>
<td><strong>Amazon EBS Multi-attach</strong></td>
<td>Not supported</td>
<td>Not supported</td>
</tr>
<tr>
<td><strong>Boot volume</strong></td>
<td>Not supported</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

Previous generation volume types

The following table describes previous-generation EBS volume types. If you need higher performance or performance consistency than previous-generation volumes can provide, we recommend that you consider using General Purpose SSD (gp2 and gp3) or other current volume types. For more information, see Previous Generation Volumes.
General Purpose SSD volumes (gp3)

General Purpose SSD (gp3) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver a consistent baseline rate of 3,000 IOPS and 125 MiB/s, included with the price of storage. You can provision additional IOPS (up to 16,000) and throughput (up to 1,000 MiB/s) for an additional cost.

The maximum ratio of provisioned IOPS to provisioned volume size is 500 IOPS per GiB. The maximum ratio of provisioned throughput to provisioned IOPS is .25 MiB/s per IOPS. The following volume configurations support provisioning either maximum IOPS or maximum throughput:

- 32 GiB or larger: 500 IOPS/GiB x 32 GiB = 16,000 IOPS
- 8 GiB or larger and 4,000 IOPS or higher: 4,000 IOPS x 0.25 MiB/s/IOPS = 1,000 MiB/s

General Purpose SSD volumes (gp2)

General Purpose SSD (gp2) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. Between a minimum of 100 IOPS (at 33.33 GiB and below) and a maximum of 16,000 IOPS (at 5,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size. AWS designs gp2 volumes to deliver their provisioned performance 99% of the time. A gp2 volume can range in size from 1 GiB to 16 TiB.

I/O Credits and burst performance

The performance of gp2 volumes is tied to volume size, which determines the baseline performance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher baseline performance levels and accumulate I/O credits faster. I/O credits represent the available bandwidth that your gp2 volume can use to burst large amounts of I/O when more than the baseline performance is needed. The more credits your volume has for I/O, the more time it can burst beyond its baseline performance level and the better it performs when more performance is needed. The following diagram shows the burst-bucket behavior for gp2.
Each volume receives an initial I/O credit balance of 5.4 million I/O credits, which is enough to sustain the maximum burst performance of 3,000 IOPS for at least 30 minutes. This initial credit balance is designed to provide a fast initial boot cycle for boot volumes and to provide a good bootstrapping experience for other applications. Volumes earn I/O credits at the baseline performance rate of 3 IOPS per GiB of volume size. For example, a 100 GiB gp2 volume has a baseline performance of 300 IOPS.
When your volume requires more than the baseline performance I/O level, it draws on I/O credits in the credit balance to burst to the required performance level, up to a maximum of 3,000 IOPS. When your volume uses fewer I/O credits than it earns in a second, unused I/O credits are added to the I/O credit balance. The maximum I/O credit balance for a volume is equal to the initial credit balance (5.4 million I/O credits).

When the baseline performance of a volume is higher than maximum burst performance, I/O credits are never spent. If the volume is attached to an instance built on the Nitro System (p. 146), the burst balance is not reported. For other instances, the reported burst balance is 100%.

The burst duration of a volume is dependent on the size of the volume, the burst IOPS required, and the credit balance when the burst begins. This is shown in the following equation:

\[
\text{Burst duration} = \frac{\text{(Credit balance)}}{\text{(Burst IOPS)} - 3(\text{Volume size in GiB})}
\]

The following table lists several volume sizes and the associated baseline performance of the volume (which is also the rate at which it accumulates I/O credits), the burst duration at the 3,000 IOPS...
maximum (when starting with a full credit balance), and the time in seconds that the volume would take to refill an empty credit balance.

<table>
<thead>
<tr>
<th>Volume size (GiB)</th>
<th>Baseline performance (IOPS)</th>
<th>Burst duration when driving sustained 3,000 IOPS (second)</th>
<th>Seconds to fill empty credit balance when driving no IO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>1,802</td>
<td>54,000</td>
</tr>
<tr>
<td>100</td>
<td>300</td>
<td>2,000</td>
<td>18,000</td>
</tr>
<tr>
<td>250</td>
<td>750</td>
<td>2,400</td>
<td>36,000</td>
</tr>
<tr>
<td>334 (Min. size for max throughput)</td>
<td>1,002</td>
<td>2,703</td>
<td>5,389</td>
</tr>
<tr>
<td>500</td>
<td>1,500</td>
<td>3,600</td>
<td>3,600</td>
</tr>
<tr>
<td>750</td>
<td>2,250</td>
<td>7,200</td>
<td>2,400</td>
</tr>
<tr>
<td>1,000</td>
<td>3,000</td>
<td>N/A*</td>
<td>N/A*</td>
</tr>
<tr>
<td>5,334 (Min. size for max IOPS)</td>
<td>16,000</td>
<td>N/A*</td>
<td>N/A*</td>
</tr>
<tr>
<td>16,384 (16 TiB, max volume size)</td>
<td>16,000</td>
<td>N/A*</td>
<td>N/A*</td>
</tr>
</tbody>
</table>

* The baseline performance of the volume exceeds the maximum burst performance.

**What happens if I empty my I/O credit balance?**

If your gp2 volume uses all of its I/O credit balance, the maximum IOPS performance of the volume remains at the baseline IOPS performance level (the rate at which your volume earns credits) and the volume’s maximum throughput is reduced to the baseline IOPS multiplied by the maximum I/O size. Throughput can never exceed 250 MiB/s. When I/O demand drops below the baseline level and unused credits are added to the I/O credit balance, the maximum IOPS performance of the volume again exceeds the baseline. For example, a 100 GiB gp2 volume with an empty credit balance has a baseline performance of 300 IOPS and a throughput limit of 75 MiB/s (300 I/O operations per second * 256 KiB per I/O operation = 75 MiB/s). The larger a volume is, the greater the baseline performance is and the faster it replenishes the credit balance. For more information about how IOPS are measured, see I/O characteristics and monitoring (p. 1363).

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 switching to a gp3 volume.

For information about using CloudWatch metrics and alarms to monitor your burst bucket balance, see Monitor the burst bucket balance for volumes (p. 1181).

**Throughput performance**

Throughput for a gp2 volume can be calculated using the following formula, up to the throughput limit of 250 MiB/s:

\[
\text{Throughput in MiB/s} = \left(\frac{\text{Volume size in GiB}}{\text{IOPS per GiB}}\times\left(\frac{\text{I/O size in KiB}}{\text{I/O rate}}\right)\right)
\]

Assuming V = volume size, I = I/O size, R = I/O rate, and T = throughput, this can be simplified to:
The smallest volume size that achieves the maximum throughput is given by:

\[
\frac{T}{V} = \frac{250 \text{ MiB/s}}{256 \text{ KiB}(3 \text{ IOPS/GiB})}
\]

\[
= \frac{[(250)(2^{20})\text{(Bytes)}/\text{s}]}{(256)(2^{10})\text{(Bytes)}([3 \text{ IOP/s}]/[(2^{30})\text{(Bytes)}])}
\]

\[
= \frac{(250)(2^{20})(2^{30})\text{(Bytes)}}{(256)(2^{10})(3)}
\]

\[
= 357,913,941,333 \text{ Bytes}
\]

\[
= 333\# \text{ GiB} \text{ (334 GiB in practice because volumes are provisioned in whole gibibytes)}
\]

### Provisioned IOPS SSD volumes

Provisioned IOPS SSD (io1 and io2) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency. Provisioned IOPS SSD volumes use a consistent IOPS rate, which you specify when you create the volume, and Amazon EBS delivers the provisioned performance 99.9 percent of the time.

**io1** volumes are designed to provide 99.8 to 99.9 percent volume durability with an annual failure rate (AFR) no higher than 0.2 percent, which translates to a maximum of two volume failures per 1,000 running volumes over a one-year period. **io2** volumes are designed to provide 99.999 percent volume durability with an AFR no higher than 0.001 percent, which translates to a single volume failure per 100,000 running volumes over a one-year period.

Provisioned IOPS SSD **io1** and **io2** volumes are available for all Amazon EC2 instance types. Provisioned IOPS SSD **io2** volumes attached to R5b instances run on EBS Block Express. For more information, see **io2 Block Express volumes**.

### Considerations for **io2** volumes

- Keep the following in mind when **launching instances with** **io2** **volumes**:
  - If you launch an R5b instance with an **io2** volume, the volume automatically runs on Block Express (p. 1172), regardless of the volume's size and IOPS.
  - You can't launch an instance type that does not support Block Express (p. 1172) with an **io2** volume that has a size greater than 16 TiB or IOPS greater than 64,000.
  - You can't launch an R5b instance with an encrypted **io2** volume that has a size greater than 16 TiB or IOPS greater than 64,000 from an unencrypted AMI or a shared encrypted AMI. In this case, you must first create an encrypted AMI in your account and then use that AMI to launch the instance.
- Keep the following in mind when **creating** **io2** **volumes**:
  - If you create an **io2** volume with a size greater than 16 TiB or IOPS greater than 64,000 in a Region where Block Express (p. 1172) is supported, the volume automatically runs on Block Express.
You can't create an io2 volume with a size greater than 16 TiB or IOPS greater than 64,000 in a Region where Block Express (p. 1172) is not supported.

If you create an io2 volume with a size of 16 TiB or less and IOPS of 64,000 or less in a Region where Block Express (p. 1172) is supported, the volume does not run on Block Express.

You can't create an encrypted io2 volume that has a size greater than 16 TiB or IOPS greater than 64,000 from an unencrypted snapshot or a shared encrypted snapshot. In this case, you must first create an encrypted snapshot in your account and then use that snapshot to create the volume.

Keep the following in mind when attaching io2 volumes to instances:

If you attach an io2 volume to an R5b instance, the volume automatically runs on Block Express (p. 1172). It can take up to 48 hours to optimize the volume for Block Express. During this time, the volume provides io2 latency. After the volume has been optimized, it provides the sub-millisecond latency supported by Block Express.

You can't attach an io2 volume with a size greater than 16 TiB or IOPS greater than 64,000 to an instance type that does not support Block Express (p. 1172).

If you detach an io2 volume with a size of 16 TiB or less and IOPS of 64,000 or less from an R5b instance and attach it to an instance type that does not support Block Express (p. 1172), the volume no longer runs on Block Express and it provides io2 latency.

Keep the following in mind when modifying io2 volumes:

You can't modify an io2 volume and increase its size beyond 16 TiB or its IOPS beyond 64,000 while it is attached to an instance type that does not support Block Express (p. 1172).

You can’t modify the size or provisioned IOPS of an io2 volume that is attached to an R5b instance.

Performance

Provisioned IOPS SSD volumes can range in size from 4 GiB to 16 TiB and you can provision from 100 IOPS up to 64,000 IOPS per volume. You can achieve up to 64,000 IOPS only on Instances built on the Nitro System (p. 146). On other instance families you can achieve performance up to 32,000 IOPS. The maximum ratio of provisioned IOPS to requested volume size (in GiB) is 50:1 for io1 volumes, and 500:1 for io2 volumes. For example, a 100 GiB io1 volume can be provisioned with up to 5,000 IOPS, while a 100 GiB io2 volume can be provisioned with up to 50,000 IOPS. On a supported instance type, the following volume sizes allow provisioning up to the 64,000 IOPS maximum:

- io1 volume 1,280 GiB in size or greater (50 × 1,280 GiB = 64,000 IOPS)
- io2 volume 128 GiB in size or greater (500 × 128 GiB = 64,000 IOPS)

Provisioned IOPS SSD volumes provisioned with up to 32,000 IOPS support a maximum I/O size of 256 KiB and yield as much as 500 MiB/s of throughput. With the I/O size at the maximum, peak throughput is reached at 2,000 IOPS. Volumes provisioned with more than 32,000 IOPS (up to the maximum of 64,000 IOPS) yield a linear increase in throughput at a rate of 16 KiB per provisioned IOPS. For example, a volume provisioned with 48,000 IOPS can support up to 750 MiB/s of throughput (16 KiB per provisioned IOPS x 48,000 provisioned IOPS = 750 MiB/s). To achieve the maximum throughput of 1,000 MiB/s, a volume must be provisioned with 64,000 IOPS (16 KiB per provisioned IOPS x 64,000 provisioned IOPS = 1,000 MiB/s). The following graph illustrates these performance characteristics:
Your per-I/O latency experience depends on the provisioned IOPS and on your workload profile. For the best I/O latency experience, ensure that you provision IOPS to meet the I/O profile of your workload.

io2 Block Express volumes

Note

io2 Block Express volumes are supported with R5b instances only.

io2 Block Express volumes is the next generation of Amazon EBS storage server architecture. It has been built for the purpose of meeting the performance requirements of the most demanding I/O intensive applications that run on Nitro-based Amazon EC2 instances.

Block Express architecture increases performance and scale. Block Express servers communicate with Nitro-based instances using the Scalable Reliable Datagram (SRD) networking protocol. This interface is implemented in the Nitro Card dedicated for Amazon EBS I/O function on the host hardware of the instance. It minimizes I/O delay and latency variation (network jitter), which provides faster and more consistent performance for your applications. For more information, see io2 Block Express volumes.

io2 Block Express volumes are suited for workloads that benefit from a single volume that provides sub-millisecond latency, and supports higher IOPS, higher throughput, and larger capacity than io2 volumes.

io2 Block Express volumes support the same features as io2 volumes, including Multi-Attach, Elastic Volume operations, and encryption.

Topics

- Considerations (p. 1173)
- Performance (p. 1173)
Considerations

- **io2** Block Express volumes are currently supported with R5b instances only.
- **io2** Block Express volumes are currently available in all Regions where R5b instances are available, including us-east-1, us-east-2, us-west-2, ap-southeast-1, ap-northeast-1, and eu-central-1. R5b instance availability might vary by Availability Zone. For more information about R5b availability, see Find an Amazon EC2 instance type.
- **io2** Block Express volumes do not support fast snapshot restore. We recommend that you initialize these volumes to ensure that they deliver full performance. For more information, see Initialize Amazon EBS volumes (p. 1366).

Performance

With **io2** Block Express volumes, you can provision volumes with:

- Sub-millisecond average latency
- Storage capacity up to 64 TiB (65,536 GiB)
- Provisioned IOPS up to 256,000, with an IOPS:GiB ratio of 1,000:1. Maximum IOPS can be provisioned with volumes 256 GiB in size and larger (1,000 IOPS x 256 GiB = 256,000 IOPS).
- Volume throughput up to 4,000 MiB/s. Throughput scales proportionally up to 0.256 MiB/s per provisioned IOPS. Maximum throughput can be achieved at 16,000 IOPS or higher.

Quotas

**io2** Block Express volumes adhere to the same service quotas as **io2** volumes. For more information, see Amazon EBS quotas.
**Pricing and billing**

io2 volumes and io2 Block Express volumes are billed at the same rate. For more information, see Amazon EBS pricing.

Usage reports do not distinguish between io2 Block Express volumes and io2 volumes. We recommend that you use tags to help you identify costs associated with io2 Block Express volumes.

**Throughput Optimized HDD 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. 1180).

**Throughput credits and burst performance**

Like gp2, st1 uses a burst-bucket model for performance. Volume size determines the baseline throughput of your volume, which is the rate at which the volume accumulates throughput credits. Volume size also determines the burst throughput of your volume, which is the rate at which you can spend credits when they are available. Larger volumes have higher baseline and burst throughput. The more credits your volume has, the longer it can drive I/O at the burst level.

The following diagram shows the burst-bucket behavior for st1.

Subject to throughput and throughput-credit caps, the available throughput of an st1 volume is expressed by the following formula:

\[
(Volume \ size) \times (Credit \ accumulation \ rate \ per \ TiB) = Throughput
\]

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 0.125 TiB to 16 TiB, baseline throughput varies from 5 MiB/s to a cap of 500 MiB/s, which is reached at 12.5 TiB as follows:

\[
\frac{40 \text{ MiB/s}}{1 \text{ TiB}} \times 12.5 \text{ TiB} = 500 \text{ MiB/s}
\]

Burst throughput varies from 31 MiB/s to a cap of 500 MiB/s, which is reached at 2 TiB as follows:

\[
\frac{250 \text{ MiB/s}}{1 \text{ TiB}} \times 2 \text{ TiB} = 500 \text{ MiB/s}
\]

The following table states the full range of base and burst throughput values for st1:

<table>
<thead>
<tr>
<th>Volume size (TiB)</th>
<th>ST1 base throughput (MiB/s)</th>
<th>ST1 burst throughput (MiB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.125</td>
<td>5</td>
<td>31</td>
</tr>
<tr>
<td>0.5</td>
<td>20</td>
<td>125</td>
</tr>
<tr>
<td>1</td>
<td>40</td>
<td>250</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>500</td>
</tr>
<tr>
<td>3</td>
<td>120</td>
<td>500</td>
</tr>
<tr>
<td>4</td>
<td>160</td>
<td>500</td>
</tr>
<tr>
<td>5</td>
<td>200</td>
<td>500</td>
</tr>
<tr>
<td>6</td>
<td>240</td>
<td>500</td>
</tr>
<tr>
<td>7</td>
<td>280</td>
<td>500</td>
</tr>
<tr>
<td>8</td>
<td>320</td>
<td>500</td>
</tr>
<tr>
<td>9</td>
<td>360</td>
<td>500</td>
</tr>
<tr>
<td>10</td>
<td>400</td>
<td>500</td>
</tr>
<tr>
<td>11</td>
<td>440</td>
<td>500</td>
</tr>
<tr>
<td>12</td>
<td>480</td>
<td>500</td>
</tr>
<tr>
<td>12.5</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>13</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>14</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>15</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>16</td>
<td>500</td>
<td>500</td>
</tr>
</tbody>
</table>

The following diagram plots the table values:
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 Monitor the burst bucket balance for volumes (p. 1181).

Cold HDD volumes

Cold HDD (sc1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With a lower throughput limit than st1, sc1 is a good fit for large, sequential cold-data workloads. If you require infrequent access to your data and are looking to save costs, sc1 provides inexpensive block storage. Bootable sc1 volumes are not supported.

Cold HDD (sc1) volumes, though similar to Throughput Optimized HDD (st1) volumes, are designed to support infrequently accessed data.

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. 1180).

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.
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 0.125 TiB to 16 TiB, baseline throughput varies from 1.5 MiB/s to a maximum of 192 MiB/s, which is reached at 16 TiB as follows:

\[
\frac{12 \text{ MiB/s}}{16 \text{ TiB}} = 0.75 \text{ MiB/s/TiB}
\]

\[
\frac{192 \text{ MiB/s}}{1 \text{ TiB}} = 192 \text{ MiB/s/TiB}
\]

Burst throughput varies from 10 MiB/s to a cap of 250 MiB/s, which is reached at 3.125 TiB as follows:

\[
\frac{80 \text{ MiB/s}}{3.125 \text{ TiB}} = 25.6 \text{ MiB/s/TiB}
\]

\[
\frac{250 \text{ MiB/s}}{1 \text{ TiB}} = 250 \text{ MiB/s/TiB}
\]

The following table states the full range of base and burst throughput values for sc1:

<table>
<thead>
<tr>
<th>Volume Size (TiB)</th>
<th>SC1 Base Throughput (MiB/s)</th>
<th>SC1 Burst Throughput (MiB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.125</td>
<td>1.5</td>
<td>10</td>
</tr>
<tr>
<td>0.5</td>
<td>6</td>
<td>40</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>160</td>
</tr>
<tr>
<td>3</td>
<td>36</td>
<td>240</td>
</tr>
</tbody>
</table>
### EBS volumes

<table>
<thead>
<tr>
<th>Volume Size (TiB)</th>
<th>SC1 Base Throughput (MiB/s)</th>
<th>SC1 Burst Throughput (MiB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.125</td>
<td>37.5</td>
<td>250</td>
</tr>
<tr>
<td>4</td>
<td>48</td>
<td>250</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>250</td>
</tr>
<tr>
<td>6</td>
<td>72</td>
<td>250</td>
</tr>
<tr>
<td>7</td>
<td>84</td>
<td>250</td>
</tr>
<tr>
<td>8</td>
<td>96</td>
<td>250</td>
</tr>
<tr>
<td>9</td>
<td>108</td>
<td>250</td>
</tr>
<tr>
<td>10</td>
<td>120</td>
<td>250</td>
</tr>
<tr>
<td>11</td>
<td>132</td>
<td>250</td>
</tr>
<tr>
<td>12</td>
<td>144</td>
<td>250</td>
</tr>
<tr>
<td>13</td>
<td>156</td>
<td>250</td>
</tr>
<tr>
<td>14</td>
<td>168</td>
<td>250</td>
</tr>
<tr>
<td>15</td>
<td>180</td>
<td>250</td>
</tr>
<tr>
<td>16</td>
<td>192</td>
<td>250</td>
</tr>
</tbody>
</table>

The following diagram plots the table values:

![Diagram]

**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 Monitor the burst bucket balance for volumes (p. 1181).

**Magnetic volumes**

Magnetic volumes are backed by magnetic drives and are suited for workloads where data is accessed infrequently, and scenarios where low-cost storage for small volume sizes is important. These volumes deliver approximately 100 IOPS on average, with burst capability of up to hundreds of IOPS, and they can range in size from 1 GiB to 1 TiB.
Note
Magnetic is a previous generation volume type. For new applications, we recommend using one of the newer volume types. For more information, see Previous Generation Volumes.

For information about using CloudWatch metrics and alarms to monitor your burst bucket balance, see Monitor the burst bucket balance for volumes (p. 1181).

Performance considerations when using HDD volumes

For optimal throughput results using HDD volumes, plan your workloads with the following considerations in mind.

Comparing Throughput Optimized HDD and Cold HDD

The \textit{st1} and \textit{sc1} bucket sizes vary according to volume size, and a full bucket contains enough tokens for a full volume scan. However, larger \textit{st1} and \textit{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 \textit{st1} or \textit{sc1} throughput limits.

Both \textit{st1} and \textit{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.

In general, scan times are expressed by this formula:

\[
\text{Volume size} \div \text{Throughput} = \text{Scan time}
\]

For example, taking the performance consistency guarantees and other optimizations into account, an \textit{st1} customer with a 5-TiB volume can expect to complete a full volume scan in 2.91 to 3.27 hours.

- **Optimal scan time**

  \[
  \frac{5 \text{ TiB}}{500 \text{ MiB/s}} = \frac{5 \text{ TiB}}{0.00047684 \text{ TiB/s}} = 10,486 \text{ seconds} = 2.91 \text{ hours}
  \]

- **Maximum scan time**

  \[
  \frac{5.83 \text{ hours}}{0.90 \times 0.99} = 3.27 \text{ hours}
  \]

Similarly, an \textit{sc1} customer with a 5-TiB volume can expect to complete a full volume scan in 5.83 to 6.54 hours.

- **Optimal scan time**

  \[
  \frac{5 \text{ TiB}}{250 \text{ MiB/s}} = \frac{5 \text{ TiB}}{0.000238418 \text{ TiB/s}} = 20972 \text{ seconds} = 5.83 \text{ hours}
  \]

- **Maximum scan time**

  \[
  \frac{5.83 \text{ hours}}{0.90 \times 0.99} = 6.54 \text{ hours}
  \]
The following table shows ideal scan times for volumes of various size, assuming full buckets and sufficient instance throughput.

<table>
<thead>
<tr>
<th>Volume size (TiB)</th>
<th>ST1 scan time with burst (hours)*</th>
<th>SC1 scan time with burst (hours)*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.17</td>
<td>3.64</td>
</tr>
<tr>
<td>2</td>
<td>1.17</td>
<td>3.64</td>
</tr>
<tr>
<td>3</td>
<td>1.75</td>
<td>3.64</td>
</tr>
<tr>
<td>4</td>
<td>2.33</td>
<td>4.66</td>
</tr>
<tr>
<td>5</td>
<td>2.91</td>
<td>5.83</td>
</tr>
<tr>
<td>6</td>
<td>3.50</td>
<td>6.99</td>
</tr>
<tr>
<td>7</td>
<td>4.08</td>
<td>8.16</td>
</tr>
<tr>
<td>8</td>
<td>4.66</td>
<td>9.32</td>
</tr>
<tr>
<td>9</td>
<td>5.24</td>
<td>10.49</td>
</tr>
<tr>
<td>10</td>
<td>5.83</td>
<td>11.65</td>
</tr>
<tr>
<td>11</td>
<td>6.41</td>
<td>12.82</td>
</tr>
<tr>
<td>12</td>
<td>6.99</td>
<td>13.98</td>
</tr>
<tr>
<td>13</td>
<td>7.57</td>
<td>15.15</td>
</tr>
<tr>
<td>14</td>
<td>8.16</td>
<td>16.31</td>
</tr>
<tr>
<td>15</td>
<td>8.74</td>
<td>17.48</td>
</tr>
<tr>
<td>16</td>
<td>9.32</td>
<td>18.64</td>
</tr>
</tbody>
</table>

* These scan times assume an average queue depth (rounded to the nearest whole number) of four or more when performing 1 MiB of sequential I/O.

Therefore if you have a throughput-oriented workload that needs to complete scans quickly (up to 500 MiB/s), or requires several full volume scans a day, use st1. If you are optimizing for cost, your data is relatively infrequently accessed, and you don’t need more than 250 MiB/s of scanning performance, then use sc1.

Inefficiency of small read/writes on HDD

The performance model for st1 and sc1 volumes is optimized for sequential I/Os, favoring high-throughput workloads, offering acceptable performance on workloads with mixed IOPS and throughput, and discouraging workloads with small, random I/O.

For example, an I/O request of 1 MiB or less counts as a 1 MiB I/O credit. However, if the I/Os are sequential, they are merged into 1 MiB I/O blocks and count only as a 1 MiB I/O credit.

Limitations on per-instance throughput

Throughput for st1 and sc1 volumes is always determined by the smaller of the following:
- Throughput limits of the volume
- Throughput limits of the instance

As for all Amazon EBS volumes, we recommend that you select an appropriate EBS-optimized EC2 instance in order to avoid network bottlenecks. For more information, see Amazon EBS–optimized instances (p. 1344).

**Monitor the burst bucket balance for volumes**

You can monitor the burst-bucket level for gp2, st1, and sc1 volumes using the EBS BurstBalance metric available in Amazon CloudWatch. This metric shows the percentage of I/O credits (for gp2) or throughput credits (for st1 and sc1) remaining in the burst bucket. For more information about the BurstBalance metric and other metrics related to I/O, see I/O characteristics and monitoring (p. 1363). CloudWatch also allows you to set an alarm that notifies you when the BurstBalance value falls to a certain level. For more information, see Creating Amazon CloudWatch Alarms.

**Constraints on the size and configuration of an EBS volume**

The size of an Amazon EBS volume is constrained by the physics and arithmetic of block data storage, as well as by the implementation decisions of operating system (OS) and file system designers. AWS imposes additional limits on volume size to safeguard the reliability of its services.

The following sections describe the most important factors that limit the usable size of an EBS volume and offer recommendations for configuring your EBS volumes.

**Contents**
- Storage capacity (p. 1181)
- Service limitations (p. 1182)
- Partitioning schemes (p. 1182)
- Data block sizes (p. 1183)

**Storage capacity**

The following table summarizes the theoretical and implemented storage capacities for the most commonly used file systems on Amazon EBS, assuming a 4,096 byte block size.

<table>
<thead>
<tr>
<th>Partitioning scheme</th>
<th>Max addressable blocks</th>
<th>Theoretical max size (blocks × block size)</th>
<th>Ext4 implemented max size*</th>
<th>XFS implemented max size**</th>
<th>NTFS implemented max size</th>
<th>Max supported by EBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MBR</td>
<td>$2^{32}$</td>
<td>2 TiB</td>
<td>2 TiB</td>
<td>2 TiB</td>
<td>2 TiB</td>
<td>2 TiB</td>
</tr>
<tr>
<td>GPT</td>
<td>$2^{64}$</td>
<td>64 ZiB</td>
<td>1 EiB = $1024^2$ TiB (50 TiB certified on RHEL7)</td>
<td>500 TiB</td>
<td>256 TiB</td>
<td>64 TiB †</td>
</tr>
</tbody>
</table>

Service limitations

Amazon EBS abstracts the massively distributed storage of a data center into virtual hard disk drives. To an operating system installed on an EC2 instance, an attached EBS volume appears to be a physical hard disk drive containing 512-byte disk sectors. The OS manages the allocation of data blocks (or clusters) onto those virtual sectors through its storage management utilities. The allocation is in conformity with a volume partitioning scheme, such as master boot record (MBR) or GUID partition table (GPT), and within the capabilities of the installed file system (ext4, NTFS, and so on).

EBS is not aware of the data contained in its virtual disk sectors; it only ensures the integrity of the sectors. This means that AWS actions and OS actions are independent of each other. When you are selecting a volume size, be aware of the capabilities and limits of both, as in the following cases:

• EBS currently supports a maximum volume size of 64 TiB. This means that you can create an EBS volume as large as 64 TiB, but whether the OS recognizes all of that capacity depends on its own design characteristics and on how the volume is partitioned.

• Amazon EC2 requires Windows boot volumes to use MBR partitioning. As discussed in Partitioning schemes (p. 1182), this means that boot volumes cannot be larger than 2 TiB. Windows data volumes are not subject to this limitation and can use GPT partitioning. If a Windows boot volume that is 2 TiB or larger is converted to use a dynamic MBR partition table, you will see an error for the volume in Disk Manager.

• Windows non-boot volumes that are 2 TiB (2048 GiB) or larger must use a GPT partition table to access the entire volume. If an EBS volume over 2 TiB in size is attached to a Windows instance at launch, it is automatically formatted with a GPT partition table. If you attach an EBS volume over 2 TiB in size to a Windows instance after launch, you must initialize it with a GPT table manually. For more information, see Make an Amazon EBS volume available for use on Windows (p. 1187).

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. 1183). The common partitioning schemes in use are master boot record (MBR) and GUID partition table (GPT). The important differences between these schemes can be summarized as follows.

MBR

MBR uses a 32-bit data structure to store block addresses. This means that each data block is mapped with one of $2^{32}$ possible integers. The maximum addressable size of a volume is given by the following formula:

\[(2^{32} - 1) \times \text{Block size}\]

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 the following formula:

$$(2^{64} - 1) \times \text{Block size}$$

The block size for GPT volumes is commonly 4,096 bytes. Therefore:

$$(2^{64} - 1) \times 4,096 \text{ bytes}$$
$$= 2^{64} \times 4,096 \text{ bytes} - 1 \times 4,096 \text{ bytes}$$
$$= 2^{64} \times 2^{12} \text{ bytes} - 4,096 \text{ bytes}$$
$$= 2^{70} \text{ bytes} - 4,096 \text{ bytes}$$
$$= 64 \text{ ZiB} - 4,096 \text{ bytes}$$

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:

<table>
<thead>
<tr>
<th>Block size</th>
<th>Max volume size</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 KiB (default)</td>
<td>16 TiB</td>
</tr>
<tr>
<td>8 KiB</td>
<td>32 TiB</td>
</tr>
<tr>
<td>16 KiB</td>
<td>64 TiB</td>
</tr>
<tr>
<td>32 KiB</td>
<td>128 TiB</td>
</tr>
<tr>
<td>64 KiB (maximum)</td>
<td>256 TiB</td>
</tr>
</tbody>
</table>

The EBS-imposed limit on volume size (16 TiB) is currently equal to the maximum size enabled by 4-KiB data blocks.

Create an Amazon EBS volume

You can create an Amazon EBS volume and then attach it to any EC2 instance in the same Availability Zone. If you create an encrypted EBS volume, you can only attach it to supported instance types. For more information, see Supported instance types (p. 1329).

If you are creating a volume for a high-performance storage scenario, you should make sure to use a Provisioned IOPS SSD volume (io1 or io2) and attach it to an instance with enough bandwidth to
support your application, such as an EBS-optimized instance. The same advice holds for Throughput Optimized HDD (st1) and Cold HDD (sc1) volumes. For more information, see Amazon EBS–optimized instances (p. 1344).

**Note**
If you create a volume for use with a Windows instance, and it's larger than 2048 GiB (or is a volume that's smaller than 2048 GiB but might be increased later), ensure that you configure the volume to use GPT partition tables. For more information, see Windows support for hard disks that are larger than 2 TB.

Empty EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). However, storage blocks on volumes that were created from snapshots must be initialized (pulled down from Amazon S3 and written to the volume) before you can access the block. This preliminary action takes time and can cause a significant increase in the latency of an I/O operation the first time each block is accessed. Volume performance is achieved after all blocks have been downloaded and written to the volume. For most applications, amortizing this cost over the lifetime of the volume is acceptable. To avoid this initial performance hit in a production environment, you can force immediate initialization of the entire volume or enable fast snapshot restore. For more information, see Initialize Amazon EBS volumes (p. 1366).

**Important**
If you create an io2 volume with a size greater than 16 TiB or with IOPS greater than 64,000 in a Region where EBS Block Express is supported, the volume automatically runs on Block Express. io2 Block Express volumes can be attached to R5b instances only. For more information, see io2 Block Express volumes.

**Methods of creating a volume**

- Create and attach EBS volumes when you launch instances by specifying a block device mapping. For more information, see Launch an instance using the Launch Instance Wizard (p. 392) and Block device mappings (p. 1413).
- Create an empty EBS volume and attach it to a running instance. For more information, see Create an empty volume (p. 1184) below.
- Create an EBS volume from a previously created snapshot and attach it to a running instance. For more information, see Create a volume from a snapshot (p. 1185) below.

**Create an empty volume**

Empty volumes receive their maximum performance the moment that they are available and do not require initialization.

You can create an empty EBS volume using one of the following methods.

**Console**

To create an empty EBS volume using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region in which you would like to create your volume. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can't. For more information, see Resource locations (p. 1442).
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. 1163).
6. For Size, enter the size of the volume, in GiB. For more information, see Constraints on the size and configuration of an EBS volume (p. 1181).
7. For **IOPS**, enter the maximum number of input/output operations per second (IOPS) that the volume should provide. You can specify IOPS only for gp3, io1, and io2 volumes.

8. For **Throughput**, enter the throughput that the volume should provide, in MiB/s. You can specify throughput only for gp3 volumes.

9. For **Availability Zone**, choose the Availability Zone in which to create the volume. An EBS volume must be attached to an EC2 instance that is in the same Availability Zone as the volume.

10. (Optional) If the instance type supports EBS encryption and you want to encrypt the volume, select **Encrypt this volume** and choose a CMK. If encryption by default is enabled in this Region, EBS encryption is enabled and the default CMK for EBS encryption is chosen. You can choose a different CMK from **Master Key** or paste the full ARN of any key that you can access. For more information, see Amazon EBS encryption (p. 1327).

11. (Optional) Choose **Create additional tags** to add tags to the volume. For each tag, provide a tag key and a tag value. For more information, see Tag your Amazon EC2 resources (p. 1450).

12. Choose **Create Volume**. The volume is ready for use when the volume status is **Available**.

13. To use your new volume, attach it to an instance, format it, and mount it. For more information, see Attach an Amazon EBS volume to an instance (p. 1186).

**AWS CLI**

**To create an empty EBS volume using the command line**

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

- `create-volume` (AWS CLI)
- `New-EC2Volume` (AWS Tools for Windows PowerShell)

**Create a volume from a snapshot**

Volumes created from snapshots load lazily in the background. This means that there is no need to wait for all of the data to transfer from Amazon S3 to your EBS volume before the instance can start accessing an attached volume and all its data. If your instance accesses data that hasn't yet been loaded, the volume immediately downloads the requested data from Amazon S3, and then continues loading the rest of the volume data in the background. Volume performance is achieved after all blocks are downloaded and written to the volume. To avoid the initial performance hit in a production environment, see Initialize Amazon EBS volumes (p. 1366).

New EBS volumes that are created from encrypted snapshots are automatically encrypted. You can also encrypt a volume on-the-fly while restoring it from an unencrypted snapshot. Encrypted volumes can only be attached to instance types that support EBS encryption. For more information, see Supported instance types (p. 1329).

You can create a volume from a snapshot using one of the following methods.

**Console**

**To create an EBS volume from a snapshot using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. From the navigation bar, select the Region that your snapshot is located in.

   To use the snapshot to create a volume in a different region, copy your snapshot to the new Region and then use it to create a volume in that Region. For more information, see Copy an Amazon EBS snapshot (p. 1229).
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. 1163).

6. For **Snapshot ID**, start typing the ID or description of the snapshot from which you are restoring the volume, and choose it from the list of suggested options.

7. (Optional) Select **Encrypt this volume** to change the encryption state of your volume. This is optional if encryption by default (p. 1331) is enabled. Select a CMK from **Master Key** to specify a CMK other than the default CMK for EBS encryption.

8. For **Size**, verify that the default size of the snapshot meets your needs or enter the size of the volume, in GiB.

   If you specify both a volume size and a snapshot, the size must be equal to or greater than the snapshot size. When you select a volume type and a snapshot, the minimum and maximum sizes for the volume are shown next to **Size**. For more information, see Constraints on the size and configuration of an EBS volume (p. 1181).

9. For **IOPS**, enter the maximum number of input/output operations per second (IOPS) that the volume should provide. You can specify IOPS only for gp3, io1, and io2 volumes.

10. For **Throughput**, enter the throughput that the volume should provide, in MiB/s. You can specify throughput only for gp3 volumes.

11. For **Availability Zone**, choose the Availability Zone in which to create the volume. An EBS volume must be attached to an EC2 instance that is in the same Availability Zone as the volume.

12. (Optional) Choose **Create additional tags** to add tags to the volume. For each tag, provide a tag key and a tag value.

13. Choose **Create Volume**.

14. To use your new volume, attach it to an instance and mount it. For more information, see Attach an Amazon EBS volume to an instance (p. 1186).

15. If you created a volume that is larger than the snapshot, you must extend the file system on the volume to take advantage of the extra space. For more information, see Amazon EBS Elastic Volumes (p. 1315).

**AWS CLI**

To create an EBS volume from a snapshot using the command line

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

- `create-volume` (AWS CLI)
- `New-EC2Volume` (AWS Tools for Windows PowerShell)

**Attach an Amazon EBS volume to an instance**

You can attach an available EBS volume to one or more of your instances that is in the same Availability Zone as the volume.

For information about adding EBS volumes to your instance at launch, see Instance block device mapping (p. 1419).

**Prerequisites**

- Determine how many volumes you can attach to your instance. For more information, see Instance volume limits (p. 1407).
- 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. 1329).
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.

**Important**
If you attach an \io2\ volume to an R5b instance, the volume always runs on EBS Block Express. Currently, only R5b instances support \io2\ Block Express volumes. For more information, see \io2\ Block Express volumes.

You can attach a volume to an instance using one of the following methods.

**Console**

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 names on Windows instances (p. 1412).
6. Choose Attach.
7. Connect to your instance and mount the volume. For more information, see Make an Amazon EBS volume available for use on Windows (p. 1187).

**AWS CLI**

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 Access Amazon EC2 (p. 3).

- attach-volume (AWS CLI)
- Add-EC2Volume (AWS Tools for Windows PowerShell)

**Note**
In some situations, you may find that a volume other than the volume attached to /dev/xvda or /dev/sda has become the root volume of your instance. This can happen when you have attached the root volume of another instance, or a volume created from the snapshot of a root volume, to an instance with an existing root volume. For more information, see Boot from the wrong volume.

**Make an Amazon EBS volume available for use on Windows**

After you attach an Amazon EBS volume to your instance that runs on Xen hypervisor, 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.
On Nitro instances, the Amazon EBS volume is exposed as a block device when the NVMe controller scans the PCI bus. The disk does not appear as removable. Unlike Xen-based instances, there is only one NVMe controller per EBS volume on Nitro instances.

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. 1207).

You can get directions for volumes on a Linux instance from Make 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 Connect to your Windows instance (p. 413).

2. Start the Disk Management utility. On the taskbar, open the context (right-click) menu for the Windows logo and choose *Disk Management*.
   
   **Note**
   

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 Connect to your Windows instance (p. 413).

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.

   ```batch
   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 run the following command:

   C:\> diskpart /s diskpart.txt

**View information about an Amazon EBS volume**

You can view descriptive information about your EBS volumes. For example, you can view information about all volumes in a specific Region or view detailed information about a single volume, including its size, volume type, whether the volume is encrypted, which master key was used to encrypt the volume, and the specific instance to which the volume is attached.

You can get additional information about your EBS volumes, such as how much disk space is available, from the operating system on the instance.

**View volume information**

You can view information about a volume using one of the following methods.

**Console**

**To view information about an EBS volume using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Volumes**.
3. (Optional) Use the filter options in the search field to display only the volumes that interest you. For example, if you know the instance ID, choose **Instance ID** from the search field menu, and then choose the instance ID from the list provided. To remove a filter, choose it again.
4. Select the volume.
5. In the details pane, you can inspect the information provided about the volume. **Attachment information** shows the instance ID this volume is attached to and the device name under which it is attached.
6. (Optional) Choose the **Attachment information** link to view additional details about the instance.

**To view the EBS volumes that are attached to an instance using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance.
4. In the **Storage** tab, view the information provided about root and block devices.
5. (Optional) Choose a link in the **Volume ID** column to view additional details for the volume.

**AWS CLI**

**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 [Access Amazon EC2](p. 3).

- `describe-volumes` (AWS CLI)
- `Get-EC2Volume` (AWS Tools for Windows PowerShell)
Volume state

Volume state describes the availability of an Amazon EBS volume. You can view the volume state in the **State** column on the **Volumes** page in the console, or by using the **describe-volumes** AWS CLI command.

The possible volume states are:

- **creating**
  - The volume is being created.
- **available**
  - The volume is not attached to an instance.
- **in-use**
  - The volume is attached to an instance.
- **deleting**
  - The volume is being deleted.
- **deleted**
  - The volume is deleted.
- **error**
  - The underlying hardware related to your EBS volume has failed, and the data associated with the volume is unrecoverable. For information about how to restore the volume or recover the data on the volume, see *My EBS volume has a status of "error"*.

View volume metrics

You can get additional information about your EBS volumes from Amazon CloudWatch. For more information, see *Amazon CloudWatch metrics for Amazon EBS* (p. 1375).

View 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
```
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
```

For information about viewing free disk space on a Linux instance, see View free disk space in the Amazon EC2 User Guide for Linux Instances.

**Replace an Amazon EBS volume**

Amazon EBS snapshots are the preferred backup tool on Amazon EC2 because of their speed, convenience, and cost. When creating a volume from a snapshot, you recreate its state at a specific point in the past with all data intact. By attaching a volume created from a snapshot to an instance, you can duplicate data across Regions, create test environments, replace a damaged or corrupted production volume in its entirety, or retrieve specific files and directories and transfer them to another attached volume. For more information, see Amazon EBS snapshots (p. 1207).

The procedure for replacing a volume differs depending on whether the volume is the root volume or a data volume.

**Topics**

- Replace a root volume (p. 1193)
- Replace a data volume (p. 1196)

**Replace a root volume**

Amazon EC2 enables you to replace the root EBS volume for an instance without stopping the instance. You can restore the root volume for an instance to its launch state, or to a specific snapshot. This allows you to fix issues, such as root volume corruption or guest operating system network configuration errors, while retaining the following:

- Data stored on instance store volumes — Instance store volumes remain attached to the instance after the root volume has been replaced.
- Network configuration — All network interfaces remain attached to the instance and they retain their IP addresses, identifiers, and attachment IDs. When the instance becomes available, all pending network traffic is flushed. Additionally, the instance remains on the same physical host, so it retains its public and private IP addresses and DNS name.
- IAM policies — IAM profiles and policies (such as tag-based policies) that are associated with the instance are retained and enforced.

When you replace the root volume for an instance, a new volume is restored to the original volume's launch state, or using a specific snapshot. The original volume is detached from the instance, and the new volume is attached to the instance in its place. The original volume is not automatically deleted. If you no longer need it, you can delete it manually after the root volume replacement task completes. For more information about root volume replacement task states, see View root volume replacement tasks (p. 1195).

**Topics**

- Considerations (p. 239)
Replace a root volume (p. 1194)  
View root volume replacement tasks (p. 1195)

Considerations

- The instance is automatically rebooted when the root volume is replaced. The contents of the memory (RAM) is erased during the reboot.
- You can't replace the root volume if it is an instance store volume.
- You can't replace the root volume for metal instances.
- You can only use snapshots that belong to the same lineage as the instance's current root volume. You can't use snapshot copies created from snapshots that were taken from the root volume. Additionally, after successfully completing a root volume replacement task, snapshots taken from the previous root volume can't be used to create a root volume replacement task for the new volume.

Replace a root volume

When you replace the root volume for an instance, you can choose to restore the volume to its initial launch state, or you can choose to restore the volume to a specific snapshot. If you choose to restore the volume to a specific snapshot, then you must select a snapshot that was taken of that root volume. If you choose to restore the root volume to its initial launch state, the root volume is restored from the snapshot that was used to create the volume.

You can replace the root volume for an instance using one of the following methods. If you use the Amazon EC2 console, note that replacing the root volume is only available in the new console.

Amazon EC2 console

To replace the root volume

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance for which to replace the root volume and choose Actions, Monitor and troubleshoot, Replace root volume.
4. In the Replace root volume screen, do one of the following:
   - To restore the instance's root volume to its initial launch state, choose Create replacement task without selecting a snapshot.
   - To restore the instance's root volume to a specific snapshot, for Snapshot, select the snapshot to use, and then choose Create replacement task.

AWS CLI

To restore the root volume to the initial launch state

Use the `create-replace-root-volume-task` command. Specify the ID of the instance for which to replace the root volume and omit the --snapshot-id parameter.

```
$ aws ec2 create-replace-root-volume-task --instance-id instance_id
```

For example:

```
$ aws ec2 create-replace-root-volume-task --instance-id i-1234567890abcdef0
```

To restore the root volume to a specific snapshot
Use the `create-replace-root-volume-task` command. Specify the ID of the instance for which to replace the root volume and the ID of the snapshot to use.

```bash
# aws ec2 create-replace-root-volume-task --instance-id instance_id --snapshot-id snapshot_id
```

For example:

```bash
# aws ec2 create-replace-root-volume-task --instance-id i-1234567890abcdef0 --snapshot-id snap-9876543210abcdef0
```

**View root volume replacement tasks**

After you start a root volume replacement task, the task enters the following states:

- **pending** — the replacement volume is being created.
- **in-progress** — the original volume is being detached and the replacement volume is being attached.
- **succeeded** — the replacement volume has been successfully attached to the instance and the instance is available.
- **failing** — the replacement task is in the process of failing.
- **failed** — the replacement task has failed but the original root volume is still attached.
- **failing-detached** — the replacement task is in the process of failing. The instance might have no root volume attached.
- **failed-detached** — the replacement task has failed and the instance has no root volume attached.

You can view the root volume replacement tasks for an instance using one of the following methods.

**Amazon EC2 console**

To view the root volume replacement tasks

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance for which to view the root volume replacement tasks, and then choose the **Storage** tab.
4. In the **Storage** tab, expand **Recent root volume replacement tasks**.

**AWS CLI**

To view the status of a root volume replacement task

Use the `describe-replace-root-volume-tasks` command and specify the IDs of the root volume replacement tasks to view.

```bash
# aws ec2 describe-replace-root-volume-tasks --replace-root-volume-task-ids task_id_1 task_id_2
```

For example:

```bash
# aws ec2 describe-replace-root-volume-tasks --replace-root-volume-task-ids replacevol-1234567890abcdef0
```
Replace a data volume

You can use the following procedure to replace a (non-root) data volume with another volume created from a previous snapshot of that volume. You must detach the current volume and then attach the new volume.

Note that EBS volumes can only be attached to EC2 instances in the same Availability Zone.

Use the following method.

Console

To replace a data volume

1. Create a volume from the snapshot and write down the ID of the new volume. For more information, see Create a volume from a snapshot (p. 1185).

2. On the volumes page, select the check box for the volume to replace. On the Description tab, find Attachment information and write down the device name of the volume (for example, /dev/sda1) and the ID of the instance.

3. With the volume still selected, choose Actions, Detach Volume. When prompted for confirmation, choose Yes, Detach. Clear the check box for this volume.

4. Select the check box for the new volume that you created in step 1. Choose Actions, Attach Volume. Enter the instance ID and device name that you wrote down in step 2, and then choose Attach.

5. Connect to your instance and mount the volume. For more information, see Make an Amazon EBS volume available for use on Windows (p. 1187).

Monitor the status of your volumes

Amazon Web Services (AWS) automatically provides data that you can use to monitor your Amazon Elastic Block Store (Amazon EBS) volumes.
EBS volume status checks

Volume status checks enable you to better understand, track, and manage potential inconsistencies in the data on an Amazon EBS volume. They are designed to provide you with the information that you need to determine whether your Amazon EBS volumes are impaired, and to help you control how a potentially inconsistent volume is handled.

Volume status checks are automated tests that run every 5 minutes and return a pass or fail status. If all checks pass, the status of the volume is *ok*. If a check fails, the status of the volume is *impaired*. If the status is *insufficient-data*, the checks may still be in progress on the volume. You can view the results of volume status checks to identify any impaired volumes and take any necessary actions.

When Amazon EBS determines that a volume's data is potentially inconsistent, the default is that it disables I/O to the volume from any attached EC2 instances, which helps to prevent data corruption. After I/O is disabled, the next volume status check fails, and the volume status is *impaired*. In addition, you'll see an event that lets you know that I/O is disabled, and that you can resolve the impaired status of the volume by enabling I/O to the volume. We wait until you enable I/O to give you the opportunity to decide whether to continue to let your instances use the volume, or to run a consistency check using a command, such as *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.) For information about volume states, see *Volume state* (p. 1192).

If the consistency of a particular volume is not a concern, and you'd prefer that the volume be made available immediately if it's impaired, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the *Auto-Enable IO* volume attribute (*autoEnableIO* in the API), the volume status check continues to pass. In addition, you'll see an event that lets you know that the volume was determined to be potentially inconsistent, but that its I/O was automatically enabled. This enables you to check the volume's consistency or replace it at a later time.

The I/O performance status check compares actual volume performance to the expected performance of a volume. It alerts you if the volume is performing below expectations. This status check is available only for Provisioned IOPS SSD (*io1* and *io2*) volumes that are attached to an instance. The status check is not valid for General Purpose SSD (*gp2* and *gp3*), 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. It might take up to 5 minutes from the moment that you attach an *io1* or *io2* volume to an instance for the status check to report the I/O performance status.

**Important**

While initializing Provisioned IOPS SSD 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 Provisioned IOPS SSD volumes while you are initializing them. For more information, see *Initialize Amazon EBS volumes* (p. 1366).
The following table lists statuses for Amazon EBS volumes.

<table>
<thead>
<tr>
<th>Volume status</th>
<th>I/O enabled status</th>
<th>I/O performance status (io1 and io2 volumes only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ok</td>
<td>Enabled (I/O Enabled or I/O Auto-Enabled)</td>
<td>Normal (Volume performance is as expected)</td>
</tr>
<tr>
<td>warning</td>
<td>Enabled (I/O Enabled or I/O Auto-Enabled)</td>
<td>Degraded (Volume performance is below expectations)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Severely Degraded (Volume performance is well below expectations)</td>
</tr>
<tr>
<td>impaired</td>
<td>Enabled (I/O Enabled or I/O Auto-Enabled)</td>
<td>Stalled (Volume performance is severely impacted)</td>
</tr>
<tr>
<td></td>
<td>Disabled (Volume is offline and pending recovery, or is waiting for the user to enable I/O)</td>
<td>Not Available (Unable to determine I/O performance because I/O is disabled)</td>
</tr>
<tr>
<td>insufficient-data</td>
<td>Enabled (I/O Enabled or I/O Auto-Enabled)</td>
<td>Insufficient Data</td>
</tr>
<tr>
<td></td>
<td>Insufficient Data</td>
<td></td>
</tr>
</tbody>
</table>

You can view and work with status checks using the following methods.

**Console**

**To view status checks**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose Volumes. The Volume Status column displays the operational status of each volume.
3. To view the status details of a volume, select the volume and choose Status Checks.
4. If you have a volume with a failed status check (status is impaired), see Work with an impaired volume (p. 1200).

Alternatively, you can choose Events in the navigator to view all the events for your instances and volumes. For more information, see EBS volume events (p. 1199).

**AWS CLI**

**To view volume status information**

Use one of the following commands.
EBS volume events

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure.

To automatically enable I/O on a volume with potential data inconsistencies, change the setting of the Auto-Enabled IO volume attribute (autoEnableIO in the API). For more information about changing this attribute, see Work with an impaired volume (p. 1200).

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, io2, and gp3 volumes only. Volume performance is as expected.

Degraded

For io1, io2, and gp3 volumes only. Volume performance is below expectations.

Severely Degraded

For io1, io2, and gp3 volumes only. Volume performance is well below expectations.

Stalled

For io1, io2, and gp3 volumes only. Volume performance is severely impacted.

You can view events for your volumes using the following methods.

Console

To view events for your volumes

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Events. All instances and volumes that have events are listed.
3. You can filter by volume to view only volume status. You can also filter on specific status types.

4. Select a volume to view its specific event.

AWS CLI

To view events for your volumes

Use one of the following commands.

- describe-volume-status (AWS CLI)
- Get-EC2VolumeStatus (AWS Tools for Windows PowerShell)

For more information about these command line interfaces, see Access Amazon EC2 (p. 3).

If you have a volume where I/O is disabled, see Work with an impaired volume (p. 1200). If you have a volume where I/O performance is below normal, this might be a temporary condition due to an action you have taken (for example, creating a snapshot of a volume during peak usage, running the volume on an instance that cannot support the I/O bandwidth required, accessing data on the volume for the first time, etc.).

Work with an impaired volume

Use the following options if a volume is impaired because the volume's data is potentially inconsistent.

Options

- Option 1: Perform a consistency check on the volume attached to its instance (p. 1200)
- Option 2: Perform a consistency check on the volume using another instance (p. 1201)
- Option 3: Delete the volume if you no longer need it (p. 1202)

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. Use one of the following methods.

   **Console**
   1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
   2. In the navigation pane, choose *Volumes*.
   3. Select the volume on which to enable I/O operations.
   4. In the details pane, choose **Enable Volume IO**, and then choose **Yes, Enable**.

   ![Amazon EC2 console screenshot](image.png)

   **AWS CLI**

   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 *Access Amazon EC2* (p. 3).

   - `enable-volume-io` (AWS CLI)
   - `Enable-EC2VolumeIO` (AWS Tools for Windows PowerShell)

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 the AWS Support Center. Choose *Troubleshoot*, and then in the *Troubleshoot Status Checks* dialog box, choose *Contact Support* to submit a support case.

**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. For more information, see *Detach an Amazon EBS volume from a Windows instance* (p. 1204).
3. Enable I/O on the volume. Use one of the following methods.

   **Console**
   1. In the navigation pane, choose *Volumes*.
   2. Select the volume that you detached in the previous step.
   3. In the details pane, choose **Enable Volume IO**, and then choose **Yes, Enable**.
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 Access Amazon EC2 (p. 3).

- enable-volume-io (AWS CLI)
- Enable-EC2VolumeIO (AWS Tools for Windows PowerShell)

4. Attach the volume to another instance. For more information, see Launch your instance (p. 390) and Attach an Amazon EBS volume to an instance (p. 1186).

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 the AWS Support Center. Choose Troubleshoot, and then in the troubleshooting dialog box, choose Contact Support to submit a support case.

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 Delete an Amazon EBS volume (p. 1206).

If you have a recent snapshot that backs up the data on the volume, you can create a new volume from the snapshot. For more information, see Create a volume from a snapshot (p. 1185).

Work with the Auto-Enabled IO volume attribute

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure. If the consistency of a particular volume is not a concern, and you prefer that the volume be made available immediately if it's impaired, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the Auto-Enabled IO volume attribute (autoEnableIO in the API), I/O between the volume and the instance is automatically re-enabled and the volume's status check will pass. In addition, you'll see an event that lets you know that the volume was in a potentially inconsistent state, but that its I/O was automatically enabled. When this event occurs, you should check the volume's consistency and replace it if necessary. For more information, see EBS volume events (p. 1199).

You can view and modify the Auto-Enabled IO attribute of a volume using the following methods.

Console

To view the Auto-Enabled IO attribute of a volume

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Volumes.
3. Select the volume and choose Status Checks. Auto-Enabled IO displays the current setting (Enabled or Disabled) for your volume.

![Image showing Volumes and Status Checks]

To modify the Auto-Enabled IO attribute of a volume

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Volumes.
3. Select the volume and choose Actions, Change Auto-Enable IO Setting. Alternatively, choose the Status Checks tab, and for Auto-Enabled IO, choose Edit.

![Image showing Actions and Change Auto-Enable IO Setting]

4. Select the Auto-Enable Volume IO check box to automatically enable I/O for an impaired volume. To disable the feature, clear the check box.

![Image showing Change Auto-Enable IO Setting]

5. Choose Save.

AWS CLI

To view the autoEnableIO attribute of a volume

Use one of the following commands.
• describe-volume-attribute (AWS CLI)
• Get-EC2VolumeAttribute (AWS Tools for Windows PowerShell)

**To modify the autoEnableIO attribute of a volume**

Use one of the following commands.

• modify-volume-attribute (AWS CLI)
• Edit-EC2VolumeAttribute (AWS Tools for Windows PowerShell)

For more information about these command line interfaces, see Access Amazon EC2 (p. 3)

**Detach an Amazon EBS volume from a Windows instance**

You need to detach an Amazon Elastic Block Store (Amazon EBS) volume from an instance before you can attach it to a different instance or delete it. Detaching a volume does not affect the data on the volume.

For information about detaching volumes from a Linux instance, see Detach a volume from a Linux instance in the Amazon EC2 User Guide for Linux Instances.

**Topics**

• Considerations (p. 239)
• Unmount and detach a volume (p. 1204)
• Troubleshoot (p. 1206)

**Considerations**

• 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.
• 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.
• 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 Delete an Amazon EBS volume (p. 1206).

**Unmount and detach a volume**

Use the following procedures to unmount and detach a volume from an instance. This can be useful when you need to attach the volume to a different instance or when you need to delete the volume.

**Steps**

• Step 1: Unmount the volume (p. 1205)
• Step 2: Detach the volume from the instance (p. 1205)
• Step 3: Uninstall the offline device locations (p. 1205)
Step 1: Unmount the volume

From your Windows instance, unmount the volume as follows.

1. Start the Disk Management utility.
   - (Windows Server 2012 and later) On the taskbar, right-click the Windows logo and choose Disk Management.
2. Right-click the disk (for example, right-click Disk 1) and then choose Offline. Wait for the disk status to change to Offline before opening the Amazon EC2 console.

Step 2: Detach the volume from the instance

To detach the volume from the instance, use one of the following methods:

Console

To detach 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, Detach Volume.
4. When prompted for confirmation, choose Yes, Detach.

Command line

To detach an EBS volume from an instance using the command line

After unmounting the volume, you can use one of the following commands to detach it. For more information about these command line interfaces, see Access Amazon EC2 (p. 3).

- detach-volume (AWS CLI)
- Dismount-EC2Volume (AWS Tools for Windows PowerShell)

Step 3: Uninstall the offline device locations

When you unmount and detach a volume from an instance, Windows flags the device location as offline. The device location remains offline after rebooting, and stopping and restarting the instance. When you restart the instance, Windows might mount one of the remaining volumes to the offline device location. This causes the volume to be unavailable in Windows. To prevent this from happening and to ensure that all volumes are attached to online device locations the next time Windows starts, perform the following steps:

1. On the instance, open the Device Manager.
2. In the Device Manager, select View, Show hidden devices.
3. In the list of devices, expand the Storage controllers node.

   The device locations to which the detached volumes were mounted should appear greyed out.
4. Right-click each greyed out device location, select Uninstall device and choose Uninstall.

   Important
   Do not select the Delete the driver software for this device check box.
Troubleshoot

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 can 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:

```json
"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.

Delete an Amazon EBS volume

After you no longer need an Amazon EBS volume, you can delete it. After deletion, its data is gone and the volume can't be attached to any instance. However, before deletion, you can store a snapshot of the volume, which you can use to re-create the volume later.

**Note**
You can't delete a volume if it's attached to an instance. To delete a volume, you must first detach it. For more information, see [Detach an Amazon EBS volume from a Windows instance](#).

You can check if a volume is attached to an instance. In the console, on the **Volumes** page, you can view the state of your volumes.

- If a volume is attached to an instance, it's in the **in-use** state.
• If a volume is detached from an instance, it's in the available state. You can delete this volume.

You can delete an EBS volume using one of the following methods.

Console

To delete an EBS volume using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Volumes.
3. Select a volume and choose Actions, Delete Volume. If Delete Volume is greyed out, the volume is attached to an instance.
4. In the confirmation dialog box, choose Yes, Delete.

AWS CLI

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 Access Amazon EC2 (p. 3).

• delete-volume (AWS CLI)
• Remove-EC2Volume (AWS Tools for Windows PowerShell)

Amazon EBS snapshots

You can back up the data on your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves on storage costs by not duplicating data. Each snapshot contains all of the information that is needed to restore your data (from the moment when the snapshot was taken) to a new EBS volume.

When you create an EBS volume based on a snapshot, the new volume begins as an exact replica of the original volume that was used to create the snapshot. The replicated volume loads data in the background so that you can begin using it immediately. If you access data that hasn't been loaded yet, the volume immediately downloads the requested data from Amazon S3, and then continues loading the rest of the volume's data in the background. For more information, see Create Amazon EBS snapshots (p. 1211).

When you delete a snapshot, only the data unique to that snapshot is removed. For more information, see Delete an Amazon EBS snapshot (p. 1226).

Snapshot events

You can track the status of your EBS snapshots through CloudWatch Events. For more information, see EBS snapshot events (p. 1385).

Application-consistent snapshots

Using Systems Manager Run Command, you can take application-consistent snapshots of all 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. You don't need to shut down
your instances or disconnect them when you back up all attached volumes. For more information, see Creating a VSS Application-Consistent Snapshot.

**Multi-volume snapshots**

Snapshots can be used to create a backup of critical workloads, such as a large database or a file system that spans across multiple EBS volumes. Multi-volume snapshots allow you to take exact point-in-time, data coordinated, and crash-consistent snapshots across multiple EBS volumes attached to an EC2 instance. You are no longer required to stop your instance or to coordinate between volumes to ensure crash consistency, because snapshots are automatically taken across multiple EBS volumes. For more information, see the steps for creating a multi-volume EBS snapshot under Create Amazon EBS snapshots (p. 1211).

**Snapshot pricing**

Charges for your snapshots are based on the amount of data stored. Because snapshots are incremental, deleting a snapshot might not reduce your data storage costs. Data referenced exclusively by a snapshot is removed when that snapshot is deleted, but data referenced by other snapshots is preserved. For more information, see Amazon Elastic Block Store Volumes and Snapshots in the AWS Billing and Cost Management User Guide.

**Contents**

- How incremental snapshots work (p. 1208)
- Copy and share snapshots (p. 1210)
- Encryption support for snapshots (p. 1211)
- Create Amazon EBS snapshots (p. 1211)
- Create a VSS application-consistent snapshot (p. 1214)
- Delete an Amazon EBS snapshot (p. 1226)
- Copy an Amazon EBS snapshot (p. 1229)
- View Amazon EBS snapshot information (p. 1233)
- Share an Amazon EBS snapshot (p. 1234)
- Amazon EBS local snapshots on Outposts (p. 1238)
- Use EBS direct APIs to access the contents of an EBS snapshot (p. 1248)
- Automate the snapshot lifecycle (p. 1272)

**How incremental snapshots work**

This section shows how an EBS snapshot captures the state of a volume at a point in time, and how successive snapshots of a changing volume create a history of those changes.

**Relations among multiple snapshots of the same volume**

The diagram in this section shows Volume 1 at three points in time. A snapshot is taken of each of these three volume states. The diagram specifically shows the following:

- 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 being copied again. This is indicated by the dashed arrow.
- In State 3, 2 GiB of data have been added to the volume, for a total of 12 GiB. Snap C needs to copy the 2 GiB that were added after Snap B was taken. As shown by the dashed arrows, Snap C also references 4 GiB of data stored in Snap B, and 6 GiB of data stored in Snap A.
• The total storage required for the three snapshots is 16 GiB.

Relations among incremental snapshots of different volumes

The diagram in this section shows how incremental snapshots can be taken from different volumes.

**Important**
The diagram assumes that you own Vol 1 and that you have created Snap A. If Vol 1 was owned by another AWS account and that account took Snap A and shared it with you, then Snap B would be a full snapshot.
1. **Vol 1** has 10 GiB of data. Because **Snap A** is the first snapshot taken of the volume, the entire 10 GiB of data is copied and stored.

2. **Vol 2** is created from **Snap A**, so it is an exact replica of **Vol 1** at the time the snapshot was taken.

3. Over time, 4 GiB of data is added to **Vol 2** and its total size becomes 14 GiB.

4. **Snap B** is taken from **Vol 2**. For **Snap B**, only the 4 GiB of data that was added after the volume was created from **Snap A** is copied and stored. The other 10 GiB of unchanged data, which is already stored in **Snap A**, is referenced by **Snap B** instead of being copied and stored again.

   **Snap B** is an incremental snapshot of **Snap A**, even though it was created from a different volume.

For more information about how data is managed when you delete a snapshot, see [Delete an Amazon EBS snapshot](p. 1226).

**Copy and share 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 [Share an Amazon EBS snapshot](p. 1234).
A snapshot is constrained to the AWS Region where it was created. After you create a snapshot of an EBS volume, you can use it to create new volumes in the same Region. For more information, see Create a volume from a snapshot (p. 1185). 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 Copy an Amazon EBS snapshot (p. 1229).

Encryption support for snapshots

EBS snapshots fully support EBS encryption.

- Snapshots of encrypted volumes are automatically encrypted.
- Volumes that you create from encrypted snapshots are automatically encrypted.
- Volumes that you create from an unencrypted snapshot that you own or have access to can be encrypted on-the-fly.
- When you copy an unencrypted snapshot that you own, you can encrypt it during the copy process.
- When you copy an encrypted snapshot that you own or have access to, you can reencrypt it with a different key during the copy process.
- The first snapshot you take of an encrypted volume that has been created from an unencrypted snapshot is always a full snapshot.
- The first snapshot you take of a reencrypted volume, which has a different CMK compared to the source snapshot, is always a full snapshot.

Complete documentation of possible snapshot encryption scenarios is provided in Create Amazon EBS snapshots (p. 1211) and in Copy an Amazon EBS snapshot (p. 1229).

For more information, see Amazon EBS encryption (p. 1327).

Create Amazon EBS snapshots

To create an application-consistent snapshot, see Create a VSS application-consistent snapshot (p. 1214).

You can create a point-in-time snapshot of an EBS volume and use it as a baseline for new volumes or for data backup. If you make periodic snapshots of a volume, the snapshots are incremental—the new snapshot saves only the blocks that have changed since your last snapshot.

Snapshots occur asynchronously; the point-in-time snapshot is created immediately, but the status of the snapshot is pending until the snapshot is complete (when all of the modified blocks have been transferred to Amazon S3), which can take several hours for large initial snapshots or subsequent snapshots where many blocks have changed. While it is completing, an in-progress snapshot is not affected by ongoing reads and writes to the volume.

You can take a snapshot of an attached volume that is in use. However, snapshots only capture data that has been written to your Amazon EBS volume at the time the snapshot command is issued. This might exclude any data that has been cached by any applications or the operating system. If you can pause any file writes to the volume long enough to take a snapshot, your snapshot should be complete. However, if you can’t pause all file writes to the volume, you should unmount the volume from within the instance, issue the snapshot command, and then remount the volume to ensure a consistent and complete snapshot. You can remount and use your volume while the snapshot status is pending.

To make snapshot management easier, you can tag your snapshots during creation or add tags afterward. For example, you can apply tags describing the original volume from which the snapshot was created, or the device name that was used to attach the original volume to an instance. For more information, see Tag your Amazon EC2 resources (p. 1450).
Snapshot encryption

Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. The data in your encrypted volumes and any associated snapshots is protected both at rest and in motion. For more information, see Amazon EBS encryption (p. 1327).

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 Share an Amazon EBS snapshot (p. 1234).

You can share an encrypted snapshot only with specific AWS accounts. For others to use your shared, encrypted snapshot, you must also share the CMK key that was used to encrypt it. Users with access to your encrypted snapshot must create their own personal copy of it and then use that copy. Your copy of a shared, encrypted snapshot can also be re-encrypted using a different key. For more information, see Share an Amazon EBS snapshot (p. 1234).

Multi-volume snapshots

You can create multi-volume snapshots, which are point-in-time snapshots for all EBS volumes attached to an EC2 instance. You can also create lifecycle policies to automate the creation and retention of multi-volume snapshots. For more information, see Amazon Data Lifecycle Manager (p. 1272).

After the snapshots are created, each snapshot is treated as an individual snapshot. You can perform all snapshot operations, such as restore, delete, and copy across Regions or accounts, just as you would with a single volume snapshot. You can also tag your multi-volume snapshots as you would a single volume snapshot. We recommend you tag your multiple volume snapshots to manage them collectively during restore, copy, or retention.

Multi-volume, crash-consistent snapshots are typically restored as a set. It is helpful to identify the snapshots that are in a crash-consistent set by tagging your set with the instance ID, name, or other relevant details. You can also choose to automatically copy tags from the source volume to the corresponding snapshots. This helps you to set the snapshot metadata, such as access policies, attachment information, and cost allocation, to match the source volume.

After creating your snapshots, they appear in your EC2 console created at the exact point-in-time.

If any one snapshot for the multi-volume snapshot set fails, all of the other snapshots display an error status and a createSnapshots CloudWatch event with a result of failed is sent to your AWS account. For more information, see Create snapshots (createSnapshots) (p. 1385).

Amazon Data Lifecycle Manager

You can create, retain, and delete snapshots manually, or you can use Amazon Data Lifecycle Manager to manage your snapshots for you. For more information, see Amazon Data Lifecycle Manager (p. 1272).

Considerations

The following considerations apply to creating snapshots:

- When you create a snapshot for an EBS volume that serves as a root device, you should stop the instance before taking the snapshot.
- You cannot create snapshots from instances for which hibernation is enabled.
- You cannot create snapshots from hibernated instances.
- Although you can take a snapshot of a volume while a previous snapshot of that volume is in the pending status, having multiple pending snapshots of a volume can result in reduced volume performance until the snapshots complete.
There is a limit of one pending snapshot for a single st1 or sc1 volume, or five pending snapshots for a single volume of the other volume types. If you receive a ConcurrentSnapshotLimitExceeded error while trying to create multiple concurrent snapshots of the same volume, wait for one or more of the pending snapshots to complete before creating another snapshot of that volume.

When a snapshot is created from a volume with an AWS Marketplace product code, the product code is propagated to the snapshot.

Create a snapshot

To create a snapshot from the specified volume, use one of the following methods.

Console

**To create a snapshot using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Snapshots** under **Elastic Block Store** in the navigation pane.
3. Choose **Create Snapshot**.
4. For **Select resource type**, choose **Volume**.
5. For **Volume**, select the volume.
6. (Optional) Enter a description for the snapshot.
7. (Optional) Choose **Add Tag** to add tags to your snapshot. For each tag, provide a tag key and a tag value.
8. Choose **Create Snapshot**.

AWS CLI

**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 Access Amazon EC2 (p. 3).

- create-snapshot (AWS CLI)
- New-EC2Snapshot (AWS Tools for Windows PowerShell)

Create a multi-volume snapshot

To create a snapshot from the volumes of an instance, use one of the following methods.

Console

**To create multi-volume snapshots using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Snapshots** under **Elastic Block Store** in the navigation pane.
3. Choose **Create Snapshot**.
4. For **Select resource type**, choose **Instance**.
5. Select the instance ID for which you want to create simultaneous backups for all of the attached EBS volumes. Multi-volume snapshots support up to 40 EBS volumes per instance.
6. (Optional) Set **Exclude root volume**.

7. (Optional) Set **Copy tags from volume** flag to automatically copy tags from the source volume to the corresponding snapshots. This sets snapshot metadata—such as access policies, attachment information, and cost allocation—to match the source volume.

8. (Optional) Choose **Add Tag** to add tags to your snapshot. For each tag, provide a tag key and a tag value.

9. Choose **Create Snapshot**.

**AWS CLI**

**To create multi-volume snapshots using the command line**

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

- `create-snapshots` (AWS CLI)
- `New-EC2SnapshotBatch` (AWS Tools for Windows PowerShell)

**To create application-consistent snapshots using Systems Manager Run Command**

You can use Systems Manager Run Command to take application-consistent snapshots of all 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. You don't need to shut down your instances or disconnect them when you back up all attached volumes. For more information, see [Create a VSS application-consistent snapshot (p. 1214)](#).

If all of the snapshots complete successfully, a `createSnapshots` CloudWatch event with a result of `succeeded` is sent to your AWS account. If any one snapshot for the multi-volume snapshot set fails, all of the other snapshots display an error status and a `createSnapshots` CloudWatch event with a result of `failed` is sent to your AWS account. For more information, see [Create snapshots (createSnapshots) (p. 1385)](#).

**Work with EBS snapshots**

You can copy snapshots, share snapshots, and create volumes from snapshots. For more information, see the following:

- [Copy an Amazon EBS snapshot (p. 1229)](#)
- [Share an Amazon EBS snapshot (p. 1234)](#)
- [Create a volume from a snapshot (p. 1185)](#)

**Create a VSS application-consistent snapshot**

You can take application-consistent snapshots of all Amazon EBS volumes attached to your Windows on Amazon EC2 instances using [AWS Systems Manager Run Command](#). The snapshot process uses the Windows Volume Shadow Copy Service (VSS) to take image-level backups of VSS-aware applications. The snapshots include data from pending transactions between these applications and the disk. You don't have to shut down your instances or disconnect them when you need to back up all attached volumes.

There is no additional cost to use VSS-enabled EBS snapshots. You only pay for EBS snapshots created by the backup process. For more information, see [How is my EBS snapshot bill calculated?](#)
Contents

- How it works (p. 1215)
- Before you begin (p. 1215)
- Get started (p. 1216)
- Create a VSS application-consistent snapshot using the AWS CLI, AWS Tools for Windows PowerShell, or the AWSEC2-ManageVssIO SSM document (p. 1220)
- Restore volumes from VSS-enabled EBS snapshots (p. 1225)
- AWS VSS component package version history (p. 1226)

How it works

The process for taking application-consistent, VSS-enabled EBS snapshots consists of the following steps.

1. Complete Systems Manager prerequisites.
2. Enter parameters for the AWSEC2-CreateVssSnapshot SSM document and run this document by using Run Command. You can't create a VSS-enabled EBS snapshot for a specific volume. You can, however, specify a parameter to exclude the boot volume from the backup process.
3. The VSS agent on your instance coordinates all ongoing I/O operations for running applications.
4. The system flushes all I/O buffers and temporarily pauses all I/O operations. The pause lasts, at most, ten seconds.
5. During the pause, the system creates snapshots of all volumes attached to the instance.
6. The pause is lifted and I/O resumes operation.
7. The system adds all newly-created snapshots to the list of EBS snapshots. The system tags all VSS-enabled EBS snapshots successfully created by this process with AppConsistent:true. This tag helps you identify snapshots created by this process, as opposed to other processes. If the system encounters an error, the snapshot created by this process does not include the AppConsistent:true tag.
8. If you need to restore from a snapshot, you can use the standard EBS process of creating a volume from a snapshot, or you can restore all volumes to an instance by using a sample script, which is described later in this section.

Before you begin

Before you create VSS-enabled EBS snapshots by using Run Command, review the following requirements and limitations, and complete the required tasks.

Important

The AWSVssComponents package and the AWSEC2-CreateVssSnapshot and AWSEC2-ManageVssIO SSM documents no longer receive updates for Windows Server 2008 R2. The AWSVssComponents package supports Windows Server 2008 R2 up to version 1.3.1.0 and no later.

You can query the latest version of Windows 2008 R2 supported by the AWSEC2-CreateVssSnapshot and AWSEC2-ManageVssIO SSM documents by using the GetDocument API and specifying 2008R2 for -VersionName. For example:

Get-SSMDocument -Name AWSEC2-CreateVssSnapshot -VersionName "2008R2"

Amazon EC2 Windows instance requirements

VSS-enabled EBS snapshots are supported for instances running Windows Server 2012 or later. Verify that your instances meet all requirements for Amazon EC2 Windows. For more information, see Setting Up AWS Systems Manager in the AWS Systems Manager User Guide.
.NET Framework version

The AWSVssComponents package requires .NET Framework version 4.6 and later. If you are using Windows 2012, or 2012 R2, the default .NET Framework version is earlier than 4.6 and you must install version 4.6 or later using Windows Update.

SSM Agent version

Update your instances to use SSM Agent version 2.2.58.0 or later. If you are using an older version of SSM Agent, you can update it by using Run Command. For more information, see Update SSM Agent by using Run Command in the AWS Systems Manager User Guide.

AWS Tools for Windows PowerShell version

Ensure that your instance is running version 3.3.48.0 or later of the AWS Tools for Windows PowerShell. To check your version number, run the following command on the instance:

```
Get-AWSPowerShellVersion
```


Get started

These instructions describe how to install the VSS components and perform an application-consistent snapshot of the EBS volumes attached to an EC2 Windows instance. For more information, see Getting Started with Amazon EC2 Windows Instances.

Contents

- Create an IAM role for VSS-enabled snapshots (p. 1216)
- Download and install VSS components to the Windows on EC2 instance (p. 1218)
- Create a VSS application-consistent snapshot using the console (p. 1219)

Create an IAM role for VSS-enabled snapshots

The following procedures describes how to work with IAM policies and IAM roles. The policy enables Systems Manager to create snapshots, tags snapshots, and attach metadata like a device ID to the default snapshot tags that the system creates.

To create an IAM policy for VSS-enabled snapshots

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose Policies, and then choose Create policy.
3. On the Create policy page, choose the JSON tab, and then replace the default content with the following JSON policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Action": "ec2:CreateTags",
        "Resource": [
            "arn:aws:ec2:*::snapshot/*",
            "arn:aws:ec2:*::image/*"
        ]
    }
}
```

```
If you do not intend to set the `CreateAmi` parameter to `True`, then you can omit `arn:aws:ec2:*::image/*` from the first policy statement and you can omit `ec2:CreateImage` and `ec2:DescribeImages` from the second policy statement.

If you intend to always set the `CreateAmi` parameter to `True`, then you can omit `ec2:CreateSnapshot` from the second policy statement.

4. Choose **Review policy**.

5. For **Name**, enter a name to identify the policy, such as `VssSnapshotRole` or another name that you prefer.

6. (Optional) For **Description**, enter a description of the role's purpose.

7. Choose **Create policy**.

Use the following procedure to create an IAM role for VSS-enabled snapshots. This role includes policies for Amazon EC2 and Systems Manager.

**To create an IAM role for VSS-enabled EBS snapshots**

2. In the navigation pane, choose **Roles**, and then choose **Create role**.
3. Under **Select type of trusted entity**, choose **AWS Service**.
4. Immediately under **Choose the service that will use this role**, choose **EC2**, and then choose **Next: Permissions**.
5. Under **Select your use case**, choose **EC2**, and then choose **Next: Permissions**.
6. In the list of policies, choose the box next to `AmazonSSMManagedInstanceCore`. (Type `SSM` in the search box if you need to narrow the list.)
7. Choose **Next: Tags**.
8. (Optional) Add one or more tag key-value pairs to organize, track, or control access for this role, and then choose **Next: Review**.
9. For **Role name**, enter a name for the role, such as `VssSnapshotRole` or another name that you prefer.
10. (Optional) For **Role description**, replace the default text with a description of this role's purpose.
11. Choose **Create role**. The system returns you to the **Roles** page.
12. Choose the role that you just created. The role **Summary page** opens.
13. Choose **Attach policies**.
14. Search for and choose the box next to the policy your created in the previous procedure, such as `VssSnapshotRole` or another name that you chose.
15. Choose **Attach policy**.
16. Attach this role to the instances for which you want to create VSS-enabled EBS snapshots. For more information, see Attach an IAM role to an instance (p. 1120).

Download and install VSS components to the Windows on EC2 instance

Systems Manager requires VSS components to be installed on your instances. Use the following procedure to install the components using the AWSVssComponents package. The package installs two components: a VSS requester and a VSS provider. We recommend that you install the latest AWS VSS component package to improve reliability and performance of application-consistent snapshots on your EC2 Windows instances. To view the latest package version, see the AWS VSS component package version history (p. 1226).

2. In the navigation pane, choose Run Command.
3. Choose Run command.
4. For Command document, choose the button next to AWS-ConfigureAWSPackage.
5. For Command parameters, do the following:
   a. Verify that Action is set to Install.
   b. For Name, enter AwsVssComponents.
   c. For Version, leave the field empty so that Systems Manager installs the latest version.
6. For Targets, identify the instances on which you want to run this operation by specifying tags or selecting instances manually.

   Note
   If you choose to select instances manually, and an instance you expect to see is not included in the list, see Where Are My Instances? in the AWS Systems Manager User Guide for troubleshooting tips.

7. For Other parameters:
   • (Optional) For Comment, type information about this command.
   • For Timeout (seconds), specify the number of seconds for the system to wait before failing the overall command execution.
8. (Optional) For Rate control:
   • For Concurrency, specify either a number or a percentage of instances on which to run the command at the same time.

   Note
   If you selected targets by choosing Amazon EC2 tags, and you are not certain how many instances use the selected tags, then limit the number of instances that can run the document at the same time by specifying a percentage.
   • For Error threshold, specify when to stop running the command on other instances after it fails on either a number or a percentage of instances. For example, if you specify three errors, then Systems Manager stops sending the command when the fourth error is received. Instances still processing the command might also send errors.
9. (Optional) For Output options section, if you want to save the command output to a file, select the box next to Enable writing to an S3 bucket. Specify the bucket and (optional) prefix (folder) names.

   Note
   The S3 permissions that grant the ability to write the data to an S3 bucket are those of the instance profile assigned to the instance, not those of the IAM user performing this task. For more information, see Create an IAM Instance Profile for Systems Manager in the AWS Systems Manager User Guide.

10. (Optional) Specify options for SNS notifications.
For information about configuring Amazon SNS notifications for Run Command, see Configuring Amazon SNS Notifications for AWS Systems Manager.


Create a VSS application-consistent snapshot using the console

Use the following procedure to create a VSS-enabled EBS snapshot.

To create VSS-enabled EBS snapshots using the console

2. In the navigation pane, choose Run Command.
3. Choose Run command.
4. For Command document, choose AWSEC2-CreateVssSnapshot for the Document name, then choose Latest version at runtime as the Document version.
5. For Targets, identify the instances on which you want to run this operation by specifying tags or selecting instances manually.

   Note
   If you choose to select instances manually, and an instance you expect to see is not included in the list, see Where Are My Instances? for troubleshooting tips.

6. For Command parameters, do the following:

   a. Choose an option from the Exclude Boot Volume list. Use this parameter to exclude boot volumes from the backup process.
   b. (Optional) For Description field, type a description. This description is applied to any snapshot created by this process.
   c. (Optional) For Tags, type keys and values for tags that you want to apply to any snapshot created by this process. Tags can help you locate, manage, and restore volumes from a list of snapshots. By default, the system populates the tag parameter with a Name key. For the value of this key, specify a name that you want to apply to snapshots created by this process. If you want to specify additional tags, separate tags by using a semicolon. For example, Key=Environment,Value=Test;Key=User,Value=TestUser1.

   We recommended that you tag snapshots. By default, the systems tags snapshots with the device ID, and AppConsistent (for indicating successful, application-consistent VSS-enabled EBS snapshots).
   d. For Copy Only, choose True to perform a copy only backup operation. This option is set to False by default so that the system performs a full backup operation. A full backup operation prevents the system from breaking the differential backup chain in SQL Server when performing a backup.

      Note
      This option requires that AWS VSS provider version 1.2.00 or later be installed.
   e. For No Writers, choose True to exclude application VSS writers from the snapshot process. This can help you resolve conflicts with third-party VSS backup components. This option is set to False by default.

      Note
      This option requires that AWS VSS provider version 1.2.00 or later be installed.
   f. For CreateAmi, choose True to create an Amazon Machine Image (AMI) backup that is VSS-enabled, instead of an EBS snapshot. This option is set to False by default. For more information about creating an AMI, see Create a Windows AMI from a running instance.
   g. (Optional) For AmiName, specify a name for the created AMI. This option applies only if the CreateAmi option is set to True.
7. For **Other parameters**:  
   • For **Comment**, type information about this command.  
   • For **Timeout (seconds)**, specify the number of seconds for the system to wait before failing the overall command execution.

8. (Optional) For **Rate control**:  
   • For **Concurrence**, specify either a number or a percentage of instances on which to run the command at the same time.  
     
     **Note**  
     If you selected targets by choosing Amazon EC2 tags, and you are not certain how many instances use the selected tags, then limit the number of instances that can run the document at the same time by specifying a percentage.  
   • For **Error threshold**, specify when to stop running the command on other instances after it fails on either a number or a percentage of instances. For example, if you specify three errors, then Systems Manager stops sending the command when the fourth error is received. Instances still processing the command might also send errors.

9. (Optional) For **Output options**, to save the command output to a file, select the box next to **Enable writing to an S3 bucket**. Specify the bucket and (optional) prefix (folder) names.  
   
   **Note**  
   The S3 permissions that grant the ability to write the data to an S3 bucket are those of the instance profile assigned to the instance, not those of the IAM user performing this task. For more information, see [Setting Up Systems Manager](#).

10. (Optional) Specify options for **SNS notifications**.  
    
    For information about configuring Amazon SNS notifications for Run Command, see [Configuring Amazon SNS Notifications for AWS Systems Manager](#) in the [AWS Systems Manager User Guide](#).

11. Choose **Run**.  
    
    If successful, the command populates the list of EBS snapshots with the new snapshots. You can locate these snapshots in the list of EBS snapshots by searching for the tags you specified, or by searching for **AppConsistent**. If the command execution failed, view the Systems Manager command output for details about why the execution failed. If the command successfully completed, but a specific volume backup failed, you can troubleshoot the failure in the list of EBS volumes.  

    If the command failed and you are using Systems Manager with VPC endpoints, verify that you configured the `com.amazonaws.region.ec2` endpoint. Without the EC2 endpoint defined, the call to enumerate attached EBS volumes fails, which causes the Systems Manager command to fail. For more information about setting up VPC endpoints with Systems Manager, see [Create a Virtual Private Cloud Endpoint](#) in the [AWS Systems Manager User Guide](#).  

    **Note**  
    You can automate backups by creating a maintenance window task that uses the `AWSEC2-CreateVssSnapshot` SSM document. For more information, see [Working with Maintenance Windows (Console)](#).

---

**Create a VSS application-consistent snapshot using the AWS CLI, AWS Tools for Windows PowerShell, or the AWSEC2-ManageVssIO SSM document**

This section includes procedures for creating VSS-enabled EBS snapshots by using the AWS CLI or AWS Tools for Windows PowerShell. It also contains an advanced method for creating VSS-enabled snapshots using the `AWSEC2-ManageVssIO` SSM document.

**Contents**  
- Install the VSS package using the AWS CLI or Tools for Windows PowerShell (p. 1221)
• Create VSS-enabled EBS snapshots using the AWS CLI, Tools for Windows PowerShell, or the AWSEC2-ManageVssIO SSM document (p. 1222)
• Troubleshoot VSS-enabled EBS snapshots (p. 1224)

Install the VSS package using the AWS CLI or Tools for Windows PowerShell

Use one of the following command-line procedures to download and install the VSS components to the Windows on EC2 instance.

Install the VSS package by using the AWS CLI

Use the following procedure to download and install the `AwsVssComponents` package on your instances by using Run Command from the AWS CLI. The package installs two components: a VSS requestor and a VSS provider. The system copies these components to a directory on the instance, and then registers the provider DLL as a VSS provider.

To install the VSS package by using the AWS CLI

1. Install and configure the AWS CLI, if you have not already.
   
   For information, see Install or Upgrade and then Configure the AWS CLI in the AWS Systems Manager User Guide.
2. Run the following command to download and install the required VSS components for Systems Manager.

   ```bash
   aws ssm send-command --document-name "AWS-ConfigureAWSPackage" --instance-ids "i-12345678" --parameters '{"action":["Install"],"name":["AwsVssComponents"]}'
   ```

Install the VSS package by using Tools for Windows PowerShell

Use the following procedure to download and install the `AwsVssComponents` package on your instances by using Run Command from the Tools for Windows PowerShell. The package installs two components: a VSS requestor and a VSS provider. The system copies these components to a directory on the instance, and then registers the provider DLL as a VSS provider.

To install the VSS package by using AWS Tools for Windows PowerShell

1. Open AWS Tools for Windows PowerShell and run the following command to specify your credentials. You must either have administrator privileges in Amazon EC2 or have been granted the appropriate permission in IAM. For more information, see Setting Up AWS Systems Manager in the AWS Systems Manager User Guide.

   ```powershell
   Set-AWSCredentials -AccessKey key_name -SecretKey key_name
   ```
2. Run the following command to set the Region for your PowerShell session. The example uses the us-east-2 Region.

   ```powershell
   Set-DefaultAWSRegion -Region us-east-2
   ```
3. Run the following command to download and install the required VSS components for Systems Manager.

   ```powershell
   Send-SSMCommand -DocumentName AWS-ConfigureAWSPackage -InstanceId "$instance"-Parameter @
   @
   @{"action"="Install";"name"="AwsVssComponents"}
   ```
Create VSS-enabled EBS snapshots using the AWS CLI, Tools for Windows PowerShell, or the AWSEC2-ManageVssIO SSM document

Use one of the following command-line procedures to create VSS-enabled EBS snapshots.

Create VSS-enabled EBS snapshots using the AWS CLI

Use the following procedure to create VSS-enabled EBS snapshots by using the AWS CLI. When you run the command, you can specify the following parameters:

- **Instance (Required)**: Specify one or more Amazon EC2 Windows instances. You can either manually specify instances, or you can specify tags.
- **Description (Optional)**: Specify details about this backup.
- **Tags (Optional)**: Specify key-value tag pairs that you want to assign to the snapshots. Tags can help you locate, manage, and restore volumes from a list of snapshots. By default, the system populates the tag parameter with a `Name` key. For the value of this key, specify a name that you want to apply to snapshots created by this process. You can also add custom tags to this list by using the following format: `Key=Environment,Value=Test;Key=User,Value=TestUser1`.

This parameter is optional, but we recommended that you tag snapshots. By default, the systems tags snapshots with the device ID, and `AppConsistent` (for indicating successful, application-consistent VSS-enabled EBS snapshots).
- **Exclude Boot Volume (Optional)**: Use this parameter to exclude boot volumes from the backup process.

To create VSS-enabled EBS snapshots by using the AWS CLI

1. Install and configure the AWS CLI, if you have not already.
   For information, see Install or Upgrade and then Configure the AWS CLI in the AWS Systems Manager User Guide.
2. Run the following command to create VSS-enabled EBS snapshots.

   ```bash
   aws ssm send-command --document-name "AWSEC2-CreateVssSnapshot" --instance-ids "i-12345678" --parameters '{"ExcludeBootVolume": ["False"], "description": ["Description"], "tags": ["Key=key_name,Value=tag_value"]}'
   ```

   If successful, the command populates the list of EBS snapshots with the new snapshots. You can locate these snapshots in the list of EBS snapshots by searching for the tags you specified, or by searching for `AppConsistent`. If the command execution failed, view the command output for details about why the execution failed.

   You can automate backups by creating a maintenance window task that uses the AWSEC2-CreateVssSnapshot SSM document. For more information, see Working with Maintenance Windows (Console) in the AWS Systems Manager User Guide.

Create VSS-enabled EBS snapshots using AWS Tools for Windows PowerShell

Use the following procedure to create VSS-enabled EBS snapshots by using the AWS Tools for Windows PowerShell. When you run the command, you can specify the following parameters:

- **Instance (Required)**: Specify one or more Amazon EC2 Windows instances. You can either manually specify instances, or you can specify tags.
- **Description (Optional)**: Specify details about this backup.
- **Tags (Optional)**: Specify key-value tag pairs that you want to assign to the snapshots. Tags can help you locate, manage, and restore volumes from a list of snapshots. By default, the system populates...
the tag parameter with a Name key. For the value of this key, specify a name that you want to apply to snapshots created by this process. You can also add custom tags to this list by using the following format: `Key=Environment,Value=Test;Key=User,Value=TestUser1`.

This parameter is optional, but we recommend that you tag snapshots. By default, the systems tags snapshots with the device ID, and AppConsistent (for indicating successful, application-consistent VSS-enabled EBS snapshots).

- Exclude Boot Volume (Optional): Use this parameter to exclude boot volumes from the backup process.

To create VSS-enabled EBS snapshots by using AWS Tools for Windows PowerShell

1. Open AWS Tools for Windows PowerShell and run the following command to specify your credentials. You must either have administrator privileges in Amazon EC2, or you must have been granted the appropriate permission in IAM. For more information, see Setting Up AWS Systems Manager in the AWS Systems Manager User Guide.

   ```powershell
   Set-AWSCredentials -AccessKey key_name -SecretKey key_name
   ```

2. Execute the following command to set the Region for your PowerShell session. The example uses the us-east-2 Region.

   ```powershell
   Set-DefaultAWSRegion -Region us-east-2
   ```

3. Execute the following command to create VSS-enabled EBS snapshots.

   ```powershell
   Send-SSMCommand -DocumentName AWSEC2-CreateVssSnapshot -InstanceId "$instance" -Parameter @{'ExcludeBootVolume'='False';'description'='a_description';'tags'='Key=key_name,Value=tag_value'}
   ```

If successful, the command populates the list of EBS snapshots with the new snapshots. You can locate these snapshots in the list of EBS snapshots by searching for the tags you specified, or by searching for AppConsistent. If the command execution failed, view the command output for details about why the execution failed. If the command successfully completed, but a specific volume backup failed, you can troubleshoot the failure in the list of EBS snapshots.

You can automate backups by creating a maintenance window task that uses the AWSEC2-CreateVssSnapshot SSM document. For more information, see Working with Maintenance Windows (Console) in the AWS Systems Manager User Guide.

Create VSS-enabled EBS snapshots by using the AWSEC2-ManageVssIO SSM document (advanced)

You can use the following script and the pre-defined AWSEC2-ManageVssIO SSM document to temporarily pause I/O, create VSS-enabled EBS snapshots, and restart I/O. This process runs in the context of the user who runs the command. If the user has sufficient permission to create and tag snapshots, then AWS Systems Manager can create and tag VSS-enabled EBS snapshots without the need for the additional IAM snapshot role on the instance.

In contrast, the AWSEC2-CreateVssSnapshot document requires that you assign the IAM snapshot role to each instance for which you want to create EBS snapshots. If you don't want to provide additional IAM permissions to your instances for policy or compliance reasons, then you can use the following script.

Before you begin

Note the following important details about this process:
This process uses a PowerShell script (CreateVssSnapshotAdvancedScript.ps1) to take snapshots of all volumes on the instances you specify, except root volumes. If you need to take snapshots of root volumes, then you must use the AWSEC2–CreateVssSnapshot SSM document.

The script calls the AWSEC2–ManageVssIO document twice. The first time with the Action parameter set to Freeze, which pauses all I/O on the instances. The second time, the Action parameter is set to Thaw, which forces I/O to resume.

Don't attempt to use the AWSEC2–ManageVssIO document without using the CreateVssSnapshotAdvancedScript.ps1 script. A limitation in VSS requires that the Freeze and Thaw actions be called no more than ten seconds apart, and manually calling these actions without the script could result in errors.

To create VSS-enabled EBS snapshots by using the AWSEC2–ManageVssIO SSM document

1. Open AWS Tools for Windows PowerShell and run the following command to specify your credentials. You must either have administrator privileges in Amazon EC2 or have been granted the appropriate permission in IAM. For more information, see Setting Up AWS Systems Manager in the AWS Systems Manager User Guide.

   ```powershell
   Set-AWSCredentials -AccessKey <key_name> -SecretKey <key_name>
   ```

2. Execute the following command to set the Region for your PowerShell session. The example uses the us-east-2 Region.

   ```powershell
   Set-DefaultAWSRegion -Region us-east-2
   ```

3. Download the CreateVssSnapshotAdvancedScript.zip file and extract the file contents.

4. Open CreateVssSnapshotAdvancedScript.ps1 in a text editor, edit the sample call at the bottom of the script with a valid EC2 instance ID, snapshot description, and desired tag values, and then run the script from PowerShell.

If successful, the command populates the list of EBS snapshots with the new snapshots. You can locate these snapshots in the list of EBS snapshots by searching for the tags you specified, or by searching for AppConsistent. If the command execution failed, view the command output for details about why the execution failed. If the command was successfully completed, but a specific volume backup failed, you can troubleshoot the failure in the list of EBS volumes.

Troubleshoot VSS-enabled EBS snapshots

General: Checking the log files

If you experience problems or receive error messages when creating VSS-enabled EBS snapshots, you can view the command output in the Systems Manager console. You can also view the following logs:

- `%ProgramData%\Amazon\SSM\InstanceData\InstanceID\document\orchestration \SSMCommandID\awsrunPowerShellScript\runPowerShellScript\stdout`
- `%ProgramData%\Amazon\SSM\InstanceData\InstanceID\document\orchestration \SSMCommandID\awsrunPowerShellScript\runPowerShellScript\stderr`

You can also open the Event Viewer Windows application and choose Windows Logs, Application to view additional logs. To see events specifically from the EC2 Windows VSS Provider and the Volume Shadow Copy Service, filter by Source on the terms Ec2VssSoftwareProvider and VSS.

Error: Thaw pipe connection timed out, error on thaw, timeout waiting for VSS Freeze, or other timeout errors
The EC2 Windows VSS Provider might time out due to activity or services on the instance preventing VSS-enabled snapshots from proceeding in a timely manner. The Windows VSS Framework provides a non-configurable 10-second window during which communication to the file system is paused. During this time, \texttt{AWSEC2-CreateVssSnapshot} snapshots your volumes.

The following items can cause the EC2 Windows VSS Provider to run into time limits during a snapshot:

- Excessive I/O to a volume
- Slow responsiveness of the EC2 API on the instance
- Fragmented volumes
- Incompatibility with some antivirus software
- Issues with a VSS Application writer
- When Module Logging is enabled for a large number of PowerShell modules, that can cause PowerShell scripts to run slowly

Usually, when running into time limits with the \texttt{AWSEC2-CreateVssSnapshot} command, the cause is related to the workload on the instance being too high at the time of backup. The following actions can help you take a successful snapshot:

- Retry the \texttt{AWSEC2-CreateVssSnapshot} command to see if the snapshot attempt is successful. If retrying succeeds in some cases, reducing the instance load might make snapshots more successful.
- Wait a while for the workload on the instance to decrease, and retry the \texttt{AWSEC2-CreateVssSnapshot} command. Alternatively, you can attempt snapshots when the instance is known to be under low stress.
- Attempt VSS snapshots when the antivirus software on the system is turned off. If this resolves the issue, refer to the antivirus software instructions and configure it to allow VSS snapshots.
- If there are a lot of EC2 API calls being made at the time of the snapshot, API throttling might cause the snapshots to take too long to start. Try taking snapshots again when there is less API activity in the account.
- Run the command \texttt{vssadmin list writers} in a shell and see if it reports any errors in the \texttt{Last error} field for any writers on the system. If any writers report a \texttt{time out} error, consider retrying snapshots when the instance is under less load.
- If one or more PowerShell modules have Group Policies that enable PowerShell module logging, try temporarily disabling the logging before you take a snapshot.

### Restore volumes from VSS-enabled EBS snapshots

You can use the \texttt{RestoreVssSnapshotSampleScript.ps1} script to restore volumes on an instance from VSS-enabled EBS snapshots. This script performs the following tasks:

- Stops an instance
- Removes all existing drives from the instance (except the boot volume, if it was excluded)
- Creates new volumes from the snapshots
- Attaches the volumes to the instance by using the device ID tag on the snapshot
- Restarts the instance

**Important**

The following script detaches all volumes attached to an instance, and then creates new volumes from a snapshot. Make sure that you have properly backed-up the instance. The old volumes are not deleted. If you want, you can edit the script to delete the old volumes.
To restore volumes from VSS-enabled EBS snapshots

1. Open AWS Tools for Windows PowerShell and run the following command to specify your credentials. You must either have administrator privileges in Amazon EC2 or have been granted the appropriate permission in IAM. For more information, see Setting Up AWS Systems Manager in the AWS Systems Manager User Guide.

   ```powershell
   Set-AWSCredentials -AccessKey key_name -SecretKey key_name
   ```

2. Run the following command to set the Region for your PowerShell session. The example uses the us-east-2 Region.

   ```powershell
   Set-DefaultAWSRegion -Region us-east-2
   ```

3. Download the RestoreVssSnapshotSampleScript.zip file and extract the file contents.
4. Open RestoreVssSnapshotSampleScript.zip in a text editor and edit the sample call at the bottom of the script with a valid EC2 instance ID and EBS snapshot ID, and then run the script from PowerShell.

AWS VSS component package version history

The following table describes the released versions of the AWS VSS component package.

<table>
<thead>
<tr>
<th>Version</th>
<th>Details</th>
</tr>
</thead>
</table>
| 1.3.1.0 | • Fixed snapshots failing on domain controllers in relation to an NTDS VSS writer logging error.  
• Fixed VSS agent error when uninstalling version 1.0 VSS provider. |
| 1.3.00  | • Improved logging by reducing unwanted verbosity.  
• Fixed regionalization issues during installation.  
• Fixed return codes for some provider registration error conditions.  
• Fixed various installation issues. |
| 1.2.00  | • Added command line parameters -nw (no-writers) and -copy (copy-only) to agent.  
• Fixed EventLog errors caused by improper memory allocation calls. |
| 1.1     | Fixed AwsVssProvider.dll being used incorrectly as the default Windows Backup and Restore provider. |
| 1.0     | Initial release. |

Delete an Amazon EBS snapshot

After you no longer need an Amazon EBS snapshot of a volume, you can delete it. Deleting a snapshot has no effect on the volume. Deleting a volume has no effect on the snapshots made from it.

Incremental snapshot deletion

If you make periodic snapshots of a volume, the snapshots are incremental. This means that only the blocks on the device that have changed after your most recent snapshot are saved in the new snapshot.
Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to create volumes.

If data was present on a volume held in an earlier snapshot or series of snapshots, and that data is subsequently deleted from the volume later on, the data is still considered to be unique data of the earlier snapshots. Unique data is only deleted from the sequence of snapshots if all snapshots that reference the unique data are deleted.

When you delete a snapshot, only the data that is referenced exclusively by that snapshot is removed. Unique data is only deleted if all of the snapshots that reference it are deleted. Deleting previous snapshots of a volume does not affect your ability to create volumes from later snapshots of that volume.

Deleting a snapshot might not reduce your organization's data storage costs. Other snapshots might reference that snapshot's data, and referenced data is always preserved. If you delete a snapshot containing data being used by a later snapshot, costs associated with the referenced data are allocated to the later snapshot. For more information about how snapshots store data, see How incremental snapshots work (p. 1208) and the following example.

In the following diagram, Volume 1 is shown at three points in time. A snapshot has captured each of the first two states, and in the third, a snapshot has been deleted.

- In State 1, the volume has 10 GiB of data. Because Snap A is the first snapshot taken of the volume, the entire 10 GiB of data must be copied.
- In State 2, the volume still contains 10 GiB of data, but 4 GiB have changed. Snap B needs to copy and store only the 4 GiB that changed after Snap A was taken. The other 6 GiB of unchanged data, which are already copied and stored in Snap A, are referenced by Snap B rather than (again) copied. This is indicated by the dashed arrow.
- In state 3, the volume has not changed since State 2, but Snapshot A has been deleted. The 6 GiB of data stored in Snapshot A that were referenced by Snapshot B have now been moved to Snapshot B, as shown by the heavy arrow. As a result, you are still charged for storing 10 GiB of data; 6 GiB of unchanged data preserved from Snap A and 4 GiB of changed data from Snap B.

Deleting a snapshot with some of its data referenced by another snapshot
Considerations

The following considerations apply to deleting snapshots:

- You can't delete a snapshot of the root device of an EBS volume used by a registered AMI. You must first deregister the AMI before you can delete the snapshot. For more information, see Deregister your Windows AMI (p. 51).
- You can't delete a snapshot that is managed by the AWS Backup service using Amazon EC2. Instead, use AWS Backup to delete the corresponding recovery points in the backup vault.
- You can create, retain, and delete snapshots manually, or you can use Amazon Data Lifecycle Manager to manage your snapshots for you. For more information, see Amazon Data Lifecycle Manager (p. 1272).
- Although you can delete a snapshot that is still in progress, the snapshot must complete before the deletion takes effect. This might take a long time. If you are also at your concurrent snapshot limit, and you attempt to take an additional snapshot, you might get a ConcurrentSnapshotLimitExceeded error. For more information, see the Service Quotas for Amazon EBS in the Amazon Web Services General Reference.

Delete a snapshot

To delete a snapshot, use one of the following methods.
EBS snapshots

Console

**To delete a snapshot using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](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**.

AWS CLI

**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 [Access Amazon EC2](p. 3).

- `delete-snapshot` (AWS CLI)
- `Remove-EC2Snapshot` (AWS Tools for Windows PowerShell)

Delete a multi-volume snapshot

To delete multi-volume snapshots, retrieve all of the snapshots for your multi-volume snapshot set using the tag you applied to the set when you created the snapshots. Then, delete the snapshots individually.

You will not be prevented from deleting individual snapshots in the multi-volume snapshot set. If you delete a snapshot while it is in the **pending** state, only that snapshot is deleted. The other snapshots in the multi-volume snapshot set still complete successfully.

Copy an Amazon EBS snapshot

With Amazon EBS, you can create point-in-time snapshots of volumes, which we store for you in Amazon S3. After you create a snapshot and it has finished copying to Amazon S3 (when the snapshot status is **completed**), you can copy it from one AWS Region to another, or within the same Region. Amazon S3 server-side encryption (256-bit AES) protects a snapshot’s data in transit during a copy operation. The snapshot copy receives an ID that is different from the ID of the original snapshot.

To copy multi-volume snapshots to another AWS Region, retrieve the snapshots using the tag you applied to the multi-volume snapshot set when you created it. Then individually copy the snapshots to another Region.

If you would like another account to be able to copy your snapshot, you must either modify the snapshot permissions to allow access to that account or make the snapshot public so that all AWS accounts can copy it. For more information, see [Share an Amazon EBS snapshot](p. 1234).

For information about copying an Amazon RDS snapshot, see [Copying a DB Snapshot](p. 1234) in the **Amazon RDS User Guide**.

Use cases

- Geographic expansion: Launch your applications in a new AWS Region.
- Migration: Move an application to a new Region, to enable better availability and to minimize cost.
- Disaster recovery: Back up your data and logs across different geographical locations at regular intervals. In case of disaster, you can restore your applications using point-in-time backups stored in the secondary Region. This minimizes data loss and recovery time.
- Encryption: Encrypt a previously unencrypted snapshot, change the key with which the snapshot is encrypted, or create a copy that you own in order to create a volume from it (for encrypted snapshots that have been shared with you).
• Data retention and auditing requirements: Copy your encrypted EBS snapshots from one AWS account to another to preserve data logs or other files for auditing or data retention. Using a different account helps prevent accidental snapshot deletions, and protects you if your main AWS account is compromised.

Prerequisites

• You can copy any accessible snapshots that have a completed status, including shared snapshots and snapshots that you have created.

• You can copy AWS Marketplace, VM Import/Export, and AWS Storage Gateway snapshots, but you must verify that the snapshot is supported in the destination Region.

Considerations

• Each account can have up to twenty concurrent snapshot copy requests to a single destination Region.

• User-defined tags are not copied from the source snapshot to the new snapshot. You can add user-defined tags during or after the copy operation. For more information, see Tag your Amazon EC2 resources (p. 1450).

• Snapshots created by a snapshot copy operation have an arbitrary volume ID that should not be used for any purpose.

• Resource-level permissions specified for the snapshot copy operation apply only to the new snapshot. You cannot specify resource-level permissions for the source snapshot. For an example, see Example: Copying snapshots (p. 1079).

Pricing

• For pricing information about copying snapshots across AWS Regions and accounts, see Amazon EBS Pricing.

• 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.

• If you copy a snapshot and encrypt it to a new KMS key, a complete (non-incremental) copy is created. This results in additional storage costs.

• If you copy a snapshot to a new Region, a complete (non-incremental) copy is created. This results in additional storage costs. Subsequent copies of the same snapshot are incremental.

Incremental snapshot copying

Whether a snapshot copy is incremental is determined by the most recently completed snapshot copy. When you copy a snapshot across Regions or accounts, the copy is an incremental copy if the following conditions are met:

• The snapshot was copied to the destination Region or account previously.

• The most recent snapshot copy still exists in the destination Region or account.

• All copies of the snapshot in the destination Region or account are either unencrypted or were encrypted using the same KMS key.

If the most recent snapshot copy was deleted, the next copy is a full copy, not an incremental copy. If a copy is still pending when you start another copy, the second copy starts only after the first copy finishes.

We recommend that you tag your snapshots with the volume ID and creation time so that you can keep track of the most recent snapshot copy of a volume in the destination Region or account.
To see whether your snapshot copies are incremental, check the `copySnapshot` (p. 1387) CloudWatch event.

**Encryption and snapshot copying**

When you copy a snapshot, you can encrypt the copy or you can specify a KMS key that is different than the original, and the resulting copied snapshot uses the new KMS key. 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 KMS key that you own. This protects you if the original KMS key is compromised, or if the owner revokes it, which could cause you to lose access to any encrypted volumes that you created using the snapshot. For more information, see Share an Amazon EBS snapshot (p. 1234).

You apply encryption to EBS snapshot copies by setting the `Encrypted` parameter to `true`. (The `Encrypted` parameter is optional if `encryption by default` (p. 1331) is enabled).

Optionally, you can use `KmsKeyId` to specify a custom key to use to encrypt the snapshot copy. (The `Encrypted` parameter must also be set to `true`, even if encryption by default is enabled.) If `KmsKeyId` is not specified, the key that is used for encryption depends on the encryption state of the source snapshot and its ownership.

The following tables describe the encryption outcome for each possible combination of settings.

**Topics**

- Encryption outcomes: Copying snapshots that you own (p. 1231)
- Encryption outcomes: Copying snapshots that are shared with you (p. 1232)

**Encryption outcomes: Copying snapshots that you own**

<table>
<thead>
<tr>
<th>Encryption by default</th>
<th>Is Encrypted parameter set?</th>
<th>Source snapshot encryption status</th>
<th>Default (no KMS key specified)</th>
<th>Custom (KMS key specified)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled</td>
<td>No</td>
<td>Unencrypted</td>
<td>Unencrypted</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encrypted</td>
<td>Encrypted by AWS managed key</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>Unencrypted</td>
<td>Encrypted by default KMS key</td>
<td>Encrypted by specified KMS key**</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encrypted</td>
<td>Encrypted by default KMS key</td>
<td></td>
</tr>
<tr>
<td>Enabled</td>
<td>No</td>
<td>Unencrypted</td>
<td>Encrypted by default KMS key</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encrypted</td>
<td>Encrypted by default KMS key</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>Unencrypted</td>
<td>Encrypted by default KMS key</td>
<td>Encrypted by specified KMS key**</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encrypted</td>
<td>Encrypted by default KMS key</td>
<td></td>
</tr>
</tbody>
</table>
** This is a customer managed key specified for the copy action. This customer managed key is used instead of the default customer managed key for the AWS account and Region.

**Encryption outcomes: Copying snapshots that are shared with you**

<table>
<thead>
<tr>
<th>Encryption by default</th>
<th>Is Encrypted parameter set?</th>
<th>Source snapshot encryption status</th>
<th>Default (no KmsKeyId specified)</th>
<th>Custom (KmsKeyId specified)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled</td>
<td>No</td>
<td>Unencrypted</td>
<td>Unencrypted</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encrypted</td>
<td>Encrypted by AWS managed key</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Unencrypted</td>
<td>Encrypted by default KMS key</td>
<td>Encrypted by specified KMS key**</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encrypted</td>
<td>Encrypted by default KMS key</td>
<td></td>
</tr>
<tr>
<td>Enabled</td>
<td>No</td>
<td>Unencrypted</td>
<td>Encrypted by default KMS key</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encrypted</td>
<td>Encrypted by default KMS key</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Unencrypted</td>
<td>Encrypted by default KMS key</td>
<td>Encrypted by specified KMS key**</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encrypted</td>
<td>Encrypted by default KMS key</td>
<td></td>
</tr>
</tbody>
</table>

** This is a customer managed key specified for the copy action. This customer managed key is used instead of the default customer managed key for the AWS account and Region.

**Copy a snapshot**

To copy a snapshot, use one of the following methods.

**Console**

**To copy a snapshot using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Snapshots**.
3. Select the snapshot to copy, and then choose **Copy** from the **Actions** list.
4. In the **Copy Snapshot** dialog box, update the following as necessary:
   - **Destination region**: Select the Region where you want to write the copy of the snapshot.
   - **Description**: By default, the description includes information about the source snapshot so that you can identify a copy from the original. You can change this description as necessary.
   - **Encryption**: If the source snapshot is not encrypted, you can choose to encrypt the copy. If you have enabled encryption by default (p. 1331), the **Encryption** option is set and cannot be unset from the snapshot console. If the **Encryption** option is set, you can choose to encrypt it to a customer managed CMK by selecting one in the field, described below.

You cannot strip encryption from an encrypted snapshot.
• **Master Key**: The customer master key (CMK) to be used to encrypt this snapshot. The default key for your account is displayed initially, but you can optionally select from the master keys in your account or type/paste the ARN of a key from a different account. You can create new master encryption keys in the AWS KMS 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.

**AWS CLI**

**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 [Access Amazon EC2](p. 3).

- `copy-snapshot` (AWS CLI)
- `Copy-EC2Snapshot` (AWS Tools for Windows PowerShell)

**To check for failure**

If you attempt to copy an encrypted snapshot without having permissions to use the encryption key, the operation fails silently. The error state is not displayed in the console until you refresh the page. You can also check the state of the snapshot from the command line, as in the following example.

```
aws ec2 describe-snapshots --snapshot-id snap-0123abcd
```

If the copy failed because of insufficient key permissions, you see the following message: "StateMessage": "Given key ID is not accessible".

When copying an encrypted snapshot, you must have `DescribeKey` permissions on the default CMK. Explicitly denying these permissions results in copy failure. For information about managing CMK keys, see [Controlling Access to Customer Master Keys](p. 3).

**View Amazon EBS snapshot information**

You can view detailed information about your snapshots using one of the following methods.

**Console**

**To view snapshot information using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. Choose **Snapshots** in the navigation pane.
3. To reduce the list, choose an option from the **Filter** list. For example, to view only your snapshots, choose **Owned By Me**. You can also filter your snapshots using tags and snapshot attributes. Choose the search bar to view the available tags and attributes.
4. To view more information about a snapshot, select it.
To view snapshot information using the command line

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

- `describe-snapshots` (AWS CLI)
- `Get-EC2Snapshot` (AWS Tools for Windows PowerShell)

**Example Example 1: Filter based on tags**

The following command describes the snapshots with the tag `Stack=production`.

```
aws ec2 describe-snapshots --filters Name=tag:Stack,Values=production
```

**Example Example 2: Filter based on volume**

The following command describes the snapshots created from the specified volume.

```
aws ec2 describe-snapshots --filters Name=volume-id,Values=vol-049df61146c4d7901
```

**Example Example 3: Filter based on snapshot age**

With the AWS CLI, you can use JMESPath to filter results using expressions. For example, the following command displays the IDs of all snapshots created by your AWS account (represented by `123456789012`) before the specified date (represented by `2020-03-31`). If you do not specify the owner, the results include all public snapshots.

```
aws ec2 describe-snapshots --filters Name=owner-id,Values=123456789012 --query "Snapshots[?(StartTime<=`2020-03-31`)].[SnapshotId]" --output text
```

The following command displays the IDs of all snapshots created in the specified date range.

```
aws ec2 describe-snapshots --filters Name=owner-id,Values=123456789012 --query "Snapshots[?(StartTime>=`2019-01-01`) && (StartTime<=`2019-12-31`)].[SnapshotId]" --output text
```

**Share an Amazon EBS snapshot**

You can modify the permissions of a snapshot if you want to share it with other AWS accounts. You can share snapshots publicly with all other AWS accounts, or you can share them privately with individual AWS accounts that you specify. Users that you have authorized can use the snapshots that you share to create their own EBS volumes, while your original snapshot remains unaffected.

**Important**

When you share a snapshot, you are giving others access to all of the data on the snapshot. Share snapshots only with people that you trust with all of your snapshot data.

**Topics**

- Before you share a snapshot (p. 1235)
- Share a snapshot (p. 1235)
- Share a KMS key (p. 1236)
View snapshots that are shared with you (p. 1237)
Use snapshots that are shared with you (p. 1238)
Determine the use of snapshots that you share (p. 1238)

Before you share a snapshot

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 and then share the copy. For more information, see Copy an Amazon EBS snapshot (p. 1229).
- You can’t share snapshots that are encrypted with the default AWS managed key. You can only share snapshots that are encrypted with a customer managed key. For more information, see Creating Keys in the AWS Key Management Service Developer Guide.
- You can share only unencrypted snapshots publicly.
- When you share an encrypted snapshot, you must also share the customer managed key used to encrypt the snapshot. For more information, see Share a KMS key (p. 1236).

Share a snapshot

You can share a snapshot using one of the methods described in the section.

Console

To share a snapshot

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.
   - To share the snapshot with one or more AWS accounts, choose Private, enter the AWS account ID (without hyphens) in AWS Account Number, and choose Add Permission. Repeat for any additional AWS accounts.
5. Choose Save.

AWS CLI

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 share a snapshot publicly

Use one of the following commands.

- **modify-snapshot-attribute (AWS CLI)**

  For --attribute, specify createVolumePermission. For --operation-type, specify add. For --group-names, specify all.

  ```bash
  # aws ec2 modify-snapshot-attribute --snapshot-id 1234567890abcdef0 --attribute createVolumePermission --operation-type add --group-names all
  ```
• **Edit-EC2SnapshotAttribute** (AWS Tools for Windows PowerShell)
  
  For **-Attribute**, specify CreateVolumePermission. For **-OperationType**, specify Add. For **-GroupName**, specify all.

  
  ```powershell
  PS C:\> Edit-EC2SnapshotAttribute -SnapshotId 1234567890abcdef0 -Attribute CreateVolumePermission -OperationType Add -GroupName all
  ```

**To share a snapshot privately**

Use one of the following commands.

• **modify-snapshot-attribute** (AWS CLI)
  
  For **--attribute**, specify createVolumePermission. For **--operation-type**, specify add. For **--user-ids**, specify the 12-digit IDs of the AWS accounts with which to share the snapshots.

  ```
  # aws ec2 modify-snapshot-attribute --snapshot-id 1234567890abcdef0 --attribute createVolumePermission --operation-type add --user-ids 123456789012
  ```

• **Edit-EC2SnapshotAttribute** (AWS Tools for Windows PowerShell)
  
  For **-Attribute**, specify CreateVolumePermission. For **-OperationType**, specify Add. For **-UserId**, specify the 12-digit IDs of the AWS accounts with which to share the snapshots.

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

**Share a KMS key**

When you share an encrypted snapshot, you must also share the customer managed key used to encrypt the snapshot. You can apply cross-account permissions to a customer managed key either when it is created or at a later time.

Users of your shared customer managed key who are accessing encrypted snapshots must be granted permissions to perform the following actions on the key:

• kms:DescribeKey
• kms:CreateGrant
• kms:GenerateDataKey
• kms:ReEncrypt
• kms:Decrypt

For more information about controlling access to a customer managed key, see [Using key policies in AWS KMS](https://docs.aws.amazon.com/kms/latest/developerguide/key-policy.html) in the *AWS Key Management Service Developer Guide*.

**To share customer managed key using the AWS KMS console**

2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. Choose **Customer managed keys** in the navigation pane.
4. In the **Alias** column, choose the alias (text link) of the customer managed key that you used to encrypt the snapshot. The key details open in a new page.
5. In the **Key policy** section, you see either the **policy view** or the **default view**. The policy view displays the key policy document. The default view displays sections for **Key administrators**, **Key deletion**, **Key Use**, and **Other AWS accounts**. The default view displays if you created the policy in the console and have not customized it. If the default view is not available, you'll need to manually edit the policy in the policy view. For more information, see Viewing a Key Policy (Console) in the AWS Key Management Service Developer Guide.

Use either the policy view or the default view, depending on which view you can access, to add one or more AWS account IDs to the policy, as follows:

- **(Policy view)** Choose **Edit**. Add one or more AWS account IDs to the following statements: "Allow use of the key" and "Allow attachment of persistent resources". Choose **Save changes**. In the following example, the AWS account ID 444455556666 is added to the policy.

  ```json
  {  
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
  "AWS": [   
  "arn:aws:iam::111122223333:user/KeyUser",
  "arn:aws:iam::444455556666:root"
  ]},
  "Action": [
  "kms:Encrypt",
  "kms:Decrypt",
  "kms:ReEncrypt:*",
  "kms:GenerateDataKey*",
  "kms:DescribeKey",
  
  ],
  "Resource": "*
    
  },
  {
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {
  "AWS": [   
  "arn:aws:iam::111122223333:user/KeyUser",
  "arn:aws:iam::444455556666:root"
  ]},
  "Action": [
  "kms:CreateGrant",
  "kms:ListGrants",
  "kms:RevokeGrant"
  ],
  "Resource": "*
    
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
  }
  ```

- **(Default view)** Scroll down to **Other AWS accounts**. Choose **Add other AWS accounts** and enter the AWS account ID as prompted. To add another account, choose **Add another AWS account** and enter the AWS account ID. When you have added all AWS accounts, choose **Save changes**.

**View snapshots that are shared with you**

You can view snapshots that are shared with you using one of the following methods.

**Console**

**To view shared snapshots using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Snapshots**.
3. Filter the listed snapshots. In the top-left corner of the screen, choose one of the following options:

- **Private snapshots** — To view only snapshots that are shared with you privately.
- **Public snapshots** — To view only snapshots that are shared with you publicly.

AWS CLI

**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)*

**Use snapshots that are shared with you**

**To use a shared unencrypted snapshot**

Locate the shared snapshot by ID or description. For more information, see View snapshots that are shared with you *(p. 1237)*. You can use this snapshot as you would any other snapshot that you own in your account. For example, you can create a volume from the snapshot or copy it to a different Region.

**To use a shared encrypted snapshot**

Locate the shared snapshot by ID or description. For more information, see View snapshots that are shared with you *(p. 1237)*. Create a copy of the shared snapshot in your account, and encrypt the copy with a KMS key that you own. You can then use the copy to create volumes or you can copy it to different Regions.

**Determine the use of snapshots that you share**

You can use AWS CloudTrail to monitor whether a snapshot that you have shared with others is copied or used to create a volume. The following events are logged in CloudTrail:

- **SharedSnapshotCopyInitiated** — A shared snapshot is being copied.
- **SharedSnapshotVolumeCreated** — A shared snapshot is being used to create a volume.

For more information about using CloudTrail, see Log Amazon EC2 and Amazon EBS API calls with AWS CloudTrail *(p. 876)*.

**Amazon EBS local snapshots on Outposts**

Amazon EBS snapshots are a point-in-time copy of your EBS volumes.

By default, snapshots of EBS volumes on an Outpost are stored in Amazon S3 in the Region of the Outpost. You can also use Amazon EBS local snapshots on Outposts to store snapshots of volumes on an Outpost locally in Amazon S3 on the Outpost itself. This ensures that the snapshot data resides on the Outpost, and on your premises. In addition, you can use AWS Identity and Access Management (IAM) policies and permissions to set up data residency enforcement policies to ensure that snapshot data does not leave the Outpost. This is especially useful if you reside in a country or region that is not yet served by an AWS Region and that has data residency requirements.

This topic provides information about working with Amazon EBS local snapshots on Outposts. For more information about Amazon EBS snapshots and about working with snapshots in an AWS Region, see Amazon EBS snapshots *(p. 1207)*.
Frequently asked questions

1. What are local snapshots?

By default, Amazon EBS snapshots of volumes on an Outpost are stored in Amazon S3 in the Region of the Outpost. If the Outpost is provisioned with Amazon S3 on Outposts, you can choose to store the snapshots locally on the Outpost itself. Local snapshots are incremental, which means that only the blocks of the volume that have changed after your most recent snapshot are saved. You can use these snapshots to restore a volume on the same Outpost as the snapshot at any time. For more information about Amazon EBS snapshots, see Amazon EBS snapshots (p. 1207).

2. Why should I use local snapshots?

Snapshots are a convenient way of backing up your data. With local snapshots, all of your snapshot data is stored locally on the Outpost. This means that it does not leave your premises. This is especially useful if you reside in a country or region that is not yet served by an AWS Region and that has residency requirements.

Additionally, using local snapshots can help to reduce the bandwidth used for communication between the Region and the Outpost in bandwidth constrained environments.

3. How do I enforce snapshot data residency on Outposts?

You can use AWS Identity and Access Management (IAM) policies to control the permissions that principals (AWS accounts, IAM users, and IAM roles) have when working with local snapshots and to enforce data residency. You can create a policy that prevents principals from creating snapshots from Outpost volumes and instances and storing the snapshots in an AWS Region. Currently, copying snapshots and images from an Outpost to a Region is not supported. For more information, see Controlling access with IAM (p. 1241).

4. Are multi-volume, crash-consistent local snapshots supported?

Yes, you can create multi-volume, crash-consistent local snapshots from instances on an Outpost.

5. How do I create local snapshots?

You can create snapshots manually using the AWS Command Line Interface (AWS CLI) or the Amazon EC2 console. For more information see, Working with local snapshots (p. 1242). You can also automate the lifecycle of local snapshots using Amazon Data Lifecycle Manager. For more information see, Automate snapshots on an Outpost (p. 1247).

6. Can I create, use, or delete local snapshots if my Outpost loses connectivity to its Region?

No. The Outpost must have connectivity with its Region as the Region provides the access, authorization, logging, and monitoring services that are critical for your snapshots' health. If there is no connectivity, you can't create new local snapshots, create volumes or launch instances from existing local snapshots, or delete local snapshots.

7. How quickly is Amazon S3 storage capacity made available after deleting local snapshots?

Amazon S3 storage capacity becomes available within 72 hours after deleting local snapshots and the volumes that reference them.
8. How can I ensure that I do not run out of Amazon S3 capacity on my Outpost?

We recommend that you use Amazon CloudWatch alarms to monitor your Amazon S3 storage capacity, and delete snapshots and volumes that you no longer need to avoid running out of storage capacity. If you are using Amazon Data Lifecycle Manager to automate the lifecycle of local snapshots, ensure that your snapshot retention policies do not retain snapshots for longer than is needed.

9. Can I use local snapshots and AMIs backed by local snapshots with Spot Instances and Spot Fleet?

No, you can't use local snapshots or AMIs backed by local snapshots to launch Spot Instances or a Spot Fleet.

10. Can I use local snapshots and AMIs backed by local snapshots with Amazon EC2 Auto Scaling?

Yes, you can use local snapshots and AMIs backed by local snapshots to launch Auto Scaling groups in a subnet that is on the same Outpost as the snapshots. The Amazon EC2 Auto Scaling group service-linked role must have permission to use the KMS key used to encrypt the snapshots.

You can't use local snapshots or AMIs backed by local snapshots to launch Auto Scaling groups in an AWS Region.

Prerequisites

To store snapshots on an Outpost, you must have an Outpost that is provisioned with Amazon S3 on Outposts. For more information about Amazon S3 on Outposts, see Using Amazon S3 on Outposts in the Amazon Simple Storage Service Developer Guide.

Considerations

Keep the following in mind when working with local snapshots.

- Outposts must have connectivity to their AWS Region to use local snapshots.
- Snapshot metadata is stored in the AWS Region associated with the Outpost. This does not include any snapshot data.
- Snapshots stored on Outposts are encrypted by default. Unencrypted snapshots are not supported. Snapshots that are created on an Outpost and snapshots that are copied to an Outpost are encrypted using the default KMS key for the Region or a different KMS key that you specify at the time of the request.
- When you create a volume on an Outpost from a local snapshot, you cannot re-encrypt the volume using a different KMS key. Volumes created from local snapshots must be encrypted using the same KMS key as the source snapshot.
- After you delete local snapshots from an Outpost, the Amazon S3 storage capacity used by the deleted snapshots becomes available within 72 hours. For more information, see Delete local snapshots (p. 1247).
- You can't export local snapshots from an Outpost.
- You can't enable fast snapshot restore for local snapshots.
- EBS direct APIs are not supported with local snapshots.
- You can't copy local snapshots or AMIs from an Outpost to an AWS Region, from one Outpost to another, or within an Outpost. However, you can copy snapshots from an AWS Region to an Outpost. For more information, see Copy snapshots from an AWS Region to an Outpost (p. 1245).
- When copying a snapshot from an AWS region to an Outpost, the data is transferred over the service link. Copying multiple snapshots simultaneously could impact other services running on the Outpost.
- You can't share local snapshots.
- You must use IAM policies to ensure that your data residency requirements are met. For more information, see Controlling access with IAM (p. 1241).
• Local snapshots are incremental backups. Only the blocks in the volume that have changed after your most recent snapshot are saved. Each local snapshot contains all of the information that is needed to restore your data (from the moment when the snapshot was taken) to a new EBS volume. For more information, see How incremental snapshots work (p. 1208).

• You can't use IAM policies to enforce data residency for CopySnapshot and CopyImage actions.

**Controlling access with IAM**

You can use AWS Identity and Access Management (IAM) policies to control the permissions that principals (AWS accounts, IAM users, and IAM roles) have when working with local snapshots. The following are example policies that you can use to grant or deny permission to perform specific actions with local snapshots.

**Important**

Copying snapshots and images from an Outpost to a Region is currently not supported. As a result, you currently can't use IAM policies to enforce data residency for CopySnapshot and CopyImage actions.

**Topics**

- Enforce data residency for snapshots (p. 1241)
- Prevent principals from deleting local snapshots (p. 1242)

**Enforce data residency for snapshots**

The following example policy prevents all principals from creating snapshots from volumes and instances on Outpost arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef and storing the snapshot data in an AWS Region. Principals can still create local snapshots. This policy ensures that all snapshots remain on the Outpost.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots"
            ],
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "ec2:SourceOutpostArn": "arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef",
                    "Null": true
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots"
            ],
            "Resource": "*"
        }
    ]
}
```
Prevent principals from deleting local snapshots

The following example policy prevents all principals from deleting local snapshots that are stored on Outpost arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "ec2:DeleteSnapshot"
      ],
      "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
      "Condition": {
        "StringEquals": {
          "ec2:OutpostArn": "arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteSnapshot"
      ],
      "Resource": "*"
    }
  ]
}
```

Working with local snapshots

The following sections explain how to use local snapshots.

**Topics**

- Rules for storing snapshots (p. 1242)
- Create local snapshots from volumes on an Outpost (p. 1243)
- Create multi-volume local snapshots from instances on an Outpost (p. 1244)
- Create AMIs from local snapshots (p. 1244)
- Copy snapshots from an AWS Region to an Outpost (p. 1245)
- Copy AMIs from an AWS Region to an Outpost (p. 1246)
- Create volumes from local snapshots (p. 1247)
- Launch instances from AMIs backed by local snapshots (p. 1247)
- Delete local snapshots (p. 1247)
- Automate snapshots on an Outpost (p. 1247)

Rules for storing snapshots

The following rules apply to snapshot storage:

- If the most recent snapshot of a volume is stored on an Outpost, then all successive snapshots must be stored on the same Outpost.
- If the most recent snapshot of a volume is stored in an AWS Region, then all successive snapshots must be stored in the same Region. To start creating local snapshots from that volume, do the following:
  1. Create a snapshot of the volume in the AWS Region.
2. Copy the snapshot to the Outpost from the AWS Region.
3. Create a new volume from the local snapshot.
4. Attach the volume to an instance on the Outpost.

For the new volume on the Outpost, the next snapshot can be stored on the Outpost or in the AWS Region. All successive snapshots must then be stored in that same location.

- Local snapshots, including snapshots created on an Outpost and snapshots copied to an Outpost from an AWS Region, can be used only to create volumes on the same Outpost.
- If you create a volume on an Outpost from a snapshot in a Region, then all successive snapshots of that new volume must be in the same Region.
- If you create a volume on an Outpost from a local snapshot, then all successive snapshots of that new volume must be on the same Outpost.

Create local snapshots from volumes on an Outpost

You can create local snapshots from volumes on your Outpost. You can choose to store the snapshots on the same Outpost as the source volume, or in the Region for the Outpost.

Local snapshots can be used to create volumes on the same Outpost only.

You can create local snapshots from volumes on an Outpost using one of the following methods.

Console

**To create local snapshots from volumes on an Outpost**

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

1. In the navigation pane, choose **Volumes**.
2. Select the volume on the Outpost, and choose **Actions, Create Snapshot**.
3. (Optional) For **Description**, enter a brief description for the snapshot.
4. For **Snapshot destination**, choose **AWS Outpost**. The snapshot will be created on the same Outpost as the source volume. The **Outpost ARN** field shows the Amazon Resource Name (ARN) of the destination Outpost.
5. (Optional) Choose **Add Tag** to add tags to your snapshot. For each tag, provide a tag key and a tag value.
6. Choose **Create Snapshot**.

Command line

**To create local snapshots from volumes on an Outpost**

Use the `create-snapshot` command. Specify the ID of the volume from which to create the snapshot, and the ARN of the destination Outpost on which to store the snapshot. If you omit the Outpost ARN, the snapshot is stored in the AWS Region for the Outpost.

For example, the following command creates a local snapshot of volume `vol-1234567890abcdef0`, and stores the snapshot on Outpost `arn:aws:outposts:us-east-1:1234567890abcdef0:outpost/1234567890abcdef0`.

```
$ aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --outpost-arn arn:aws:outposts:us-east-1:1234567890abcdef0:outpost/1234567890abcdef0 --description "single volume local snapshot"
```

1243
Create multi-volume local snapshots from instances on an Outpost

You can create crash-consistent multi-volume local snapshots from instances on your Outpost. You can choose to store the snapshots on the same Outpost as the source instance, or in the Region for the Outpost.

Multi-volume local snapshots can be used to create volumes on the same Outpost only.

You can create multi-volume local snapshots from instances on an Outpost using one of the following methods.

Console

To create multi-volume local snapshots from instances on an Outpost

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

1. In the navigation pane, choose Snapshots.
2. Choose Create Snapshot.
3. For Select resource type, choose Instance.
4. For Instance ID, select the instance on the Outpost from which to create the snapshots.
5. (Optional) For Description, enter a brief description for the snapshots.
6. For Snapshot destination, choose AWS Outpost. The snapshots will be created on the same Outpost as the source instance. The Outpost ARN shows the ARN of the destination Outpost.
7. (Optional) To exclude the root volume from being snapshotted, select Exclude root volume.
8. (Optional) To automatically copy tags from the source volume to the snapshots, select Copy tags from volume. This sets snapshot metadata—such as access policies, attachment information, and cost allocation—to match the source volume.
9. (Optional) Choose Add Tag to add tags to your snapshot. For each tag, provide a tag key and a tag value.
10. Choose Create Snapshot.

During snapshot creation, the snapshots are managed together. If one of the snapshots in the volume set fails, the other snapshots in the volume set are moved to error status.

Command line

To create multi-volume local snapshots from instances on an Outpost

Use the create-snapshots command. Specify the ID of the instance from which to create the snapshots, and the ARN of the destination Outpost on which to store the snapshots. If you omit the Outpost ARN, the snapshots are stored in the AWS Region for the Outpost.

For example, the following command creates snapshots of the volumes attached to instance i-1234567890abcdef0 and stores the snapshots on Outpost arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0.

```
$ aws ec2 create-snapshots --instance-specification InstanceId=i-1234567890abcdef0 --outpost-arn arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0 --description "multi-volume local snapshots"
```

Create AMIs from local snapshots

You can create Amazon Machine Images (AMIs) using a combination of local snapshots and snapshots that are stored in the Region of the Outpost. For example, if you have an Outpost in us-east-1, you can
create an AMI with data volumes that are backed by local snapshots on that Outpost, and a root volume that is backed by a snapshot in the us-east-1 Region.

Note

- You can't create AMIs that include backing snapshots stored across multiple Outposts.
- You can’t currently create AMIs directly from instances on an Outposts using CreateImage API or the Amazon EC2 console for Outposts that are enabled with Amazon S3 on Outposts.
- AMIs that are backed by local snapshots can be used to launch instances on the same Outpost only.

To create an AMI on an Outpost from snapshots in a Region

1. Copy the snapshots from the Region to the Outpost. For more information, see Copy snapshots from an AWS Region to an Outpost (p. 1245).
2. Use the Amazon EC2 console or the register-image command to create the AMI using the snapshot copies on the Outpost. For more information, see Creating an AMI from a snapshot.

To create an AMI on an Outpost from an instance on an Outpost

1. Create snapshots from the instance on the Outpost and store the snapshots on the Outpost. For more information, see Create multi-volume local snapshots from instances on an Outpost (p. 1244).
2. Use the Amazon EC2 console or the register-image command to create the AMI using the local snapshots. For more information, see Creating an AMI from a snapshot.

To create an AMI in a Region from an instance on an Outpost

1. Create snapshots from the instance on the Outpost and store the snapshots in the Region. For more information, see Create local snapshots from volumes on an Outpost (p. 1243) or Create multi-volume local snapshots from instances on an Outpost (p. 1244).
2. Use the Amazon EC2 console or the register-image command to create the AMI using the snapshot copies in the Region. For more information, see Creating an AMI from a snapshot.

Copy snapshots from an AWS Region to an Outpost

You can copy snapshots from an AWS Region to an Outpost. You can do this only if the snapshots are in the Region for the Outpost. If the snapshots are in a different Region, you must first copy the snapshot to the Region for the Outpost, and then copy it from that Region to the Outpost.

Note

You can't copy local snapshots from an Outpost to a Region, from one Outpost to another, or within the same Outpost.

You can copy snapshots from a Region to an Outpost using one of the following methods.

Console

To copy a snapshot from an AWS Region to an Outpost

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

1. In the navigation pane, choose Snapshots.
2. Select the snapshot in the Region, and choose Actions, Copy.
3. For Destination Region, choose the Region for the destination Outpost.
4. For **Snapshot Destination**, choose **AWS Outpost**.

   The **Snapshot Destination** field only appears if you have Outposts in the selected destination Region. If the field does not appear, you do not have any Outposts in the selected destination Region.

5. For **Destination Outpost ARN**, enter the ARN of the Outpost to which to copy the snapshot.
6. (Optional) For **Description**, enter a brief description of the copied snapshot.
7. Encryption is enabled by default for the snapshot copy. Encryption cannot be disabled. For **KMS key**, choose the KMS key to use.
8. Choose **Copy**.

**Command line**

**To copy a snapshot from a Region to an Outpost**

Use the `copy-snapshot` command. Specify the ID of the snapshot to copy, the Region from which to copy the snapshot, and the ARN of the destination Outpost.

For example, the following command copies snapshot `snap-1234567890abcdef0` from the `us-east-1` Region to Outpost `arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0`.

```
$ aws ec2 copy-snapshot --source-region us-east-1 --source-snapshot-id snap-1234567890abcdef0 --destination-outpost-arn arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0 --description "Local snapshot copy"
```

**Copy AMIs from an AWS Region to an Outpost**

You can copy AMIs from an AWS Region to an Outpost. When you copy an AMI from a Region to an Outpost, all of the snapshots associated with the AMI are copied from the Region to the Outpost.

You can copy an AMI from a Region to an Outpost only if the snapshots associated with the AMI are in the Region for the Outpost. If the snapshots are in a different Region, you must first copy the AMI to the Region for the Outpost, and then copy it from that Region to the Outpost.

**Note**

You can't copy an AMI from an Outpost to a Region, from one Outpost to another, or within an Outpost.

You can copy AMIs from a Region to an Outpost using the AWS CLI only.

**Command line**

**To copy an AMI from a Region to an Outpost**

Use the `copy-image` command. Specify the ID of the AMI to copy, the source Region, and the ARN of the destination Outpost.

For example, the following command copies AMI `ami-1234567890abcdef0` from the `us-east-1` Region to Outpost `arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0`.

```
$ aws ec2 copy-image --source-region us-east-1 --source-image-id ami-1234567890abcdef0 --name "Local AMI copy" --destination-outpost-arn arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0
```
Create volumes from local snapshots

You can create volumes on Outposts from local snapshots. Volumes must be created on the same Outpost as the source snapshots. You cannot use local snapshots to create volumes in the Region for the Outpost.

When you create a volume from a local snapshot, you cannot re-encrypt the volume using different KMS key. Volumes created from local snapshots must be encrypted using the same KMS key as the source snapshot.

For more information, see Create a volume from a snapshot (p. 1185).

Launch instances from AMIs backed by local snapshots

You can launch instances from AMIs that are backed by local snapshots. You must launch Instances on the same Outpost as the source AMI. For more information, see Launch an instance on your Outpost in the AWS Outposts User Guide.

Delete local snapshots

You can delete local snapshots from an Outpost. After you delete a snapshot from an Outpost, the Amazon S3 storage capacity used by the deleted snapshot becomes available within 72 hours after deleting the snapshot and the volumes that reference that snapshot.

Because Amazon S3 storage capacity does not become available immediately, we recommend that you use Amazon CloudWatch alarms to monitor your Amazon S3 storage capacity. Delete snapshots and volumes that you no longer need to avoid running out of storage capacity.

For more information about deleting snapshots, see Delete a snapshot (p. 1228).

Automate snapshots on an Outpost

You can create Amazon Data Lifecycle Manager snapshot lifecycle policies that automatically create, copy, retain, and delete snapshots of your volumes and instances on an Outpost. You can choose whether to store the snapshots in a Region or whether to store them locally on an Outpost. Additionally, you can automatically copy snapshots that are created and stored in an AWS Region to an Outpost.

The following table shows provides and Overview of the supported features.

<table>
<thead>
<tr>
<th>Resource location</th>
<th>Snapshot destination</th>
<th>Cross-region copy</th>
<th>Fast snapshot restore</th>
<th>Cross-account sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>To Region</td>
<td>To Outpost</td>
<td></td>
</tr>
<tr>
<td>Region</td>
<td>Region</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Outpost</td>
<td>Region</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Outpost</td>
<td>Outpost</td>
<td>×</td>
<td>×</td>
<td>✓</td>
</tr>
</tbody>
</table>

Considerations

- Only Amazon EBS snapshot lifecycle policies are currently supported. EBS-backed AMI policies and Cross-account sharing event policies are not supported.
- If a policy manages snapshots for volumes or instances in a Region, then snapshots are created in the same Region as the source resource.
- If a policy manages snapshots for volumes or instances on an Outpost, then snapshots can be created on the source Outpost, or in the Region for that Outpost.
- A single policy can't manage both snapshots in a Region and snapshots on an Outpost. If you need to automate snapshots in a Region and on an Outpost, you must create separate policies.
• Fast snapshot restore is not supported for snapshots created on an Outpost, or for snapshots copied to
an Outpost.
• Cross-account sharing is not supported for snapshots created on an Outpost.

For more information about creating a snapshot lifecycle that manages local snapshots, see Automating
snapshot lifecycles (p. 1277).

Use EBS direct APIs to access the contents of an EBS snapshot

You can use the Amazon Elastic Block Store (Amazon EBS) direct APIs to create EBS snapshots, write
data directly to your snapshots, read data on your snapshots, and identify the differences or changes
between two snapshots. If you’re an independent software vendor (ISV) who offers backup services for
Amazon EBS, the EBS direct APIs make it more efficient and cost-effective to track incremental changes
on your EBS volumes through snapshots. This can be done without having to create new volumes
from snapshots, and then use Amazon Elastic Compute Cloud (Amazon EC2) instances to compare the
differences.

You can create incremental snapshots directly from data on-premises into EBS volumes and the cloud
to use for quick disaster recovery. With the ability to write and read snapshots, you can write your on-
premises data to an EBS snapshot during a disaster. Then after recovery, you can restore it back to AWS
or on-premises from the snapshot. You no longer need to build and maintain complex mechanisms to
copy data to and from Amazon EBS.

This user guide provides an overview of the elements that make up the EBS direct APIs, and examples of
how to use them effectively. For more information about the actions, data types, parameters, and errors
of the APIs, see the EBS direct APIs reference. For more information about the supported AWS Regions,
endpoints, and service quotas for the EBS direct APIs, see Amazon EBS Endpoints and Quotas in the AWS
General Reference.

Contents
• Understand the EBS direct APIs (p. 1248)
• Permissions for IAM users (p. 1251)
• Use encryption (p. 1255)
• Use Signature Version 4 signing (p. 1255)
• Use checksums (p. 1256)
• Work with the EBS direct APIs using the API or AWS SDKs (p. 1256)
• Work with the EBS direct APIs using the command line (p. 1261)
• Optimize performance (p. 1264)
• Frequently asked questions (p. 1264)
• Log API Calls for EBS direct APIs with AWS CloudTrail (p. 1265)
• EBS direct APIs and interface VPC endpoints (p. 1271)
• Idempotency for StartSnapshot API (p. 1271)

Understand the EBS direct APIs

The following are the key elements that you should understand before getting started with the EBS
direct APIs.

Pricing

The price that you pay to use the EBS direct APIs depends on the requests you make. For more
information, see Amazon EBS pricing.
Snapshots

Snapshots are the primary means to back up data from your EBS volumes. With the EBS direct APIs, you can also back up data from your on-premises disks to snapshots. To save storage costs, successive snapshots are incremental, containing only the volume data that changed since the previous snapshot. For more information, see Amazon EBS snapshots (p. 1207).

**Note**
Public snapshots are not supported by the EBS direct APIs.

Blocks

A block is a fragment of data within a snapshot. Each snapshot can contain thousands of blocks. All blocks in a snapshot are of a fixed size.

Block indexes

A block index is the offset position of a block within a snapshot, and it is used to identify the block. Multiply the BlockIndex value with the BlockSize value (BlockIndex * BlockSize) to identify the logical offset of the data in the logical volume.

Block tokens

A block token is the identifying hash of a block within a snapshot, and it is used to locate the block data. Block tokens returned by EBS direct APIs are temporary. They change on the expiry timestamp specified for them, or if you run another ListSnapshotBlocks or ListChangedBlocks request for the same snapshot.

Checksum

A checksum is a small-sized datum derived from a block of data for the purpose of detecting errors that were introduced during its transmission or storage. The EBS direct APIs use checksums to validate data integrity. When you read data from an EBS snapshot, the service provides Base64-encoded SHA256 checksums for each block of data transmitted, which you can use for validation. When you write data to an EBS snapshot, you must provide a Base64 encoded SHA256 checksum for each block of data transmitted. The service validates the data received using the checksum provided. For more information, see Use checksums (p. 1256) later in this guide.

Encryption

Encryption protects your data by converting it into unreadable code that can be deciphered only by people who have access to the KMS key used to encrypt it. You can use the EBS direct APIs to read and write encrypted snapshots, but there are some limitations. For more information, see Use encryption (p. 1255) later in this guide.

API actions

The EBS direct APIs consists of six actions. There are three read actions and three write actions. The read actions are ListSnapshotBlocks, ListChangedBlocks, and GetSnapshotBlock. The write actions are StartSnapshot, PutSnapshotBlock, and CompleteSnapshot. These actions are described in the following sections.

**List snapshot blocks**

The ListSnapshotBlocks action returns the block indexes and block tokens of blocks in the specified snapshot.

**List changed blocks**

The ListChangedBlocks action returns the block indexes and block tokens of blocks that are different between two specified snapshots of the same volume and snapshot lineage.
Get snapshot block

The GetSnapshotBlock action returns the data in a block for the specified snapshot ID, block index, and block token.

Start snapshot

The StartSnapshot action starts a snapshot, either as an incremental snapshot of an existing one or as a new snapshot. The started snapshot remains in a pending state until it is completed using the CompleteSnapshot action.

Put snapshot block

The PutSnapshotBlock action adds data to a started snapshot in the form of individual blocks. You must specify a Base64-encoded SHA256 checksum for the block of data transmitted. The service validates the checksum after the transmission is completed. The request fails if the checksum computed by the service doesn't match what you specified.

Complete snapshot

The CompleteSnapshot action completes a started snapshot that is in a pending state. The snapshot is then changed to a completed state.

Use the EBS direct APIs to read snapshots

The following steps describe how to use the EBS direct APIs to read snapshots:

1. Use the ListSnapshotBlocks action to view all block indexes and block tokens of blocks in a snapshot. Or use the ListChangedBlocks action to view only the block indexes and block tokens of blocks that are different between two snapshots of the same volume and snapshot lineage. These actions help you identify the block tokens and block indexes of blocks for which you might want to get data.
2. Use the GetSnapshotBlock action, and specify the block index and block token of the block for which you want to get data.

For examples of how to run these actions, see the Work with the EBS direct APIs using the API or AWS SDKs (p. 1256) and Work with the EBS direct APIs using the command line (p. 1261) sections later in this guide.

Use the EBS direct APIs to write incremental snapshots

The following steps describe how to use the EBS direct APIs to write incremental snapshots:

1. Use the StartSnapshot action and specify a parent snapshot ID to start a snapshot as an incremental snapshot of an existing one, or omit the parent snapshot ID to start a new snapshot. This action returns the new snapshot ID, which is in a pending state.
2. Use the PutSnapshotBlock action and specify the ID of the pending snapshot to add data to it in the form of individual blocks. You must specify a Base64-encoded SHA256 checksum for the block of data transmitted. The service computes the checksum of the data received and validates it with the checksum that you specified. The action fails if the checksums don't match.
3. When you're done adding data to the pending snapshot, use the CompleteSnapshot action to start an asynchronous workflow that seals the snapshot and moves it to a completed state.

Repeat these steps to create a new, incremental snapshot using the previously created snapshot as the parent.

For example, in the following diagram, snapshot A is the first new snapshot started. Snapshot A is used as the parent snapshot to start snapshot B. Snapshot B is used as the parent snapshot to start and create snapshot C. Snapshots A, B, and C are incremental snapshots. Snapshot A is used to create EBS volume
1. Snapshot D is created from EBS volume 1. Snapshot D is an incremental snapshot of A; it is not an incremental snapshot of B or C.

Permissions for IAM users

An AWS Identity and Access Management (IAM) user must have the following policies to use the EBS direct APIs. For more information, see Changing Permissions for an IAM User.

Be cautious when assigning the following policies to IAM users. By assigning these policies, you might give access to a user who is denied access to the same resource through the Amazon EC2 APIs, such as the CopySnapshot or CreateVolume actions.

Permissions to read snapshots

The following policy allows the read EBS direct APIs to be used on all snapshots in a specific AWS Region. In the policy, replace <Region> with the Region of the snapshot.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:ListSnapshotBlocks",
                "ebs:ListChangedBlocks",
                "ebs:GetSnapshotBlock"
            ],
            "Resource": "arn:aws:ec2:<Region>::snapshot/*"
        }
    ]
}
```

The following policy allows the read EBS direct APIs to be used on snapshots with a specific key-value tag. In the policy, replace <Key> with the key value of the tag, and <Value> with the value of the tag.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:ListSnapshotBlocks",
                "ebs:ListChangedBlocks",
                "ebs:GetSnapshotBlock"
            ],
            "Resource": "arn:aws:ec2:<Region>::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "ebs:SnapshotTag.<Key>:<Value>": "true"
                }
            }
        }
    ]
}
```
The following policy allows all of the read EBS direct APIs to be used on all snapshots in the account only within a specific time range. This policy authorizes use of the EBS direct APIs based on the aws:CurrentTime global condition key. In the policy, be sure to replace the date and time range shown with the date and time range for your policy.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ebs:ListSnapshotBlocks",
        "ebs:ListChangedBlocks",
        "ebs:GetSnapshotBlock"
      ],
      "Resource": "arn:aws:ec2:*::snapshot/*",
      "Condition": {
        "DateGreaterThan": {
          "aws:CurrentTime": "2018-05-29T00:00:00Z"
        },
        "DateLessThan": {
          "aws:CurrentTime": "2020-05-29T23:59:59Z"
        }
      }
    }
  ]
}
```

The following policy grants access to decrypt an encrypted snapshot using a specific KMS key. It grants access to encrypt new snapshots using the default KMS key ID for EBS snapshots. It also provides the ability to determine if encrypt by default is enabled on the account. In the policy, replace <Region> with the Region of the KMS key, <AccountId> with the ID of the AWS account of the KMS key, and <KeyId> with the ID of the KMS key used to encrypt the snapshot that you want to read with the EBS direct APIs.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Sign",
        "kms:Unsign",
        "kms:PutKeyPolicy",
        "kms:GetKeyPolicy",
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:DescribeKey",
        "kms:UpdateKeyDescription",
        "kms:TagsKey"
      ],
      "Resource": "arn:aws:kms:<Region>:<AccountId>:key/<KeyId>",
      "Condition": {
        "StringLike": {
          "aws:PrincipalTag": {
            "kms:IsDefault": true
          }
        }
      }
    }
  ]
}
```
For more information, see Changing Permissions for an IAM User in the IAM User Guide.

Permissions to write snapshots

The following policy allows the write EBS direct APIs to be used on all snapshots in a specific AWS Region. In the policy, replace <Region> with the Region of the snapshot.

```
{  
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ebs:StartSnapshot",
        "ebs:PutSnapshotBlock",
        "ebs:CompleteSnapshot"
      ],
      "Resource": "arn:aws:ec2:<Region>::snapshot/*"
    }
  ]
}
```

The following policy allows the write EBS direct APIs to be used on snapshots with a specific key-value tag. In the policy, replace <Key> with the key value of the tag, and <Value> with the value of the tag.

```
{  
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ebs:StartSnapshot",
        "ebs:PutSnapshotBlock",
        "ebs:CompleteSnapshot"
      ],
      "Resource": "arn:aws:ec2:*::snapshot/**",
      "Condition": {
        "StringEqualsIgnoreCase": {
          "aws:ResourceTag/<Key>": "<Value>"
        }
      }
    }
  ]
}
```

The following policy allows all of the EBS direct APIs to be used. It also allows the StartSnapshot action only if a parent snapshot ID is specified. Therefore, this policy blocks the ability to start new snapshots without using a parent snapshot.

```
{  
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ['
        "ebs:StartSnapshot",
        "ebs:PutSnapshotBlock",
        "ebs:CompleteSnapshot"
      ],
      "Resource": "arn:aws:ec2:*::snapshot/**",
      "Condition": {
        "StringEqualsIgnoreCase": {
          "aws:ResourceTag/ParentSnapshotId": "<ParentSnapshotId>"
        }
      }
    }
  ]
}
```
The following policy allows all of the EBS direct APIs to be used. It also allows only the user tag key to be created for a new snapshot. This policy also ensures that the user has access to create tags. The StartSnapshot action is the only action that can specify tags.

```json


```

The following policy allows all of the write EBS direct APIs to be used on all snapshots in the account only within a specific time range. This policy authorizes use of the EBS direct APIs based on the aws:CurrentTime global condition key. In the policy, be sure to replace the date and time range shown with the date and time range for your policy.

```json

```

1254
The following policy grants access to decrypt an encrypted snapshot using a specific KMS key. It grants access to encrypt new snapshots using the default KMS key ID for EBS snapshots. It also provides the ability to determine if encrypt by default is enabled on the account. In the policy, replace <Region> with the Region of the KMS key, <AccountId> with the ID of the AWS account of the KMS key, and <KeyId> with the ID of the KMS key used to encrypt the snapshot that you want to read with the EBS direct APIs.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": [
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:GenerateDataKey",
            "kms:GenerateDataKeyWithoutPlaintext",
            "kms:ReEncrypt*",
            "kms:CreateGrant",
            "ec2:CreateTags",
            "kms:DescribeKey",
            "ec2:GetEbsDefaultKmsKeyId",
            "ec2:GetEbsEncryptionByDefault"
         ],
         "Resource": "arn:aws:kms:<Region>:<AccountId>:key/<KeyId>"
      }
   ]
}
```

For more information, see Changing Permissions for an IAM User in the IAM User Guide.

### Use encryption

If Amazon EBS encryption by default is enabled on your AWS account, you cannot start a new snapshot using an un-encrypted parent snapshot. You must first encrypt the parent snapshot by copying it. For more information, see Copy an Amazon EBS snapshot (p. 1229) and Encryption by default (p. 1331).

To start an encrypted snapshot, specify the Amazon Resource Name (ARN) of an KMS key, or specify an encrypted parent snapshot in your StartSnapshot request. If neither are specified, and Amazon EBS encryption by default is enabled on the account, then the default KMS key for the account is used. If no default KMS key has been specified for the account, then the AWS managed key is used.

**Important**

By default, all principals in the account have access to the default AWS managed key, and they can use it for EBS encryption and decryption operations. For more information, see Default KMS key for EBS encryption (p. 1330).

You might need additional IAM permissions to use the EBS direct APIs with encryption. For more information, see the Permissions for IAM users (p. 1251) section earlier in this guide.

### Use Signature Version 4 signing

Signature Version 4 is the process to add authentication information to AWS requests sent by HTTP. For security, most requests to AWS must be signed with an access key, which consists of an access key ID and secret access key. These two keys are commonly referred to as your security credentials. For information about how to obtain credentials for your account, see Understanding and getting your credentials.
If you intend to manually create HTTP requests, you must learn how to sign them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

For more information, see Signing AWS requests with Signature Version 4 in the AWS General Reference.

Use checksums

The GetSnapshotBlock action returns data that is in a block of a snapshot, and the PutSnapshotBlock action adds data to a block in a snapshot. The block data that is transmitted is not signed as part of the Signature Version 4 signing process. As a result, checksums are used to validate the integrity of the data as follows:

- When you use the GetSnapshotBlock action, the response provides a Base64-encoded SHA256 checksum for the block data using the x-amz-Checksum header, and the checksum algorithm using the x-amz-Checksum-Algorithm header. Use the returned checksum to validate the integrity of the data. If the checksum that you generate doesn't match what Amazon EBS provided, you should consider the data not valid and retry your request.

- When you use the PutSnapshotBlock action, your request must provide a Base64-encoded SHA256 checksum for the block data using the x-amz-Checksum header, and the checksum algorithm using the x-amz-Checksum-Algorithm header. The checksum that you provide is validated against a checksum generated by Amazon EBS to validate the integrity of the data. If the checksums do not correspond, the request fails.

- When you use the CompleteSnapshot action, your request can optionally provide an aggregate Base64-encoded SHA256 checksum for the complete set of data added to the snapshot. Provide the checksum using the x-amz-Checksum header, the checksum algorithm using the x-amz-Checksum-Algorithm header, and the checksum aggregation method using the x-amz-Checksum-Aggregation-Method header. To generate the aggregated checksum using the linear aggregation method, arrange the checksums for each written block in ascending order of their block index, concatenate them to form a single string, and then generate the checksum on the entire string using the SHA256 algorithm.

The checksums in these actions are part of the Signature Version 4 signing process.

Work with the EBS direct APIs using the API or AWS SDKs

The EBS direct APIs Reference provides descriptions and syntax for each of the service's actions and data types. You can also use one of the AWS SDKs to access an API that's tailored to the programming language or platform that you're using. For more information, see AWS SDKs.

The EBS direct APIs require an AWS Signature Version 4 signature. For more information, see Use Signature Version 4 signing (p. 1255).

Use the API to read snapshots

List blocks in a snapshot

The following ListChangedBlocks example request returns the block indexes and block tokens of blocks that are in snapshot snap-0acEXAMPLEcf41648. The startingBlockIndex parameter limits the results to block indexes greater than 1000, and the maxResults parameter limits the results to the first 100 blocks.

```
GET /snapshots/snap-0acEXAMPLEcf41648/blocks?maxResults=100&startingBlockIndex=1000
HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
User-Agent: <User agent parameter>
```
The following example response for the previous request lists the block indexes and block tokens in the snapshot. Use the GetSnapshotBlock action and specify the block index and block token of the block for which you want to get data. The block tokens are valid until the expiry time listed.

```
HTTP/1.1 200 OK
x-amzn-RequestId: d6e5017c-70a8-4539-8830-57f5557f3f27
Content-Type: application/json
Content-Length: 2472
Date: Wed, 17 Jun 2020 23:19:56 GMT
Connection: keep-alive
{
    "BlockSize": 524288,
    "Blocks": [
        {
            "BlockIndex": 0,
            "BlockToken": "AAUBAcuWqOCnDNuKle11s7IX6jp6FYyC/q8oT93913HhvLvA+3JRRSybp/0"
        },
        {
            "BlockIndex": 1536,
            "BlockToken": "AAUBAWudwfmofcrq6GV1w8Km2b8XPIyrgoykTSC6IuNbxKWDY1pPjvnV"
        },
        {
            "BlockIndex": 3072,
            "BlockToken": "AAUBAV7p6pC5fAC7Tok0NCttAn2Zhq27u6YEXZ3MwRe6BkDjeMx6iu6a6tsBt"
        },
        {
            "BlockIndex": 3073,
            "BlockToken": "AAUBAbqt9zpqBUEv02HNAfFaWToOWLPjbsQOlx6JUN/0+1MqLoNtWnX4"
        },
        ...
    ],
    "ExpiryTime": 1.59298379649E9,
    "VolumeSize": 3
}
```

**List blocks that are different between two snapshots**

The following ListChangedBlocks example request returns the block indexes and block tokens of blocks that are different between snapshots snap-0acEXAMPLEcf41648 and snap-0c9EXAMPLE1b30e2f. The startingBlockIndex parameter limits the results to block indexes greater than 0, and the maxResults parameter limits the results to the first 500 blocks.

```
GET /snapshots/snap-0c9EXAMPLE1b30e2f/changedblocks?
firstSnapshotId=snap-0acEXAMPLEcf41648&maxResults=500&startingBlockIndex=0 HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
User-Agent: <User agent parameter>
X-Amz-Date: 20200617T232546Z
Authorization: <Authentication parameter>
```

The following example response for the previous request shows that block indexes 0, 3072, 6002, and 6003 are different between the two snapshots. Additionally, block indexes 6002, and 6003 exist only in the first snapshot ID specified, and not in the second snapshot ID because there is no second block token listed in the response.

Use the GetSnapshotBlock action and specify the block index and block token of the block for which you want to get data. The block tokens are valid until the expiry time listed.
Get block data from a snapshot

The following GetSnapshotBlock example request returns the data in the block index 3072 with block token AAUBARGCaufCqBRZC8tEkPYGGkSv3vqVOj32xKDi3ljDFiytUXBLXYgTmkid, in snapshot snap-0c9EXAMPLE1b30e2f.

GET /snapshots/snap-0c9EXAMPLE1b30e2f/blocks/3072?
blockToken=AAUBARGCaufCqBRZC8tEkPYGGkSv3vqVOj32xKDi3ljDFiytUXBLXYgTmkid HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
User-Agent: <User agent parameter>
X-Amz-Date: 20200617T232838Z
Authorization: <Authentication parameter>

The following example response for the previous request shows the size of the data returned, the checksum to validate the data, and the algorithm used to generate the checksum. The binary data is transmitted in the body of the response and is represented as BlockData in the following example.

HTTP/1.1 200 OK
x-amzn-RequestId: 2d0db2fb-bd88-474d-a137-81c4e57d7b9f
x-amzn-Data-Length: 524288
x-amzn-Checksum: Vc0yY2j3qg8bUL9I6GQuI2orTudrQRBDMIhcy7bdEsw=
Use the API to write incremental snapshots

Start a snapshot

The following StartSnapshot example request starts an 8 GiB snapshot, using snapshot snap-123EXAMPLE1234567 as the parent snapshot. The new snapshot will be an incremental snapshot of the parent snapshot. The snapshot moves to an error state if there are no put or complete requests made for the snapshot within the specified 60 minute timeout period. The 550e8400-e29b-41d4-a716-446655440000 client token ensures idempotency for the request. If the client token is omitted, the AWS SDK automatically generates one for you. For more information about idempotency, see Idempotency for StartSnapshot API (p. 1271).

```
POST /snapshots HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
User-Agent: <User agent parameter>
X-Amz-Date: 20200618T040724Z
Authorization: <Authentication parameter>
{
    "VolumeSize": 8,
    "ParentSnapshot": "snap-123EXAMPLE1234567",
    "ClientToken": "550e8400-e29b-41d4-a716-446655440000",
    "Timeout": 60
}
```

The following example response for the previous request shows the snapshot ID, AWS account ID, status, volume size in GiB, and size of the blocks in the snapshot. The snapshot is started in a pending state. Specify the snapshot ID in a subsequent PutSnapshotBlocks request to write data to the snapshot.

```
HTTP/1.1 201 Created
x-amzn-RequestId: 929e6eb9-7183-405a-9502-5b7da37c1b18
Content-Type: application/json
Content-Length: 181
Date: Thu, 18 Jun 2020 04:07:29 GMT
Connection: keep-alive
{
    "BlockSize": 524288,
    "Description": null,
    "OwnerId": "138695307491",
    "Progress": null,
    "SnapshotId": "snap-052EXAMPLEc85d8dd",
    "StartTime": null,
    "Status": "pending",
    "Tags": null,
    "VolumeSize": 8
}
```

Put data into a snapshot

The following PutSnapshot example request writes 524288 Bytes of data to block index 1000 on snapshot snap-052EXAMPLEc85d8dd. The Base64 encoded
The checksum was generated using the SHA256 algorithm. The data is transmitted in the body of the request and is represented as BlockData in the following example.

```
PUT /snapshots/snap-052EXAMPLEc85d8dd/blocks/1000 HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
x-amz-Data-Length: 524288
x-amz-Checksum: QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUukt0w8DM=
x-amz-Checksum-Algorithm: SHA256
User-Agent: <User agent parameter>
X-Amz-Date: 20200618T042215Z
Authorization: <Authentication parameter>

BlockData
```

The following is example response for the previous request confirms the data length, checksum, and checksum algorithm for the data received by the service.

```
HTTP/1.1 201 Created
x-amzn-RequestId: 64ac797f-7e9c-4ad0-8417-97b77b43c57b
x-amz-Checksum: QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUukt0w8DM=
x-amz-Checksum-Algorithm: SHA256
Content-Type: application/json
Content-Length: 2
Date: Thu, 18 Jun 2020 04:22:12 GMT
Connection: keep-alive

{}
```

**Complete a snapshot**

The following CompleteSnapshot example request completes snapshot snap-052EXAMPLEc85d8dd. The command specifies that 5 blocks were written to the snapshot. The 6D3nmwi5f2F0wlh7xX8qprrJBFzDX8aadOdCA3KCm3c= checksum represents the checksum for the complete set of data written to a snapshot.

```
POST /snapshots/completion/snap-052EXAMPLEc85d8dd HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
x-amz-ChangedBlocksCount: 5
x-amz-Checksum: 6D3nmwi5f2F0wlh7xX8qprrJBFzDX8aadOdCA3KCm3c=
x-amz-Checksum-Algorithm: SHA256
x-amz-Checksum-Aggregation-Method: LINEAR
User-Agent: <User agent parameter>
X-Amz-Date: 20200618T043158Z
Authorization: <Authentication parameter>

The following is an example response for the previous request.

```
HTTP/1.1 202 Accepted
x-amzn-RequestId: 06cba5b5-b731-49de-af40-80333ac3a117
Content-Type: application/json
Content-Length: 20
Date: Thu, 18 Jun 2020 04:31:50 GMT
Connection: keep-alive

{"Status":"pending"}
```
Work with the EBS direct APIs using the command line

The following examples show how to use the EBS direct APIs using the AWS Command Line Interface (AWS CLI). For more information about installing and configuring the AWS CLI, see Installing the AWS CLI and Quickly Configuring the AWS CLI.

Use the AWS CLI to read snapshots

List blocks in a snapshot

The following list-snapshot-blocks example command returns the block indexes and block tokens of blocks that are in snapshot snap-0987654321. The --starting-block-index parameter limits the results to block indexes greater than 1000, and the --max-results parameter limits the results to the first 100 blocks.

```
aws ebs list-snapshot-blocks --snapshot-id snap-0987654321 --starting-block-index 1000 --max-results 100
```

The following example response for the previous command lists the block indexes and block tokens in the snapshot. Use the get-snapshot-block command and specify the block index and block token of the block for which you want to get data. The block tokens are valid until the expiry time listed.

```
{
   "Blocks": [
      {
         "BlockIndex": 1001,
         "BlockToken": "AAABAV3/PNhXOynVdMYHUpPsetaSVjLBl1dtIGfbJv5OJ0sX855EzGTWos4a4"
      },
      {
         "BlockIndex": 1002,
         "BlockToken": "AAABATGQIgwr0WwIuqIMjCA/Sy7e/YoQFZsHejsGNvjKauzNgzeI13YHBfQB"
      },
      {
         "BlockIndex": 1007,
         "BlockToken": "AAABAZ9CTuQtUvp/dXqRWw4d07eOgTZ3jvn6hiW30W9duM8MiMe6yQayzF2c"
      },
      {
         "BlockIndex": 1012,
         "BlockToken": "AAABAQdzxhw0rVV6FPnsfo/YRIxo9JPR85XPF1BLjgHec6pgyYr6laElp0"
      },
      {
         "BlockIndex": 1030,
         "BlockToken": "AAABAaYvPax6mv+iGWLdTUjQTfW0uQ7Dqz6nSD9L+CbXnvpkswA6iDID52d"
      },
      {
         "BlockIndex": 1031,
         "BlockToken": "AAABATgWZCoXfWUKvTJbUXM1SPg59KVxJGL+BWBClkw6spzCJxJvDVATskJ"
      },
      ...
   ],
   "ExpiryTime": 1576287332.806,
   "VolumeSize": 32212254720,
   "BlockSize": 524288
}
```

List blocks that are different between two snapshots

The following list-changed-blocks example command returns the block indexes and block tokens of blocks that are different between snapshots snap-1234567890 and snap-0987654321. The --starting-block-index parameter limits the results to block indexes greater than 0, and the --max-results parameter limits the results to the first 500 blocks.

```
aws ebs list-changed-blocks --snapshot-id snap-1234567890 --snapshot-id snap-0987654321 --starting-block-index 0 --max-results 500
```
aws ebs list-changed-blocks --first-snapshot-id snap-1234567890 --second-snapshot-id snap-0987654321 --starting-block-index 0 --max-results 500

The following example response for the previous command shows that block indexes 0, 6000, 6001, 6002, and 6003 are different between the two snapshots. Additionally, block indexes 6001, 6002, and 6003 exist only in the first snapshot ID specified, and not in the second snapshot ID because there is no second block token listed in the response.

Use the get-snapshot-block command and specify the block index and block token of the block for which you want to get data. The block tokens are valid until the expiry time listed.

```
{
    "ChangedBlocks": [
    {
        "BlockIndex": 0,
        "FirstBlockToken": "AAABAVahm9SO60Dyi00Ryszn2ZjGjW/KN3uygGlsQOYwesbzbBdnXz2dGpmC",
        "SecondBlockToken": "AAABAf8o00c6UFi1rBbS2G1RaCRdDyBu91tLvtCQQxxoKV8qrUPQP7vcM6iWGSr"
    },
    {
        "BlockIndex": 6000,
        "FirstBlockToken": "AAABAbYSiZvJ0/R9tz8uISdSeclJN4kkaz8inFXVintPkaqVFLfCMeKe",
        "SecondBlockToken": "AAABAbZnqTdzFmKpamAnxVx1VqF/3jJz12q2eFDCgHmyNf777e1D9oVR"
    },
    {
        "BlockIndex": 6001,
        "FirstBlockToken": "AAABASBpSJ2UAD3PLxJnT6zsun4/T4sU25Bnb8jB5q6FRXHFqAIAqE04hJoR"
    },
    {
        "BlockIndex": 6002,
        "FirstBlockToken": "AAABASqX4/524nBtyIDx12tFEDUnnePYUKof4PBOuICe2A"
    },
    {
        "BlockIndex": 6003,
        "FirstBlockToken": "AAABASmJ005XAOce25rF4P18dRtyIDx12tFEDUnnePYUKof4PBOuICe2A"
    },
    ...
    ],
    "ExpiryTime": 1576308931.973,
    "VolumeSize": 32212254720,
    "BlockSize": 524288,
    "NextToken": "AAADARqELNng/sV98CYk/bJDCxELjMjLHnNSkHVlZva00zsPH/QM3Bi3zF/06Mdi/BbJArnBp3h"
}
```

Get block data from a snapshot

The following get-snapshot-block example command returns the data in the block index 6001 with block token AAABASBpSJ2UAD3PLxJnT6zsun4/T4sU25Bnb8jB5q6FRXHFqAIAqE04hJoR, in snapshot snap-1234567890. The binary data is output to the data file in the C:\Temp directory on a Windows computer. If you run the command on a Linux or Unix computer, replace the output path with /tmp/data to output the data to the data file in the /tmp directory.

```
aws ebs get-snapshot-block --snapshot-id snap-1234567890 --block-index 6001 --block-token AAABASBpSJ2UAD3PLxJnT6zsun4/T4sU25Bnb8jB5q6FRXHFqAIAqE04hJoR C:\Temp/data
```
The following example response for the previous command shows the size of the data returned, the checksum to validate the data, and the algorithm of the checksum. The binary data is automatically saved to the directory and file you specified in the request command.

```
{
    "DataLength": "524288",
    "Checksum": "cf0Y6/Fn0oFa4VvO0a/lD0zhTfl1PTKxzGv20KowXc=",
    "ChecksumAlgorithm": "SHA256"
}
```

Use the AWS CLI to write incremental snapshots

Start a snapshot

The following `start-snapshot` example command starts an 8 GiB snapshot, using snapshot `snap-123EXAMPLE1234567` as the parent snapshot. The new snapshot will be an incremental snapshot of the parent snapshot. The snapshot moves to an error state if there are no put or complete requests made for the snapshot within the specified 60 minute timeout period. The `550e8400-e29b-41d4-a716-446655440000` client token ensures idempotency for the request. If the client token is omitted, the AWS SDK automatically generates one for you. For more information about idempotency, see Idempotency for StartSnapshot API (p. 1271).

```
aws ebs start-snapshot --volume-size 8 --parent-snapshot snap-123EXAMPLE1234567 --timeout 60 --client-token 550e8400-e29b-41d4-a716-446655440000
```

The following example response for the previous command shows the snapshot ID, AWS account ID, status, volume size in GiB, and size of the blocks in the snapshot. The snapshot is started in a pending state. Specify the snapshot ID in subsequent `put-snapshot-block` commands to write data to the snapshot, then use the `complete-snapshot` command to complete the snapshot and change its status to completed.

```
{
    "SnapshotId": "snap-0aaEXAMPLEe306d62",
    "OwnerId": "111122223333",
    "Status": "pending",
    "VolumeSize": 8,
    "BlockSize": 524288
}
```

Put data into a snapshot

The following `put-snapshot` example command writes 524288 Bytes of data to block index 1000 on snapshot `snap-0aaEXAMPLEe306d62`. The Base64 encoded `QOD3gmEQQXAFjXx2Aa34W4FU2nZGyXfgtsUuktOw8DM=` checksum was generated using the SHA256 algorithm. The data that is transmitted is in the `/tmp/data` file.

```
aws ebs put-snapshot-block --snapshot-id snap-0aaEXAMPLEe306d62 --block-index 1000 --data-length 524288 --block-data /tmp/data --checksum QOD3gmEQQXAFjXx2Aa34W4FU2nZGyXfgtsUuktOw8DM= --checksum-algorithm SHA256
```

The following example response for the previous command confirms the data length, checksum, and checksum algorithm for the data received by the service.

```
{
    "DataLength": "524288",
    "Checksum": "QOD3gmEQQXAFjXx2Aa34W4FU2nZGyXfgtsUuktOw8DM=",
    "ChecksumAlgorithm": "SHA256"
}
```
Amazon Elastic Compute Cloud
User Guide for Windows Instances
EBS snapshots

Complete a snapshot

The following `complete-snapshot` example command completes snapshot `snap-0aaEXAMPLEe306d62`. The command specifies that 5 blocks were written to the snapshot. The `6D3nmwi5f2F0wlh7xX8QprrJBFzDX8aacc0cA3EKM3c=` checksum represents the checksum for the complete set of data written to a snapshot. For more information about checksums, see Use checksums (p. 1256) earlier in this guide.

```
aws ebs complete-snapshot --snapshot-id snap-0aaEXAMPLEe306d62 --changed-blocks-count 5 --checksum 6D3nmwi5f2F0wlh7xX8QprrJBFzDX8aacc0cA3EKM3c= --checksum-algorithm SHA256 --checksum-aggregation-method LINEAR
```

The following is an example response for the previous command.

```
{
    "Status": "pending"
}
```

Optimize performance

You can run API requests concurrently. Assuming PutSnapshotBlock latency is 100ms, then a thread can process 10 requests in one second. Furthermore, assuming your client application creates multiple threads and connections (for example, 100 connections), it can make 1000 (10 * 100) requests per second in total. This will correspond to a throughput of around 500 MB per second.

The following list contains few things to look for in your application:

- Is each thread using a separate connection? If the connections are limited on the application then multiple threads will wait for the connection to be available and you will notice lower throughput.
- Is there any wait time in the application between two put requests? This will reduce the effective throughput of a thread.
- The bandwidth limit on the instance – If bandwidth on the instance is shared by other applications, it could limit the available throughput for PutSnapshotBlock requests.

Be sure to take note of other workloads that might be running in the account to avoid bottlenecks. You should also build retry mechanisms into your EBS direct APIs workflows to handle throttling, timeouts, and service unavailability.

Review the EBS direct APIs service quotas to determine the maximum API requests that you can run per second. For more information, see Amazon Elastic Block Store Endpoints and Quotas in the AWS General Reference.

Frequently asked questions

**Can a snapshot be accessed using the EBS direct APIs if it has a pending status?**

No. The snapshot can be accessed only if it has a completed status.

**Are the block indexes returned by the EBS direct APIs in numerical order?**

Yes. The block indexes returned are unique, and in numerical order.

**Can I submit a request with a MaxResults parameter value of under 100?**

No. The minimum MaxResult parameter value you can use is 100. If you submit a request with a MaxResult parameter value of under 100, and there are more than 100 blocks in the snapshot, then the API will return at least 100 results.
Can I run API requests concurrently?

You can run API requests concurrently. Be sure to take note of other workloads that might be running in the account to avoid bottlenecks. You should also build retry mechanisms into your EBS direct APIs workflows to handle throttling, timeouts, and service unavailability. For more information, see Optimize performance (p. 1264).

Review the EBS direct APIs service quotas to determine the API requests that you can run per second. For more information, see Amazon Elastic Block Store Endpoints and Quotas in the AWS General Reference.

When running the ListChangedBlocks action, is it possible to get an empty response even though there are blocks in the snapshot?

Yes. If the changed blocks are scarce in the snapshot, the response may be empty but the API will return a next page token value. Use the next page token value to continue to the next page of results. You can confirm that you have reached the last page of results when the API returns a next page token value of null.

If the NextToken parameter is specified together with a StartingBlockIndex parameter, which of the two is used?

The NextToken is used, and the StartingBlockIndex is ignored.

How long are the block tokens and next tokens valid?

Block tokens are valid for seven days, and next tokens are valid for 60 minutes.

Are encrypted snapshots supported?

Yes. Encrypted snapshots can be accessed using the EBS direct APIs.

To access an encrypted snapshot, the user must have access to the KMS key used to encrypt the snapshot, and the AWS KMS decrypt action. See the Permissions for IAM users (p. 1251) section earlier in this guide for the AWS KMS policy to assign to a user.

Are public snapshots supported?

Public snapshots are not supported.

Does list snapshot block return all block indexes and block tokens in a snapshot, or only those that have data written to them?

It returns only block indexes and tokens that have data written to them.

Can I get a history of the API calls made by the EBS direct APIs on my account for security analysis and operational troubleshooting purposes?

Yes. To receive a history of EBS direct APIs API calls made on your account, turn on AWS CloudTrail in the AWS Management Console. For more information, see Log API Calls for EBS direct APIs with AWS CloudTrail (p. 1265).

Log API Calls for EBS direct APIs with AWS CloudTrail

The EBS direct APIs service is integrated with AWS CloudTrail. CloudTrail is a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures all API calls performed in EBS direct APIs as events. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon Simple Storage Service (Amazon S3) bucket. If you don't configure a trail, you can still view the most recent management events in the CloudTrail console in Event history. Data events are not captured in Event history. You can use the information collected by CloudTrail to determine the request that was made to EBS direct APIs, the IP address from which the request was made, who made the request, when it was made, and additional details.
EBS direct APIs Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When supported event activity occurs in EBS direct APIs, that activity is recorded in a CloudTrail event along with other AWS service events in Event history. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for EBS direct APIs, create a trail. A trail enables CloudTrail to deliver log files to an S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

• Overview for Creating a Trail
• CloudTrail Supported Services and Integrations
• Configuring Amazon SNS Notifications for CloudTrail
• Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

Supported API actions

For EBS direct APIs, you can use CloudTrail to log two types of events:

• Management events — Management events provide visibility into management operations that are performed on snapshots in your AWS account. The following API actions are logged by default as management events in trails:
  • StartSnapshot
  • CompleteSnapshot

For more information about logging management events, see Logging management events for trails in the CloudTrail User Guide.

• Data events — These events provide visibility into the snapshot operations performed on or within a snapshot. The following API actions can optionally be logged as data events in trails:
  • ListSnapshotBlocks
  • ListChangedBlocks
  • GetSnapshotBlock
  • PutSnapshotBlock

Data events are not logged by default when you create a trail. You can use only advanced event selectors to record data events on EBS direct API calls. For more information, see Logging data events for trails in the CloudTrail User Guide.

Note

If you perform an action on a snapshot that is shared with you, data events are not sent to the AWS account that owns the snapshot.

Identity information

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

• Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
• Whether the request was made with temporary security credentials for a role or federated user.
• Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentityElement.

Understand EBS direct APIs Log File Entries

A trail is a configuration that enables delivery of events as log files to an S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following are example CloudTrail log entries.

StartSnapshot

```json
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:root",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "user"
  },
  "eventTime": "2020-07-03T23:27:26Z",
  "eventSource": "ebs.amazonaws.com",
  "eventName": "StartSnapshot",
  "awsRegion": "eu-west-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "PostmanRuntime/7.25.0",
  "requestParameters": {
    "volumeSize": 8,
    "clientToken": "token",
    "encrypted": true
  },
  "responseElements": {
    "snapshotId": "snap-123456789012",
    "ownerId": "123456789012",
    "status": "pending",
    "startTime": "Jul 3, 2020 11:27:26 PM",
    "volumeSize": 8,
    "blockSize": 524288,
    "kmsKeyArn": "HIDDEN_DUE_TO_SECURITY_REASONS"
  },
  "requestID": "be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
  "eventID": "6e12345-2a4e-417c-aa78-7594fEXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

CompleteSnapshot

```json
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:root",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "user"
  },
  "eventTime": "2020-07-03T23:27:26Z",
  "eventSource": "ebs.amazonaws.com",
  "eventName": "CompleteSnapshot",
  "awsRegion": "eu-west-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "PostmanRuntime/7.25.0",
  "requestParameters": {
    "snapshotId": "snap-123456789012",
    "ownerId": "123456789012",
    "status": "completed",
    "startTime": "Jul 3, 2020 11:27:26 PM",
    "volumeSize": 8,
    "blockSize": 524288,
    "kmsKeyArn": "HIDDEN_DUE_TO_SECURITY_REASONS"
  },
  "requestID": "be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
  "eventID": "6e12345-2a4e-417c-aa78-7594fEXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```
EBS snapshots

"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"userName": "user"
},
"eventTime": "2020-07-03T23:28:24Z",
"eventSource": "ebs.amazonaws.com",
"eventName": "CompleteSnapshot",
"awsRegion": "eu-west-1",
"sourceIPAddress": "192.0.2.0",
"userAgent": "PostmanRuntime/7.25.0",
"requestParameters": {
  "snapshotId": "snap-123456789012",
  "changedBlocksCount": 5
},
"responseElements": {
  "status": "completed"
},
"requestID": "be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
"eventID": "6e12345-2a4e-417c-aa78-7594fEXAMPLE",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}

ListSnapshotBlocks

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAT4HP2R3A3JEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/user",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "user"
  },
  "eventTime": "2021-06-03T00:32:46Z",
  "eventSource": "ebs.amazonaws.com",
  "eventName": "ListSnapshotBlocks",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "111.111.111.111",
  "userAgent": "PostmanRuntime/7.28.0",
  "requestParameters": {
    "snapshotId": "snap-abcdef01234567890",
    "maxResults": 100,
    "startingBlockIndex": 0
  },
  "responseElements": null,
  "requestID": "example6-0e12-4aa9-b923-1555eexample",
  "eventID": "example4-218b-4f69-a9e0-2357dexample",
  "readOnly": true,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::EC2::Snapshot",
      "ARN": "arn:aws:ec2:us-west-2::snapshot/snap-abcdef01234567890"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": false,
  "recipientAccountId": "123456789012",
  "eventCategory": "Data",
  "tlsDetails": {
    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES128-SHA",
    "clientProvidedHostHeader": "ebs.us-west-2.amazonaws.com"
  }
}
ListChangedBlocks

{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAT4HPB2A03JEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/user",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
    "eventTime": "2021-06-02T21:11:46Z",
    "eventSource": "ebs.amazonaws.com",
    "eventName": "ListChangedBlocks",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "111.111.111.111",
    "userAgent": "PostmanRuntime/7.28.0",
    "requestParameters": {
        "firstSnapshotId": "snap-abcdef01234567890",
        "secondSnapshotId": "snap-9876543210abcdef0",
        "maxResults": 100,
        "startingBlockIndex": 0
    },
    "responseElements": null,
    "requestID": "example0-f4cb-4d64-8d84-72e1bexample",
    "eventID": "example3-fac4-4a78-8e8b-e9d3example",
    "readOnly": true,
    "resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::EC2::Snapshot",
            "ARN": "arn:aws:ec2:us-west-2::snapshot/snap-abcdef01234567890"
        },
        {
            "accountId": "123456789012",
            "type": "AWS::EC2::Snapshot",
            "ARN": "arn:aws:ec2:us-west-2::snapshot/snap-9876543210abcdef0"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-SHA",
        "clientProvidedHostHeader": "ebs.us-west-2.amazonaws.com"
    }
}

GetSnapshotBlock

{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAT4HPB2A03JEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/user",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
}
EBS snapshots

PutSnapshotBlock

{  
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAT4HPB2AO3JEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/user",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "user"
  },
  "eventTime": "2021-06-02T21:09:17Z",
  "eventSource": "ebs.amazonaws.com",
  "eventName": "PutSnapshotBlock",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "111.111.111.111",
  "userAgent": "PostmanRuntime/7.28.0",
  "requestParameters": {
    "snapshotId": "snap-abcd0f1234567890",
    "blockIndex": 1,
    "blockToken": "EXAMPLEiL5E3pMFPfaDWjExM2/mnSKh1mFcbjwe2mM7BwhrgCdPAEXAMPLE"
  },
  "responseElements": {
    "checksum": "exampleodSGvFSb1e3kxWUbOQ4TbzPurnsfVexample",
    "checksumAlgorithm": "SHA256"
  },
  "requestID": "example3-d5e0-4167-8ee8-50845example",
  "eventID": "example8-4d9a-4aad-b71d-bb31fexample",
}

PutSnapshotBlock

{  
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAT4HPB2AO3JEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/user",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "user"
  },
  "eventTime": "2021-06-02T21:09:17Z",
  "eventSource": "ebs.amazonaws.com",
  "eventName": "PutSnapshotBlock",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "111.111.111.111",
  "userAgent": "PostmanRuntime/7.28.0",
  "requestParameters": {
    "snapshotId": "snap-abcd0f1234567890",
    "blockIndex": 1,
    "dataLength": 524288,
    "checksum": "exampleodSGvFSb1e3kxWUbOQ4TbzPurnsfVexample",
    "checksumAlgorithm": "SHA256"
  },
  "responseElements": {
    "checksum": "exampleodSGvFSb1e3kxWUbOQ4TbzPurnsfVexample",
    "checksumAlgorithm": "SHA256"
  },
  "requestID": "example3-d5e0-4167-8ee8-50845example",
  "eventID": "example8-4d9a-4aad-b71d-bb31fexample",
}
EBS direct APIs and interface VPC endpoints

You can establish a private connection between your VPC and EBS direct APIs by creating an interface VPC endpoint. Interface endpoints are powered by AWS PrivateLink, a technology that enables you to privately access EBS direct APIs without an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don’t need public IP addresses to communicate with EBS direct APIs. Traffic between your VPC and EBS direct APIs does not leave the Amazon network.

Each interface endpoint is represented by one or more Elastic Network Interfaces in your subnets.

For more information, see Interface VPC endpoints (AWS PrivateLink) in the Amazon VPC User Guide.

Considerations for EBS direct APIs VPC endpoints

Before you set up an interface VPC endpoint for EBS direct APIs, ensure that you review Interface endpoint properties and limitations in the Amazon VPC User Guide.

VPC endpoint policies are not supported for EBS direct APIs. By default, full access to EBS direct APIs is allowed through the endpoint. However, you can control access to the interface endpoint using security groups. For more information, see Controlling access to services with VPC endpoints in the Amazon VPC User Guide.

Create an interface VPC endpoint for EBS direct APIs

You can create a VPC endpoint for EBS direct APIs using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Creating an interface endpoint in the Amazon VPC User Guide.

Create a VPC endpoint for EBS direct APIs using the following service name:

• com.amazonaws.

If you enable private DNS for the endpoint, you can make API requests to EBS direct APIs using its default DNS name for the Region, for example, ebs.us-east-1.amazonaws.com. For more information, see Accessing a service through an interface endpoint in the Amazon VPC User Guide.

Idempotency for StartSnapshot API

Idempotency ensures that an API request completes only once. With an idempotent request, if the original request completes successful, the subsequent retries return the result from the original successful request and they have no additional effect.
The StartSnapshot API supports idempotency using a client token. A client token is a unique string that you specify when you make an API request. If you retry an API request with the same client token and the same request parameters after it has completed successfully, the result of the original request is returned. If you retry a request with the same client token, but change one or more of the request parameters, the ConflictException error is returned.

If you do not specify your own client token, the AWS SDKs automatically generates a client token for the request to ensure that it is idempotent.

A client token can be any string that includes up to 64 ASCII characters. You should not reuse the same client tokens for different requests.

To make an idempotent StartSnapshot request with your own client token using the API

Specify the ClientToken request parameter.

```
POST /snapshots HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
User-Agent: <User agent parameter>
X-Amz-Date: 20200618T040724Z
Authorization: <Authentication parameter>
{
    "VolumeSize": 8,
    "ParentSnapshot": snap-123EXAMPLE1234567,
    "ClientToken": "550e8400-e29b-41d4-a716-446655440000",
    "Timeout": 60
}
```

To make an idempotent StartSnapshot request with your own client token using the AWS CLI

Specify the client-token request parameter.

```
C:\> aws ebs start-snapshot --region us-east-2 --volume-size 8 --parent-snapshot snap-123EXAMPLE1234567 --timeout 60 --client-token 550e8400-e29b-41d4-a716-446655440000
```

Automate the snapshot lifecycle

You can use Amazon Data Lifecycle Manager to automate the creation, retention, and deletion of snapshots that you use to back up your Amazon EBS volumes.

For more information, see Amazon Data Lifecycle Manager (p. 1272).

Amazon Data Lifecycle Manager

You can use Amazon Data Lifecycle Manager to automate the creation, retention, and deletion of EBS snapshots and EBS-backed AMIs. When you automate snapshot and AMI management, it helps you to:

- Protect valuable data by enforcing a regular backup schedule.
- Create standardized AMIs that can be refreshed at regular intervals.
- Retain backups as required by auditors or internal compliance.
- Reduce storage costs by deleting outdated backups.
- Create disaster recovery backup policies that back up data to isolated accounts.

When combined with the monitoring features of Amazon CloudWatch Events and AWS CloudTrail, Amazon Data Lifecycle Manager provides a complete backup solution for Amazon EC2 instances and individual EBS volumes at no additional cost.
Important
Amazon Data Lifecycle Manager cannot be used to manage snapshots or AMIs that are created by any other means.
Amazon Data Lifecycle Manager cannot be used to automate the creation, retention, and deletion of instance store-backed AMIs.

Contents
- How Amazon Data Lifecycle Manager works (p. 1273)
- Considerations for Amazon Data Lifecycle Manager (p. 1275)
- Automate snapshot lifecycles (p. 1277)
- Automate AMI lifecycles (p. 1284)
- Automate cross-account snapshot copies (p. 1290)
- View, modify, and delete lifecycle policies (p. 1297)
- AWS Identity and Access Management (p. 1300)
- Monitor the lifecycle of snapshots and AMIs (p. 1307)

How Amazon Data Lifecycle Manager works
The following are the key elements of Amazon Data Lifecycle Manager.

Elements
- Snapshots (p. 1273)
- EBS-backed AMIs (p. 1273)
- Target resource tags (p. 1273)
- Amazon Data Lifecycle Manager tags (p. 1274)
- Lifecycle policies (p. 1274)
- Policy schedules (p. 1275)

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 that’s unique to that snapshot is removed. The rest of the captured history of the volume is preserved.

For more information, see Amazon EBS snapshots (p. 1207).

EBS-backed AMIs
An Amazon Machine Image (AMI) provides the information that’s required to launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration. Amazon Data Lifecycle Manager supports EBS-backed AMIs only. EBS-backed AMIs include a snapshot for each EBS volume that’s attached to the source instance.

For more information, see Amazon Machine Images (AMI) (p. 21).

Target resource tags
Amazon Data Lifecycle Manager uses resource tags to identify the resources to back up. Tags are customizable metadata that you can assign to your AWS resources (including Amazon EC2 instances, EBS volumes and snapshots). An Amazon Data Lifecycle Manager policy (described later) targets an instance or volume for backup using a single tag. Multiple tags can be assigned to an instance or volume if you want to run multiple policies on it.
You can't use a `\` or `=' character in a tag key.

For more information, see Tag your Amazon EC2 resources (p. 1450).

**Amazon Data Lifecycle Manager tags**

Amazon Data Lifecycle Manager applies the following tags to all snapshots and AMIs created by a policy, to distinguish them from snapshots and AMIs created by any other means:

- `aws:dlm:lifecycle-policy-id`
- `aws:dlm:lifecycle-schedule-name`
- `aws:dlm:expirationTime`
- `dlm:managed`

You can also specify custom tags to be applied to snapshots and AMIs on creation. You can't use a `\` or `=' character in a tag key.

The target tags that Amazon Data Lifecycle Manager uses to associate volumes with a snapshot policy can optionally be applied to snapshots created by the policy. Similarly, the target tags that are used to associate instances with an AMI policy can optionally be applied to AMIs created by the policy.

**Lifecycle policies**

A lifecycle policy consists of these core settings:

- **Policy type**—Defines the type of resources that the policy can manage. Amazon Data Lifecycle Manager supports the following types of lifecycle policies:
  - Snapshot lifecycle policy—Used to automate the lifecycle of EBS snapshots. These policies can target individual EBS volumes or all EBS volumes attached to an instance.
  - EBS-backed AMI lifecycle policy—Used to automate the lifecycle of EBS-backed AMIs and their backing snapshots. These policies can target instances only.
  - Cross-account copy event policy—Used to automate snapshot copies across accounts. Use this policy type in conjunction with an EBS snapshot policy that shares snapshots across accounts.
- **Resource type**—Defines the type of resources that are targeted by the policy. Snapshot lifecycle policies can target instances or volumes. Use `VOLUME` to create snapshots of individual volumes, or use `INSTANCE` to create multi-volume snapshots of all of the volumes that are attached to an instance. For more information, see Multi-volume snapshots (p. 1212). AMI lifecycle policies can target instances only. One AMI is created that includes snapshots of all of the volumes that are attached to the target instance.
- **Target tags**—Specifies the tags that must be assigned to an EBS volume or an Amazon EC2 instance for it to be targeted by the policy.
- **Schedules**—The start times and intervals for creating snapshots or AMIs. The first snapshot or AMI creation operation starts within one hour after the specified start time. Subsequent snapshot or AMI creation operations start within one hour of their scheduled time. A policy can have up to four schedules: one mandatory schedule, and up to three optional schedules. For more information, see Policy schedules (p. 1275).
- **Retention**—Specifies how snapshots or AMIs are to be retained. You can retain snapshots or AMIs based either on their total count (count-based), or their age (age-based). For snapshot policies, when the retention threshold is reached, the oldest snapshot is deleted. For AMI policies, when the retention threshold is reached, the oldest AMI is deregistered and its backing snapshots are deleted.

For example, you could create a policy with settings similar to the following:

- Manages all EBS volumes that have a tag with a key of `account` and a value of `finance`. 
Amazon Elastic Compute Cloud
User Guide for Windows Instances
Amazon Data Lifecycle Manager

- Creates snapshots every 24 hours at 0900 UTC.
- Retains only the five most recent snapshots.
- Starts snapshot creation no later than 0959 UTC each day.

Policy schedules

Policy schedules define when snapshots or AMIs are created by the policy. Policies can have up to four schedules—one mandatory schedule, and up to three optional schedules.

Adding multiple schedules to a single policy lets you create snapshots or AMIs at different frequencies using the same policy. For example, you can create a single policy that creates daily, weekly, monthly, and yearly snapshots. This eliminates the need to manage multiple policies.

For each schedule, you can define the frequency, fast snapshot restore settings (snapshot lifecycle policies only), cross-Region copy rules, and tags. The tags that are assigned to a schedule are automatically assigned to the snapshots or AMIs that are created when the schedule is initiated. In addition, Amazon Data Lifecycle Manager automatically assigns a system-generated tag based on the schedule’s frequency to each snapshot or AMI.

Each schedule is initiated individually based on its frequency. If multiple schedules are initiated at the same time, Amazon Data Lifecycle Manager creates only one snapshot or AMI and applies the retention settings of the schedule that has the highest retention period. The tags of all of the initiated schedules are applied to the snapshot or AMI.

- (Snapshot lifecycle policies only) If more than one of the initiated schedules is enabled for fast snapshot restore, then the snapshot is enabled for fast snapshot restore in all of the Availability Zones specified across all of the initiated schedules. The highest retention settings of the initiated schedules is used for each Availability Zone.

- If more than one of the initiated schedules is enabled for cross-Region copy, the snapshot or AMI is copied to all Regions specified across all of the initiated schedules. The highest retention period of the initiated schedules is applied.

Considerations for Amazon Data Lifecycle Manager

Your AWS account has the following quotas related to Amazon Data Lifecycle Manager:

- You can create up to 100 lifecycle policies per Region.
- You can add up to 45 tags per resource.

The following considerations apply to lifecycle policies:

- A policy does not begin creating snapshots or AMIs until you set its activation status to enabled. You can configure a policy to be enabled upon creation.
- The first snapshot or AMI creation operation starts within one hour after the specified start time. Subsequent snapshot or AMI creation operations start within one hour of their scheduled time.
- If you modify a policy by removing or changing its target tags, the EBS volumes or instances with those tags are no longer managed by the policy.
- If you modify a schedule name for a policy, the snapshots or AMIs created under the old schedule name are no longer affected by the policy.
- If you modify a time-based retention schedule to use a new time interval, the new interval is used only for new snapshots or AMIs created after the change. The new schedule does not affect the retention schedule of snapshots or AMIs created before the change.
- You cannot change the retention schedule of a policy from count-based to time-based after creation. To make this change, you must create a new policy.
• If you disable a policy with an age-based retention schedule, the snapshots or AMIs that are set to expire while the policy is disabled are retained indefinitely. You must delete the snapshots or deregister the AMIs manually. When you enable the policy again, Amazon Data Lifecycle Manager resumes deleting snapshots or deregistering AMIs as their retention periods expire.

• If you delete the resource to which a policy with count-based retention applies, the policy no longer manages the previously created snapshots or AMIs. You must manually delete the snapshots or deregister the AMIs if they are no longer needed.

• If you delete the resource to which a policy with age-based retention applies, the policy continues to delete snapshots or deregister AMIs on the defined schedule, up to, but not including, the last snapshot or AMI. You must manually delete the last snapshot or deregister the last AMI if it is no longer needed.

• You can create multiple policies to back up an EBS volume or an Amazon EC2 instance. For example, if an EBS volume has two tags, where tag A is the target for policy A to create a snapshot every 12 hours, and tag B is the target for policy B to create a snapshot every 24 hours, Amazon Data Lifecycle Manager creates snapshots according to the schedules for both policies. Alternatively, you can achieve the same result by creating a single policy that has multiple schedules. For example, you can create a single policy that targets only tag A, and specify two schedules—one for every 12 hours and one for every 24 hours.

• If you create a policy that targets instances, and new volumes are attached to the instance after the policy has been created, the newly-added volumes are included in the backup at the next policy run. All volumes attached to the instance at the time of the policy run are included.

• For AMI lifecycle policies, when the AMI retention threshold is reached, the oldest AMI is deregistered and its backing snapshots are deleted.

• If a policy with a custom cron-based schedule and age-based or count-based retention rule is configured to create only one snapshot or AMI, the policy will not automatically delete that snapshot or AMI when the retention threshold is reached. You must manually delete the snapshot or deregister the AMI if it is no longer needed.

The following considerations apply to snapshot lifecycle policies and fast snapshot restore (p. 1339):

• A snapshot that is enabled for fast snapshot restore remains enabled even if you delete or disable the lifecycle policy, disable fast snapshot restore for the lifecycle policy, or disable fast snapshot restore for the Availability Zone. You can disable fast snapshot restore for these snapshots manually.

• If you enable fast snapshot restore and you exceed the maximum number of snapshots that can be enabled for fast snapshot restore, Amazon Data Lifecycle Manager creates snapshots as scheduled but does not enable them for fast snapshot restore. After a snapshot that is enabled for fast snapshot restore is deleted, the next snapshot that Amazon Data Lifecycle Manager creates is enabled for fast snapshot restore.

• When you enable fast snapshot restore for a snapshot, it takes 60 minutes per TiB to optimize the snapshot. We recommend that you create a schedule that ensures that each snapshot is fully optimized before Amazon Data Lifecycle Manager creates the next snapshot.

• You are billed for each minute that fast snapshot restore is enabled for a snapshot in a particular Availability Zone. Charges are pro-rated with a minimum of one hour. For more information, see Pricing and Billing (p. 1342).

  Note

  Depending on the configuration of your lifecycle policies, you could have multiple snapshots enabled for fast snapshot restore simultaneously.

The following considerations apply to sharing snapshots across accounts:

• You can only share snapshots that are unencrypted or that are encrypted using a customer managed key.

• You can’t share snapshots that are encrypted with the default EBS encryption KMS key.
• If you share encrypted snapshots, then you must also share the KMS key that was used to encrypt the source volume with the target accounts. For more information, see Allowing users in other accounts to use a KMS key in the AWS Key Management Service Developer Guide.

The following considerations apply to cross-account copy event policies:

• You can only copy snapshots that are unencrypted or that are encrypted using a customer managed key.
• You can create a cross-account copy event policy that copies snapshots that are shared outside of Amazon Data Lifecycle Manager.
• If you want to encrypt snapshots in the target account, then the IAM role selected for the cross-account copy event policy must have permission to use the required KMS key.

The following considerations apply to EBS-backed AMI policies and AMI deprecation:

• If you increase the AMI deprecation count for a schedule with count-based retention, the change is applied to all AMIs (existing and new) created by the schedule.
• If you increase the AMI deprecation period for a schedule with age-based retention, the change is applied to new AMIs only. Existing AMIs are not affected.
• If you remove the AMI deprecation rule from a schedule, Amazon Data Lifecycle Manager will not cancel deprecation for AMIs that were previously deprecated by that schedule.
• If you decrease the AMI deprecation count or period for a schedule, Amazon Data Lifecycle Manager will not cancel deprecation for AMIs that were previously deprecated by that schedule.
• If you manually deprecate an AMI that was created by an AMI policy, Amazon Data Lifecycle Manager will not override the deprecation.
• If you manually cancel deprecation for an AMI that was previously deprecated by an AMI policy, Amazon Data Lifecycle Manager will not override the cancellation.
• If an AMI is created by multiple conflicting schedules, and one or more of those schedules do not have an AMI deprecation rule, Amazon Data Lifecycle Manager will not deprecate that AMI.
• If an AMI is created by multiple conflicting schedules, and all of those schedules have an AMI deprecation rule, Amazon Data Lifecycle Manager will use the deprecation rule with the latest deprecation date.

Automate snapshot lifecycles

The following procedure shows you how to use Amazon Data Lifecycle Manager to automate Amazon EBS snapshot lifecycles.

Use one of the following procedures to create a snapshot lifecycle policy.

New console

To create a snapshot policy

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Elastic Block Store, Lifecycle Manager, and then choose Create lifecycle policy.
3. On the Select policy type screen, choose EBS snapshot policy and then choose Next.
4. In the Target resources section, do the following:
   a. For Target resource types, choose the type of resource to back up. Choose Volume to create snapshots of individual volumes, or choose Instance to create multi-volume snapshots from the volumes attached to an instance.
b. (For AWS Outpost customers only) For **Target resource location**, specify where the source resources are located.
   - If the source resources are located in an AWS Region, choose **AWS Region**. Amazon Data Lifecycle Manager backs up all resources of the specified type that have matching target tags in the current Region only. If the resource is located in a Region, snapshots created by the policy will be stored in the same Region.
   - If the source resources are located on an Outpost in your account, choose **AWS Outpost**. Amazon Data Lifecycle Manager backs up all resources of the specified type that have matching target tags across all of the Outposts in your account. If the resource is located on an Outpost, snapshots created by the policy can be stored in the same Region or on the same Outpost as the resource.
   - If you do not have any Outposts in your account, this option is hidden and AWS Region is selected for you.

c. For **Target resource tags**, choose the resource tags that identify the volumes or instances to back up. Only resources that have the specified tag key and value pairs are backed up by the policy.

5. For **Description**, enter a brief description for the policy.

6. For **IAM role**, choose the IAM role that has permissions to manage snapshots and to describe volumes and instances. To use the default role provided by Amazon Data Lifecycle Manager, choose **Default role**. Alternatively, to use a custom IAM role that you previously created, choose **Choose another role** and then select the role to use.

7. For **Policy tags**, add the tags to apply to the lifecycle policy. You can use these tags to identify and categorize your policies.

8. For **Policy status after creation**, choose **Enable policy** to start the policy runs at the next scheduled time, or **Disable policy** to prevent the policy from running. If you do not enable the policy now, it will not start creating snapshots until you manually enable it after creation.

9. Choose **Next**.

10. On the **Configure schedule** screen, configure the policy schedules. A policy can have up to 4 schedules. Schedule 1 is mandatory. Schedules 2, 3, and 4 are optional. For each policy schedule that you add, do the following:
   
a. In the **Schedule details** section do the following:
      
i. For **Schedule name**, specify a descriptive name for the schedule.
      ii. For **Frequency** and the related fields, configure the interval between policy runs. You can configure policy runs on a daily, weekly, monthly, or yearly schedule. Alternatively, choose **Custom cron expression** to specify an interval of up to one year. For more information, see Cron expressions in the Amazon CloudWatch Events User Guide.
      iii. For **Starting at**, specify the time at which the policy runs are scheduled to start. The first policy run starts within an hour after the scheduled time. The time must be entered in the **hh:mm** UTC format.
      iv. For **Retention type**, specify the retention policy for snapshots created by the schedule. You can retain snapshots based on either their total count or their age.
         
         For count-based retention, the range is 1 to 1000. After the maximum count is reached, the oldest snapshot is deleted when a new one is created.
         
         For age-based retention, the range is 1 day to 100 years. After the retention period of each snapshot expires, it is deleted.
         
         **Note**
         All schedules must have the same retention type. You can specify the retention type for Schedule 1 only. Schedules 2, 3, and 4 inherit the retention type from Schedule 1. Each schedule can have its own retention count or period.
v. (For AWS Outposts customers only) For **Snapshot destination**, specify the destination for snapshots created by the policy.

- If the policy targets resources in a Region, snapshots must be created in the same Region. AWS Region is selected for you.
- If the policy targets resources on an Outpost, you can choose to create snapshots on the same Outpost as the source resource, or in the Region that is associated with the Outpost.
- If you do not have any Outposts in your account, this option is hidden and AWS Region is selected for you.

b. In the **Tagging** section, do the following:

i. To copy all of the user-defined tags from the source volume to the snapshots created by the schedule, select **Copy tags from source**.
ii. To specify additional tags to assign to snapshots created by this schedule, choose **Add tags**.

c. To enable fast snapshot restore for snapshots created by the schedule, in the **Fast snapshot restore** section, select **Enable fast snapshot restore**. If you enable fast snapshot restore, you must choose the Availability Zones in which to enable it. If the schedule uses an age-based retention schedule, you must specify the period for which to enable fast snapshot restore for each snapshot. If the schedule uses count-based retention, you must specify the maximum number of snapshots to enable for fast snapshot restore.

If the schedule creates snapshots on an Outpost, you can't enable fast snapshot restore. Fast snapshot restore is not supported with local snapshots that are stored on an Outpost.

**Note**
You are billed for each minute that fast snapshot restore is enabled for a snapshot in a particular Availability Zone. Charges are pro-rated with a minimum of one hour.

d. To copy snapshots created by the schedule to an Outpost or to a different Region, in the **Cross-Region copy** section, select **Enable cross-Region copy**.

If the schedule creates snapshots in a Region, you can copy the snapshots to up to three additional Regions or Outposts in your account. You must specify a separate cross-Region copy rule for each destination Region or Outpost.

For each Region or Outpost, you can choose different retention policies and you can choose whether to copy all tags or no tags. If the source snapshot is encrypted, or if encryption by default is enabled, the copied snapshots are encrypted. If the source snapshot is unencrypted, you can enable encryption. If you do not specify a KMS key, the snapshots are encrypted using the default KMS key for EBS encryption in each destination Region. If you specify a KMS key for the destination Region, then the selected IAM role must have access to the KMS key.

**Note**
You must ensure that you do not exceed the number of concurrent snapshot copies per Region.

If the policy creates snapshots on an Outpost, then you can't copy the snapshots to a Region or to another Outpost and the cross-Region copy settings are not available.

e. In the **Cross-account sharing**, configure the policy to automatically share the snapshots created by the schedule with other AWS accounts. Do the following:

i. To enable sharing with other AWS accounts, select **Enable cross-account sharing**.
ii. To add the accounts with which to share the snapshots, choose Add account, enter the 12-digit AWS account ID, and choose Add.

iii. To automatically unshare shared snapshots after a specific period, select Unshare automatically. If you choose to automatically unshare shared snapshots, the period after which to automatically unshare the snapshots cannot be longer than the period for which the policy retains its snapshots. For example, if the policy's retention configuration retains snapshots for a period of 5 days, you can configure the policy to automatically unshare shared snapshots after periods up to 4 days. This applies to policies with age-based and count-based snapshot retention configurations.

If you do not enable automatic unsharing, the snapshot is shared until it is deleted.

f. To add additional schedules, choose Add another schedule, which is located at the top of the screen. For each additional schedule, complete the fields as described previously in this topic.

g. After you have added the required schedules, choose Review policy.

11. Review the policy summary, and then choose Create policy.

Old console

To create a snapshot policy

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

2. In the navigation pane, choose Elastic Block Store, Lifecycle Manager, and then choose Create lifecycle policy.

3. Provide the following information for your policy as needed:

   • **Description**—A description of the policy.
   • **Policy type**—The type of policy to create. Choose EBS snapshot policy.
   • **Resource type**—The type of resource to back up. Choose Volume to create snapshots of individual volumes, or choose Instance to create multi-volume snapshots from the volumes attached to an instance.
   • **Resource location**—The location of the resources to backup. If the source resources are located in an AWS Region, choose AWS Region. If the source resources are located on an Outpost in your account, choose AWS Outpost. If you choose AWS Outpost, Amazon Data Lifecycle Manager backs up all resources of the specified type that have matching target tags across all of the Outposts in your account.
   
   If you do not have any Outposts in your account, then AWS Region is selected by default.

   **Note**
   
   If the resource is located in a Region, snapshots created by the policy will be stored in the same Region. If the resource is located on an Outpost, snapshots created by the policy can be stored in the same Region or on the same Outpost as the resource.

   • **Target with these tags**—The resource tags that identify the volumes or instances to back up. Only resources that have the specified tag key and value pairs are backed up by the policy.
   • **Policy tags**—The tags to apply to the lifecycle policy.

4. For IAM role, choose the IAM role that has permissions to create, delete, and describe snapshots and to describe volumes and instances. AWS provides a default role, or you can create a custom IAM role.

5. Add the policy schedules. Schedule 1 is mandatory. Schedules 2, 3, and 4 are optional. For each policy schedule that you add, specify the following information:

   • **Schedule name**—A name for the schedule.
• **Frequency**—The interval between policy runs. You can configure policy runs on a daily, weekly, monthly, or yearly schedule. Alternatively, choose **Custom cron expression** to specify an interval of up to one year. For more information, see Cron expressions in the Amazon CloudWatch Events User Guide.

• **Starting at hh:mm UTC**—The time at which the policy runs are scheduled to start. The first policy run starts within an hour after the scheduled time.

• **Retention type**—You can retain snapshots based on either their total count or their age. For count-based retention, the range is 1 to 1000. After the maximum count is reached, the oldest snapshot is deleted when a new one is created. For age-based retention, the range is 1 day to 100 years. After the retention period of each snapshot expires, it is deleted. The retention period should be greater than or equal to the interval.

**Note**
All schedules must have the same retention type. You can specify the retention type for Schedule 1 only. Schedules 2, 3, and 4 inherit the retention type from Schedule 1. Each schedule can have its own retention count or period.

• **Snapshot destination**—Specifies the destination for snapshots created by the policy. To create snapshots in the same AWS Region as the source resource, choose **AWS Region**. To create snapshots on an Outpost, choose **AWS Outpost**.

If the policy targets resources in a Region, snapshots are created in the same Region, and cannot be created on an Outpost.

If the policy targets resources on an Outpost, snapshots can be created on the same Outpost as the source resource, or in the Region that is associated with the Outpost.

• **Copy tags from source**—Choose whether to copy all of the user-defined tags from the source volume to the snapshots created by the schedule.

• **Variable tags**—If the source resource is an instance, you can choose to automatically tag your snapshots with the following variable tags:
  - **instance-id**—The ID of the source instance.
  - **timestamp**—The date and time of the policy run.

• **Additional tags**—Specify any additional tags to assign to the snapshots created by this schedule.

• **Fast snapshot restore**—Choose whether to enable fast snapshot restore for all snapshots that are created by the schedule. If you enable fast snapshot restore, you must choose the Availability Zones in which to enable it. You are billed for each minute that fast snapshot restore is enabled for a snapshot in a particular Availability Zone. Charges are pro-rated with a minimum of one hour. You can also specify the maximum number of snapshots that can be enabled for fast snapshot restore.

If the policy creates snapshots on an Outpost, you can't enable fast snapshot restore. Fast snapshot restore is not supported with local snapshots that are stored on an Outpost.

• **Cross region copy**—If the policy creates snapshots in a Region, then you can copy the snapshots to up to three additional Regions or Outposts in your account. You must specify a separate cross-Region copy rule for each destination Region or Outpost.

For each Region or Outpost, you can choose different retention policies and you can choose whether to copy all tags or no tags. If the source snapshot is encrypted, or if encryption by default is enabled, the copied snapshots are encrypted. If the source snapshot is unencrypted, you can enable encryption. If you do not specify a KMS key, the snapshots are encrypted using the default KMS key for EBS encryption in each destination Region. If you specify a KMS key for the destination Region, then the selected IAM role must have access to the KMS key.

You must ensure that you do not exceed the number of concurrent snapshot copies per Region.
If the policy creates snapshots on an Outpost, then you can't copy the snapshots to a Region or to another Outpost and the cross-Region copy settings are not available.

6. For **Policy status after creation**, choose **Enable policy** to start the policy runs at the next scheduled time, or **Disable policy** to prevent the policy from running.

7. Choose **Create Policy**.

**Command line**

Use the `create-lifecycle-policy` command to create a snapshot lifecycle policy. For **PolicyType**, specify **EBS_SNAPSHOT_MANAGEMENT**.

**Note**

To simplify the syntax, the following examples use a JSON file, `policyDetails.json`, that includes the policy details.

**Example 1—Snapshot lifecycle policy**

This example creates a snapshot lifecycle policy that creates snapshots of all volumes that have a tag key of `costcenter` with a value of `115`. The policy includes two schedules. The first schedule creates a snapshot every day at 03:00 UTC. The second schedule creates a weekly snapshot every Friday at 17:00 UTC.

```
aws dlm create-lifecycle-policy
   --description "My volume policy"
   --state ENABLED --execution-role-arn
   arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole
   --policy-details file://policyDetails.json
```

The following is an example of the `policyDetails.json` file.

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": ["VOLUME"],
    "TargetTags": [{
        "Key": "costcenter",
        "Value": "115"
    }],
    "Schedules": [{
        "Name": "DailySnapshots",
        "TagsToAdd": [{
            "Key": "type",
            "Value": "myDailySnapshot"
        }],
        "CreateRule": {
            "Interval": 24,
            "IntervalUnit": "HOURS",
            "Times": [
                "03:00"
            ]
        },
        "RetainRule": {
            "Count": 5
        },
        "CopyTags": false
    },
    {
        "Name": "WeeklySnapshots",
```
Upon success, the command returns the ID of the newly created policy. The following is example output.

```json
{
  "PolicyId": "policy-0123456789abcdef0"
}
```

**Example 2—Snapshot lifecycle policy that automates local snapshots of Outpost resources**

This example creates a snapshot lifecycle policy that creates snapshots of volumes tagged with `team=dev` across all of your Outposts. The policy creates the snapshots on the same Outposts as the source volumes. The policy creates snapshots every 12 hours starting at 00:00 UTC.

```
aws dlm create-lifecycle-policy \
  --description "My local snapshot 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.

```json
{
  "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
  "ResourceTypes": "VOLUME",
  "ResourceLocations": "OUTPOST",
  "TargetTags": [{
    "Key": "team",
    "Value": "dev"
  }],
  "Schedules": [{
    "Name": "on-site backup",
    "CreateRule": {
      "Interval": 12,
      "IntervalUnit": "HOURS",
      "Times": [
        "00:00"
      ],
      "Location": [
        "OUTPOST_LOCAL"
      ]
    },
    "RetainRule": {
      "Count": 1
    },
    "CopyTags": false
  }]
}
```
Example 3—Snapshot lifecycle policy that creates snapshots in a Region and copies them to an Outpost

The following example policy creates snapshots of volumes that are tagged with `team=dev`.
Snapshots are created in the same Region as the source volume. Snapshots are created every 12 hours starting at 00:00 UTC, and retains a maximum of 1 snapshot. The policy also copies the snapshots to Outpost arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0, encrypts the copied snapshots using the default encryption KMS key, and retains the copies for 1 month.

```
aws dlm create-lifecycle-policy \
  --description "Copy snapshots to Outpost" \
  --state ENABLED --execution-role-arn \
  arn:aws:iam::1234567891012345678910:role/AWSDataLifecycleManagerDefaultRole \
  --policy-details file://policyDetails.json
```

The following is an example of the `policyDetails.json` file.

```json
{
  "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
  "ResourceTypes": "VOLUME",
  "ResourceLocations": "CLOUD",
  "TargetTags": [{
    "Key": "team",
    "Value": "dev"
  }],
  "Schedules": [{
    "Name": "on-site backup",
    "CopyTags": false,
    "CreateRule": {
      "Interval": 12,
      "IntervalUnit": "HOURS",
      "Times": [
        "00:00"
      ],
      "Location": "CLOUD"
    },
    "RetainRule": {
      "Count": 1
    },
    "CrossRegionCopyRules" : [
      {
        "Target": "arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0",
        "Encrypted": true,
        "CopyTags": true,
        "RetainRule": {
          "Interval": 1,
          "IntervalUnit": "MONTHS"
        }
      }
    ]
  }
}
```

Automate AMI lifecycles

The following procedure shows you how to use Amazon Data Lifecycle Manager to automate EBS-backed AMI lifecycles.

Use one of the following procedures to create an AMI lifecycle policy.
To create an AMI policy

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Elastic Block Store, Lifecycle Manager, and then choose Create lifecycle policy.
3. On the Select policy type screen, choose EBS-backed AMI policy, and then choose Next.
4. In the Target resources section, for Target resource tags, choose the resource tags that identify the volumes or instances to back up. The policy backs up only the resources that have the specified tag key and value pairs.
5. For Description, enter a brief description for the policy.
6. For IAM role, choose the IAM role that has permissions to manage AMIs and snapshot and to describe instances. To use the default role provided by Amazon Data Lifecycle Manager, choose Default role. Alternatively, to use a custom IAM role that you previously created, choose Choose another role, and then select the role to use.
7. For Policy tags, add the tags to apply to the lifecycle policy. You can use these tags to identify and categorize your policies.
8. For Policy status after creation, choose Enable policy to start running the policy at the next scheduled time, or Disable policy to prevent the policy from running. If you do not enable the policy now, it will not start creating AMIs until you manually enable it after creation.
9. In the Instance reboot section, indicate whether instances should be rebooted before AMI creation. To prevent the targeted instances from being rebooted, choose No. Choosing No could cause data consistency issues. To reboot instances before AMI creation, choose Yes. Choosing this ensures data consistency, but could result in multiple targeted instances rebooting simultaneously.
10. Choose Next.
11. On the Configure schedule screen, configure the policy schedules. A policy can have up to four schedules. Schedule 1 is mandatory. Schedules 2, 3, and 4 are optional. For each policy schedule that you add, do the following:
   a. In the Schedule details section do the following:
      i. For Schedule name, specify a descriptive name for the schedule.
      ii. For Frequency and the related fields, configure the interval between policy runs. You can configure policy runs on a daily, weekly, monthly, or yearly schedule. Alternatively, choose Custom cron expression to specify an interval of up to one year. For more information, see Cron expressions in the Amazon CloudWatch Events User Guide.
      iii. For Starting at, specify the time to start the policy runs. The first policy run starts within an hour after the time that you schedule. You must enter the time in the hh:mm UTC format.
      iv. For Retention type, specify the retention policy for AMIs created by the schedule. You can retain AMIs based on either their total count or their age.

         For count-based retention, the range is 1 to 1000. After the maximum count is reached, the oldest AMI is deregistered when a new one is created.

         For age-based retention, the range is 1 day to 100 years. After the retention period of each AMI expires, it is deregistered.

         Note
         All schedules must have the same retention type. You can specify the retention type for Schedule 1 only. Schedules 2, 3, and 4 inherit the retention type from Schedule 1. Each schedule can have its own retention count or period.
   b. In the Tagging section, do the following:
To copy all of the user-defined tags from the source instance to the AMIs created by the
schedule, select **Copy tags from source**.

By default, AMIs created by the schedule are automatically tagged with the ID of the
source instance. To prevent this automatic tagging from happening, for **Variable tags**, remove the `instance-id:$(instance-id)` tile.

To specify additional tags to assign to AMIs created by this schedule, choose **Add tags**.

To deprecate AMIs when they should no longer be used, in the **AMI deprecation** section,
select **Enable AMI deprecation for this schedule** and then specify the AMI deprecation rule.
The AMI deprecation rule specifies when AMIs are to be deprecated.

If the schedule uses count-based AMI retention, you must specify the number of oldest
AMIs to deprecate. The deprecation count must be less than or equal to the schedule’s
AMI retention count, and it can’t be greater than 1000. For example, if the schedule is
configured to retain a maximum of 5 AMIs, then you can configure the scheduled to
deprecate up to 5 oldest AMIs.

If the schedule uses age-based AMI retention, you must specify the period after which AMIs
are to be deprecated. The deprecation count must be less than or equal to the schedule’s
AMI retention period, and it can’t be greater than 10 years (120 months, 520 weeks, or
3650 days). For example, if the schedule is configured to retain AMIs for 10 days, then you
can configure the scheduled to deprecate AMIs after periods up to 10 days after creation.

To copy AMIs created by the schedule to different Regions, in the **Cross-Region copy**
section, select **Enable cross-Region copy**. You can copy AMIs to up to three additional
Regions in your account. You must specify a separate cross-Region copy rule for each
destination Region.

For each destination Region, you can specify the following:

- A retention policy for the AMI copy. When the retention period expires, the copy in the
destination Region is automatically deregistered.

- Encryption status for the AMI copy. If the source AMI is encrypted, or if encryption
by default is enabled, the copied AMIs are always encrypted. If the source AMI is
unencrypted and encryption by default is disabled, you can optionally enable encryption.
If you do not specify a KMS key, the AMIs are encrypted using the default KMS key for
EBS encryption in each destination Region. If you specify a KMS key for the destination
Region, then the selected IAM role must have access to the KMS key.

- A deprecation rule for the AMI copy. When the deprecation period expires, the AMI copy is
automatically deprecated. The deprecation period must be less than or equal to thecopy
retention period, and it can’t be greater than 10 years.

- Whether to copy all tags or no tags from the source AMI.

**Note**

Do not exceed the number of concurrent AMI copies per Region.

To add additional schedules, choose **Add another schedule**, which is located at the top of
the screen. For each additional schedule, complete the fields as described previously in this
topic.

After you have added the required schedules, choose **Review policy**.

12. Review the policy summary, and then choose **Create policy**.
To create an AMI lifecycle policy

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Elastic Block Store, Lifecycle Manager, and then choose Create lifecycle policy.
3. Provide the following information for your policy as needed:
   - **Description**—A description of the policy.
   - **Policy type**—The type of policy to create. Choose EBS-backed AMI policy.
   - **Target with these tags**—The resource tags that identify the instances to back up. Only instances that have the specified tag key and value pairs are backed up by the policy.
   - **Policy tags**—The tags to apply to the lifecycle policy.
4. For IAM role, choose the IAM role that has permissions to manage images. AWS provides a default roles, or you can create a custom IAM role.
5. Add the policy schedules. Schedule 1 is mandatory. Schedules 2, 3, and 4 are optional. For each policy schedule that you add, specify the following information:
   - **Schedule name**—A name for the schedule.
   - **Frequency**—The interval between policy runs. You can configure policy runs on a daily, weekly, monthly, or yearly schedule. Alternatively, choose Custom cron expression to specify an interval of up to one year. For more information, see Cron expressions in the Amazon CloudWatch Events User Guide.
   - **Starting at hh:mm UTC**—The time at which the policy runs are scheduled to start. The first policy run starts within an hour after the scheduled time.
   - **Retention type**—You can retain AMIs based on either their total count or their age. For count-based retention, the range is 1 to 1000. After the maximum count is reached, the oldest AMI is deleted when a new one is created. For age-based retention, the range is 1 day to 100 years. After the retention period of each AMI expires, it is deleted. The retention period should be greater than or equal to the interval.

   **Note**
   All schedules must have the same retention type. You can specify the retention type for Schedule 1 only. Schedules 2, 3, and 4 inherit the retention type from Schedule 1.
   Each schedule can have its own retention count or period.
   - **Copy tags from source**—Choose whether to copy all of the user-defined tags from the source instance to the AMIs created by the schedule.
   - **Dynamic tags**—You can choose to automatically tag your AMIs with the ID of the source instance.
   - **Additional tags**—Specify any additional tags to assign to the AMIs created by this schedule.
   - **Enable cross Region copy**—You can copy AMIs to up to three additional Regions.

   For each Region, you can choose different retention policies and you can choose whether to copy all tags or no tags. If the source AMI is encrypted, or if encryption by default is enabled, the copied AMIs are encrypted. If the AMI is unencrypted, you can enable encryption. If you do not specify a KMS key, the AMIs are encrypted using the default KMS key for EBS encryption in each destination Region. If you specify a KMS key for the destination Region, then the selected IAM role must have access to the KMS key.

   Do not exceed the number of concurrent AMI copies per Region.
6. Indicate whether instances should be rebooted before AMI creation. To prevent the targeted instances from being rebooted, for Reboot Instance at policy run, choose No. Choosing this option could cause data consistency issues. To reboot instances before AMI creation, for Reboot
**Instance at policy run**, choose Yes. Choosing this ensures data consistency but could result in multiple targeted instances rebooting simultaneously.

7. For **Policy status after creation**, choose **Enable policy** to start the policy runs at the next scheduled time, or **Disable policy** to prevent the policy from running.

8. Choose **Create Policy**.

Command line

Use the `create-lifecycle-policy` command to create an AMI lifecycle policy. For **PolicyType**, specify **IMAGE_MANAGEMENT**.

**Note**

To simplify the syntax, the following examples use a JSON file, `policyDetails.json`, that includes the policy details.

**Example 1: Age-based retention and AMI deprecation**

This example creates an AMI lifecycle policy that creates AMIs of all instances that have a tag key of purpose with a value of production without rebooting the targeted instances. The policy includes one schedule that creates an AMI every day at 01:00 UTC. The policy retains AMIs for 2 days and deprecates them after 1 day. It also copies the tags from the source instance to the AMIs that it creates.

```
aws dlm create-lifecycle-policy \
  --description "My AMI policy" \
  --state ENABLED --execution-role-arn \
  arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRoleForAMIManagement \
  --policy-details file://policyDetails.json
```

The following is an example of the `policyDetails.json` file.

```
{
  "PolicyType": "IMAGE_MANAGEMENT",
  "ResourceTypes": [
    "INSTANCE"
  ],
  "TargetTags": [{
    "Key": "purpose",
    "Value": "production"
  }],
  "Schedules": [{
    "Name": "DailyAMIs",
    "TagsToAdd": [{
      "Key": "type",
      "Value": "myDailyAMI"
    }],
    "CreateRule": {
      "Interval": 24,
      "IntervalUnit": "HOURS",
      "Times": [
        "01:00"
      ]
    },
    "RetainRule": {
      "Interval": 2,
      "IntervalUnit": "DAYS"
    },
    "DeprecateRule": {
      "Interval": 1,
      "IntervalUnit": "DAYS"
    }
  }
}
```
Upon success, the command returns the ID of the newly created policy. The following is example output.

```
{
  "PolicyId": "policy-9876543210abcdef0"
}
```

**Example 2: Count-based retention and AMI deprecation with cross-Region copy**

This example creates an AMI lifecycle policy that creates AMIs of all instances that have a tag key of `purpose` with a value of `production` and reboots the target instances. The policy includes one schedule that creates an AMI every 6 hours starting at 17:30 UTC. The policy retains 3 AMIs and automatically deprecates the 2 oldest AMIs. It also has a cross-Region copy rule that copies AMIs to `us-east-1`, retains 2 AMI copies, and automatically deprecates the oldest AMI.

```
aws dlm create-lifecycle-policy \
--description "My AMI policy" \
--state ENABLED \
--execution-role-arn arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRoleForAMIManagement \
--policy-details file://policyDetails.json
```

The following is an example of the `policyDetails.json` file.

```
{
  "PolicyType": "IMAGE_MANAGEMENT",
  "ResourceTypes": [
    "INSTANCE"
  ],
  "TargetTags": [{
    "Key": "purpose",
    "Value": "production"
  }],
  "Parameters": {
    "NoReboot": true
  },
  "Schedules": [{
    "Name": "Schedule1",
    "CopyTags": true,
    "CreateRule": {
      "Interval": 6,
      "IntervalUnit": "HOURS",
      "Times": ["17:30"]
    },
    "RetainRule": {
      "Count": 3
    },
    "DeprecateRule": {
      "Count": 2
    },
    "CrossRegionCopyRules": [{
      "TargetRegion": "us-east-1",
      "Encrypted": true,
      "RetainRule": {
```

1289
Automate cross-account snapshot copies

Automating cross-account snapshot copies enables you to copy your Amazon EBS snapshots to specific Regions in an isolated account and encrypt those snapshots with an encryption key. This enables you to protect yourself against data loss in the event of your account being compromised.

Automating cross-account snapshot copies involves two accounts:

- **Source account**—The source account is the account that creates and shares the snapshots with the target account. In this account, you must create an EBS snapshot policy that creates snapshots at set intervals and then shares them with other AWS accounts.

- **Target account**—The target account is the account with destination account with which the snapshots are shared, and it is the account that creates copies of the shared snapshots. In this account, you must create a cross-account copy event policy that automatically copies snapshots that are shared with it by one or more specified source accounts.

**Topics**

- Create cross-account snapshot copy policies (p. 1290)
- Specify snapshot description filters (p. 1297)

Create cross-account snapshot copy policies

To prepare the source and target accounts for cross-account snapshot copying, you need to perform the following steps:

**Topics**

- Step 1: Create the EBS snapshot policy (Source account) (p. 1290)
- Step 2: Share the customer managed key (Source account) (p. 1291)
- Step 3: Create cross-account copy event policy (Target account) (p. 1292)
- Step 4: Allow IAM role to use the required KMS keys (Target account) (p. 1295)

**Step 1: Create the EBS snapshot policy (Source account)**

In the source account, create an EBS snapshot policy that will create the snapshots and share them with the required target accounts.

When you create the policy, ensure that you enable cross-account sharing and that you specify the target AWS accounts with which to share the snapshots. These are the accounts with which the snapshots are to be shared. If you are sharing encrypted snapshots, then you must give the selected target accounts permission to use the KMS key used to encrypt the source volume. For more information, see Step 2: Share the customer managed key (Source account) (p. 1291).

For more information about creating an EBS snapshot policy, see Automate snapshot lifecycles (p. 1277).
Use one of the following methods to create the EBS snapshot policy.

**Step 2: Share the customer managed key (Source account)**

If you are sharing encrypted snapshots, you must grant the IAM role and the target AWS accounts (that you selected in the previous step) permissions to use the customer managed key that was used to encrypt the source volume.

**Note**

Perform this step only if you are sharing encrypted snapshots. If you are sharing unencrypted snapshots, skip this step.

**Console**

2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. In the navigation pane, choose **Customer managed key** and then select the KMS key that you need to share with the target accounts.

    Make note of the KMS key ARN, you'll need this later.
4. On the **Key policy** tab, scroll down to the **Key users** section. Choose **Add**, enter the name of the IAM role that you selected in the previous step, and then choose **Add**.
5. On the **Key policy** tab, scroll down to the **Other AWS accounts** section. Choose **Add other AWS accounts**, and then add all of the target AWS accounts that you chose to share the snapshots with in the previous step.
6. Choose **Save changes**.

**Command line**

Use the `get-key-policy` command to retrieve the key policy that is currently attached to the KMS key.

For example, the following command retrieves the key policy for a KMS key with an ID of `9d5e2b3d-e410-4a27-a958-19e220d83a1e` and writes it to a file named `snapshotKey.json`.

```bash
$ aws kms get-key-policy
--policy-name default --key-id 9d5e2b3d-e410-4a27-a958-19e220d83a1e
--query Policy --output text > snapshotKey.json
```

Open the key policy using your preferred text editor. Add the ARN of the IAM role that you specified when you created the snapshot policy and the ARNs of the target accounts with which to share the KMS key.

For example, in the following policy, we added the ARN of the default IAM role, and the ARN of the root account for target account `222222222222`.

```json
{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::111111111111:role/service-role/AWSDataLifecycleManagerDefaultRole",
            "arn:aws:iam::222222222222:root"
        ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncryptToKMS",
        "kms:ReEncryptFromKMS",
        "kms:UntagResource",
        "kms:TagResource",
        "kms:GetKeyPolicy",
        "kms:ListKeyPolicies"
    ]
}
```

1291
Save and close the file. Then use the `put-key-policy` command to attach the updated key policy to the KMS key.

```bash
aws kms put-key-policy
--policy-name default --key-id 9d5e2b3d-e410-4a27-a958-19e220d83a1e
--policy file://snapshotKey.json
```

**Step 3: Create cross-account copy event policy (Target account)**

In the target account, you must create a cross-account copy event policy that will automatically copy snapshots that are shared by the required source accounts.

This policy runs in the target account only when one of the specified source accounts shares snapshot with the account.

Use one of the following methods to create the cross-account copy event policy.

**New console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose Elastic Block Store, Lifecycle Manager, and then choose Create lifecycle policy.
3. On the Select policy type screen, choose Cross-account copy event policy, and then choose Next.
4. For Policy description, enter a brief description for the policy.
5. For Policy tags, add the tags to apply to the lifecycle policy. You can use these tags to identify and categorize your policies.
6. In the Event settings section, define the snapshot sharing event that will cause the policy to run. Do the following:
a. For **Sharing accounts**, specify the source AWS accounts from which you want to copy the shared snapshots. Choose **Add account**, enter the 12-digit AWS account ID, and then choose **Add**.

b. For **Filter by description**, enter the required snapshot description using a regular expression. Only snapshots that are shared by the specified source accounts and that have descriptions that match the specified filter are copied by the policy. For more information, see **Specify snapshot description filters (p. 1297)**.

7. For **IAM role**, choose the IAM role that has permissions to perform snapshot copy actions. To use the default role provided by Amazon Data Lifecycle Manager, choose **Default role**. Alternatively, to use a custom IAM role that you previously created, choose **Choose another role** and then select the role to use.

If you are copying encrypted snapshots, you must grant the selected IAM role permissions to use the encryption KMS key used to encrypt the source volume. Similarly, if you are encrypting the snapshot in the destination Region using a different KMS key, you must grant the IAM role permission to use the destination KMS key. For more information, see **Step 4: Allow IAM role to use the required KMS keys (Target account) (p. 1295)**.

8. In the **Copy action** section, define the snapshot copy actions that the policy should perform when it is activated. The policy can copy snapshots to up to three Regions. You must specify a separate copy rule for each destination Region. For each rule that you add, do the following:

   a. For **Name**, enter a descriptive name for the copy action.
   b. For **Target Region**, select the Region to which to copy the snapshots.
   c. For **Expire**, specify how long to retain the snapshot copies in the target Region after creation.
   d. To encrypt the snapshot copy, for **Encryption**, select **Enable encryption**. If the source snapshot is encrypted, or if encryption by default is enabled for your account, the snapshot copy is always encrypted, even if you do enable encryption here. If the source snapshot is unencrypted and encryption by default is not enabled for your account, you can choose to enable or disable encryption. If you enable encryption, but do not specify a KMS key, the snapshots are encrypted using the default encryption KMS key in each destination Region. If you specify a KMS key for the destination Region, you must have access to the KMS key.

9. To add additional snapshot copy actions, choose **Add new Regions**.

10. For **Policy status after creation**, choose **Enable policy** to start the policy runs at the next scheduled time, or **Disable policy** to prevent the policy from running. If you do not enable the policy now, it will not start copying snapshots until you manually enable it after creation.

11. Choose **Create policy**.

**Old console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Lifecycle Manager** and then choose **Create Lifecycle Policy**.
3. For **Policy Type**, choose **Cross-account copy event policy**. For **Description**, enter a brief description for the policy.
4. In the **Cross-account copy event settings** section, for **Copy snapshots shared by**, enter the source AWS accounts from which you want to copy the shared snapshots.
5. For **Snapshot description filter**, enter the required snapshot description using a regular expression. Only snapshots that are shared by the specified sources accounts and that have descriptions that match the specified filter are copied by the policy. For more information, see **Specify snapshot description filters (p. 1297)**.
6. For **IAM role**, choose the IAM role that has permissions to perform the snapshot copy action. AWS provides a default role, or you can create a custom IAM role.
If you are copying encrypted snapshots, you must grant the selected IAM role permissions to use the encryption KMS key used to encrypt the source volume. Similarly, if you are encrypting the snapshot in the destination Region using a different KMS key, you must grant the IAM role permission to use the destination KMS key. For more information, see Step 4: Allow IAM role to use the required KMS keys (Target account) (p. 1295).

7. In the Copy settings section, you can configure the policy to copy snapshots to up to three Regions in the target account. Do the following:
   a. For Name, enter a descriptive name for the copy action.
   b. For Target Region, select the Region to which to copy the snapshots.
   c. For Retain copy for, specify how long to retain the snapshot copies in the target Region after creation.
   d. For Encryption, select Enable to encrypt the snapshot copy in the target Region. If the source snapshot is encrypted, or if encryption by default is enabled for your account, the snapshot copy is always encrypted, even if you do enable encryption here. If the source snapshot is unencrypted and encryption by default is not enabled for your account, you can choose to enable or disable encryption. If you enable encryption, but do not specify a KMS key, the snapshots are encrypted using the default encryption KMS key in each destination Region. If you specify a KMS key for the destination Region, you must have access to the KMS key.
   e. (Optional) To copy the snapshot to additional Regions, choose Add additional region, and then complete the required fields.

8. For Policy status after creation, choose Enable policy to start the policy runs at the next scheduled time.

9. Choose Create Policy.

Command line

Use the create-lifecycle-policy command to create a policy. To create a cross-account copy event policy, for PolicyType, specify EVENT_BASED_POLICY.

For example, the following command creates a cross-account copy event policy in target account 222222222222. The policy copies snapshots that are shared by source account 111111111111. The policy copies snapshots to sa-east-1 and eu-west-2. Snapshots copied to sa-east-1 are unencrypted and they are retained for 3 days. Snapshots copied to eu-west-2 are encrypted using KMS key 8af79514-350d-4c52-bac8-8985e84171c7 and they are retained for 1 month. The policy uses the default IAM role.

```bash
$ aws dlm create-lifecycle-policy
--description "Copy policy"
--state ENABLED --execution-role-arn arn:aws:iam::222222222222:role/service-role/AWSDataLifecycleManagerDefaultRole
--policy-details file://policyDetails.json
```

The following shows the contents of the policyDetails.json file.

```json
{
   "PolicyType" : "EVENT_BASED_POLICY",
   "EventSource" : {
      "Type" : "MANAGED_CWE",
      "Parameters": {
         "EventType" : "shareSnapshot",
         "SnapshotOwner": ["111111111111"]
      }
   },
   "Actions" : [{
```
Upon success, the command returns the ID of the newly created policy. The following is example output.

```json
{
  "PolicyId": "policy-9876543210abcdef0"
}
```

**Step 4: Allow IAM role to use the required KMS keys (Target account)**

If you are copying encrypted snapshots, you must grant the IAM role (that you selected in the previous step) permissions to use the customer managed key that was used to encrypt the source volume.

**Note**

Only perform this step if you are copying encrypted snapshots. If you are copying unencrypted snapshots, skip this step.

Use one of the following methods to add the required policies to the IAM role.

**Console**

2. In the navigation pane, select **Roles**. Search for and select the IAM role that you selected when you created the cross-account copy event policy in the previous step. If you chose to use the default role, the role is named **AWSDataLifecycleManagerDefaultRole**.
3. Choose **Add inline policy** and then select the **JSON** tab.
4. Replace the existing policy with the following, and specify the ARNs of the KMS keys.

   ```json
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": [
   ```
5. Choose **Review policy**

6. For **Name**, enter a descriptive name for the policy, and then choose **Create policy**.

**Command line**

Using your preferred text editor, create a new JSON file named `policyDetails.json`. Add the following policy and specify the ARNs of the KMS keys that the role needs permissions to use. In the following example, the policy grants the IAM role permission to use KMS key 1234abcd-12ab-34cd-56ef-1234567890ab, which was shared by source account 111111111111, and KMS key 4567dcba-23ab-34cd-56ef-0987654321yz, which exists in target account 222222222222.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:RevokeGrant",
                "kms:CreateGrant",
                "kms:ListGrants"
            ],
            "Resource": [
                "arn:aws:kms:sa-east-1:111111111111:key/1234abcd-12ab-34cd-56ef-1234567890ab",
                "arn:aws:kms:eu-west-2:222222222222:key/4567dcba-23ab-34cd-56ef-0987654321yz"
            ],
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        }
    ]
}
```
Save and close the file. Then use the `put-role-policy` command to add the policy to the IAM role.

For example

```
# aws iam put-role-policy
--role-name AWSDataLifecycleManagerDefaultRole
--policy-name CopyPolicy
--policy-document file://AdminPolicy.json
```

### Specify snapshot description filters

When you create the snapshot copy policy in the target account, you must specify a snapshot description filter. The snapshot description filter enables you to specify an additional level of filtering that lets you control which snapshots are copied by the policy. This means that a snapshot is only copied by the policy if it is shared by one of the specified source accounts, and it has a snapshot description that matches the specified filter. In other words, if a snapshot is shared by one of the specified source accounts, but it does not have a description that matches the specified filter, it is not copied by the policy.

The snapshot filter description must be specified using a regular expression. It is a mandatory field when creating cross-account copy event policies using the console and the command line. The following are example regular expressions that can be used:

- `.*`—This filter matches all snapshot descriptions. If you use this expression the policy will copy all snapshots that are shared by one of the specified source accounts.
- `Created for policy: policy-0123456789abcdef0.*`—This filter matches only snapshots that are created by a policy with an ID of `policy-0123456789abcdef0`. If you use an expression like this, only snapshots that are shared with your account by one of the specified source accounts, and that have been created by a policy with the specified ID are copied by the policy.
- `.*production.*`—This filter matches any snapshot that has the word `production` anywhere in its description. If you use this expression the policy will copy all snapshots that are shared by one of the specified source accounts and that have the specified text in their description.

### View, modify, and delete lifecycle policies

Use the following procedures to view, modify and delete existing lifecycle policies.
• View lifecycle policies (p. 1298)
• Modify lifecycle policies (p. 1299)
• Delete lifecycle policies (p. 1300)

View lifecycle policies

Use one of the following procedures to view a lifecycle policy.

Console

To view a lifecycle policy

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Elastic Block Store, Lifecycle Manager.
3. Select a lifecycle policy from the list. The Details tab displays information about the policy.

Command line

Use the get-lifecycle-policy command to display information about a lifecycle policy.

```
aws dlm get-lifecycle-policy --policy-id policy-0123456789abcdef0
```

The following is example output. It includes the information that you specified, plus metadata inserted by AWS.

```
{
   "Policy":{
      "Description": "My first policy",
      "DateCreated": "2018-05-15T00:16:21+0000",
      "State": "ENABLED",
      "ExecutionRoleArn": "arn:aws:iam::210774411744:role/AWSDataLifecycleManagerDefaultRole",
      "PolicyId": "policy-0123456789abcdef0",
      "DateModified": "2018-05-15T00:16:22+0000",
      "PolicyDetails": {
         "PolicyType":"EBS_SNAPSHOT_MANAGEMENT",
         "ResourceTypes": ["VOLUME"],
         "TargetTags": [ {
            "Value": "115",
            "Key": "costcenter"
         } ],
         "Schedules": [ {
            "TagsToAdd": [ {
               "Value": "myDailySnapshot",
               "Key": "type"
            } ],
            "RetainRule": { "Count": 5 },
            "CopyTags": false,
            "CreateRule": { "Interval": 24, "IntervalUnit": "HOURS" }]
   }
}
```
Modify lifecycle policies

Use one of the following procedures to modify a lifecycle policy.

Console

**To modify a lifecycle policy**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](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 Lifecycle Policy**.
5. Modify the policy settings as needed. For example, you can modify the schedule, add or remove tags, or enable or disable the policy.
6. Choose **Update policy**.

Command line

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.

```json
{
   "ResourceTypes": [
      "VOLUME"
   ],
   "TargetTags": [
      {
         "Key": "costcenter",
         "Value": "120"
      }
   ],
   "Schedules": [
      {
         "Name": "DailySnapshots",
         "TagsToAdd": [
            {
               "Key": "type",
               "Value": "myDailySnapshot"
            }
         ]
      }
   ]
}
```
"CreateRule": {
  "Interval": 12,
  "IntervalUnit": "HOURS",
  "Times": [
    "15:00"
  ]
},
"RetainRule": {
  "Count": 5
},
"CopyTags": false
}

To view the updated policy, use the get-lifecycle-policy command. You can see that the state, the value of the tag, the snapshot interval, and the snapshot start time were changed.

**Delete lifecycle policies**

Use one of the following procedures to delete a lifecycle policy.

**Note**
When you delete a lifecycle policy, the snapshots or AMIs created by that policy are not automatically deleted. If you no longer need the snapshots or AMIs, you must delete them manually.

**Old console**

**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 Lifecycle Policy.
5. When prompted for confirmation, choose Delete Lifecycle Policy.

**Command line**

Use the `delete-lifecycle-policy` command to delete a lifecycle policy and free up the target tags specified in the policy for reuse.

**Note**
You can delete snapshots created only by Amazon Data Lifecycle Manager.

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

The Amazon Data Lifecycle Manager API Reference provides descriptions and syntax for each of the actions and data types for the Amazon Data Lifecycle Manager Query API.

Alternatively, you can use one of the AWS SDKs to access the API in a way that's tailored to the programming language or platform that you're using. For more information, see AWS SDKs.

**AWS Identity and Access Management**

Access to Amazon Data Lifecycle Manager requires credentials. Those credentials must have permissions to access AWS resources, such as instances, volumes, snapshots, and AMIs. The following sections provide
details about how you can use AWS Identity and Access Management (IAM), and help secure access to your resources.

**Topics**
- AWS managed policies (p. 1301)
- IAM service roles (p. 1303)
- Permissions for IAM users (p. 1306)
- Permissions for encryption (p. 1306)

**AWS managed policies**

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases. AWS managed policies make it more efficient for you to assign appropriate permissions to users, groups, and roles, than if you had to write the policies yourself.

However, you can't change the permissions defined in AWS managed policies. AWS occasionally updates the permissions defined in an AWS managed policy. When this occurs, the update affects all principal entities (users, groups, and roles) that the policy is attached to.

Amazon Data Lifecycle Manager provides two AWS managed policies for common use cases. These policies make it more efficient to define the appropriate permissions and control access to your resources. The AWS managed policies provided by Amazon Data Lifecycle Manager are designed to be attached to roles that you pass to Amazon Data Lifecycle Manager.

The following are the AWS managed policies that Amazon Data Lifecycle Manager provides. You can also find these AWS managed policies in the **Policies** section of the IAM console.

**AWSDataLifecycleManagerServiceRole**

The **AWSDataLifecycleManagerServiceRole** policy provides appropriate permissions to Amazon Data Lifecycle Manager to create and manage Amazon EBS snapshot policies and cross-account copy event policies.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots",
                "ec2:DeleteSnapshot",
                "ec2:DescribeInstances",
                "ec2:DescribeVolumes",
                "ec2:DescribeSnapshots",
                "ec2:EnableFastSnapshotRestores",
                "ec2:DescribeFastSnapshotRestores",
                "ec2:DisableFastSnapshotRestores",
                "ec2:CopySnapshot",
                "ec2:ModifySnapshotAttribute",
                "ec2:DescribeSnapshotAttribute"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "*"
        }
    ]
}
```
AWSDataLifecycleManagerServiceRoleForAMIManagement

The **AWSDataLifecycleManagerServiceRoleForAMIManagement** policy provides appropriate permissions to Amazon Data Lifecycle Manager to create and manage Amazon EBS-backed AMI policies.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags",
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
                "ec2:DescribeImageAttribute",
                "ec2:DescribeVolumes",
                "ec2:DescribeSnapshots",
                "ec2:EnableImageDeprecation",
                "ec2:DisableImageDeprecation"
            ],
            "Resource": ["arn:aws:ec2:*::snapshot/**", "arn:aws:ec2:*::image/**"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteSnapshot",
                "ec2:ResetImageAttribute",
                "ec2:DeregisterImage",
                "ec2:CreateImage",
                "ec2:CopyImage",
                "ec2:ModifyImageAttribute"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
        },
        {
            "Effect": "Allow",
        },
        {
            "Effect": "Allow",
```
AWS managed policy updates

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

The following table provides details about updates to AWS managed policies for Amazon Data Lifecycle Manager since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Document history (p. 1592).

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWSDataLifecycleManagerServiceRoleForAMIManagement — Added permissions to support AMI deprecation.</td>
<td>Amazon Data Lifecycle Manager added the ec2:EnableImageDeprecation and ec2:DisableImageDeprecation actions to grant EBS-backed AMI policies permission to enable and disable AMI deprecation.</td>
<td>August 23, 2021</td>
</tr>
<tr>
<td>Amazon Data Lifecycle Manager started tracking changes</td>
<td>Amazon Data Lifecycle Manager started tracking changes for its AWS managed policies.</td>
<td>August 23, 2021</td>
</tr>
</tbody>
</table>

IAM service roles

An AWS Identity and Access Management (IAM) role is similar to a user, in that it is an AWS identity with permissions policies that determine what the identity can and can't do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. A service role is a role that an AWS service assumes to perform actions on your behalf. As a service that performs backup operations on your behalf, Amazon Data Lifecycle Manager requires that you pass it a role to assume when performing policy operations on your behalf. For more information about IAM roles, see IAM Roles in the IAM User Guide.

The role that you pass to Amazon Data Lifecycle Manager must have an IAM policy with the permissions that enable Amazon Data Lifecycle Manager to perform actions associated with policy operations, such as creating snapshots and AMIs, copying snapshots and AMIs, deleting snapshots, and deregistering AMIs. Different permissions are required for each of the Amazon Data Lifecycle Manager policy types. The role must also have Amazon Data Lifecycle Manager listed as a trusted entity, which enables Amazon Data Lifecycle Manager to assume the role.
Topics

- Default service roles for Amazon Data Lifecycle Manager (p. 1304)
- Custom service roles for Amazon Data Lifecycle Manager (p. 1304)

Default service roles for Amazon Data Lifecycle Manager

Amazon Data Lifecycle Manager uses the following default service roles:

- **AWSDataLifecycleManagerDefaultRole**—default role for managing snapshots. It trusts only the dlm.amazonaws.com service to assume the role and it allows Amazon Data Lifecycle Manager to perform the actions required by snapshot and cross-account snapshot copy policies on your behalf. This role uses the AWSDataLifecycleManagerServiceRole AWS managed policy.

- **AWSDataLifecycleManagerDefaultRoleForAMIManagement**—default role for managing AMIs. It trusts only the dlm.amazonaws.com service to assume the role and it allows Amazon Data Lifecycle Manager to perform the actions required by EBS-backed AMI policies on your behalf. This role uses the AWSDataLifecycleManagerServiceRoleForAMIManagement AWS managed policy.

If you are using the Amazon Data Lifecycle Manager console, Amazon Data Lifecycle Manager automatically creates the **AWSDataLifecycleManagerDefaultRole** service role the first time you create a snapshot or cross-account snapshot copy policy, and it automatically creates the **AWSDataLifecycleManagerDefaultRoleForAMIManagement** service role the first time you create an EBS-backed AMI policy.

If you delete the default service roles, and then need to create them again, you can use the same process to recreate them in your account.

Custom service roles for Amazon Data Lifecycle Manager

As an alternative to using the default service roles, you can create custom IAM roles with the required permissions and then select them when you create a lifecycle policy.

To create a custom IAM role

1. Create roles with the following permissions.

   - Permissions required for managing snapshot lifecycle policies

     ```json
     {
       "Version": "2012-10-17",
       "Statement": [
         {
           "Effect": "Allow",

         }
     ```
Permissions required for managing AMI lifecycle policies

```json
{
    "Version": "2012-10-17",
    "Statement": [  
        {
            "Effect": "Allow",
            "Action": [  
                "ec2:CreateTags",
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
                "ec2:DescribeImageAttribute",
                "ec2:DescribeVolumes",
                "ec2:DescribeSnapshots"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [  
                "ec2:DeleteSnapshot",
                "ec2:DeleteInstance"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*"
        },
        {
            "Effect": "Allow",
            "Action": [  
                "ec2:ResetImageAttribute",
                "ec2:DeregisterImage",
                "ec2:CreateImage",
                "ec2:CreateInstance"
            ],
            "Resource": "*"
        }
    ]
}
For more information, see Creating a Role in the IAM User Guide.

2. Add a trust relationship to the roles.
   a. In the IAM console, choose Roles.
   b. Select the roles that you created, and then choose Trust relationships.
   c. Choose Edit Trust Relationship, add the following policy, and then choose Update Trust Policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "dlm.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Permissions for IAM users

An IAM user must have the following permissions to use Amazon Data Lifecycle Manager.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["iam:PassRole", "iam:ListRoles"],
      "Resource": "arn:aws:iam::123456789012:role/AWSDataLifecycleManagerDefaultRole"
    },
    {
      "Effect": "Allow",
      "Action": ["dlm:*"],
      "Resource": "*"
    }
  ]
}
```

For more information, see Changing Permissions for an IAM User in the IAM User Guide.

Permissions for encryption

If the source volume is encrypted, ensure that the Amazon Data Lifecycle Manager default roles (AWSDataLifecycleManagerDefaultRole and AWSDataLifecycleManagerDefaultRoleForAMIManagement) have permission to use the KMS keys used to encrypt the volume.

If you enable Cross Region copy for unencrypted snapshots or AMIs backed by unencrypted snapshots, and choose to enable encryption in the destination Region, ensure that the default roles have permission to use the KMS key needed to perform the encryption in the destination Region.
If you enable **Cross Region copy** for encrypted snapshots or AMIs backed by encrypted snapshots, ensure that the default roles have permission to use both the source and destination KMS keys.

For more information, see [Allowing users in other accounts to use a KMS key](https://docs.aws.amazon.com/keymanagement/latest/devguide/userguide.html) in the *AWS Key Management Service Developer Guide*.

**Monitor the lifecycle of snapshots and AMIs**

You can use the following features to monitor the lifecycle of your snapshots and AMIs.

**Features**

- Console and AWS CLI (p. 1307)
- AWS CloudTrail (p. 1307)
- Monitor your policies using CloudWatch Events (p. 1307)
- Monitor your policies using Amazon CloudWatch (p. 1308)

**Console and AWS CLI**

You can view your lifecycle policies using the Amazon EC2 console or the AWS CLI. Each snapshot and AMI created by a policy has a timestamp and policy-related tags. You can filter snapshots and AMIs using these tags to verify that your backups are being created as you intend. For information about viewing lifecycle policies using the console, see View lifecycle policies (p. 1298).

**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*](https://docs.aws.amazon.com/AmazonCloudTrail/latest/userguide/)

**Monitor your policies using CloudWatch Events**

Amazon EBS and Amazon Data Lifecycle Manager emit events related to lifecycle policy actions. You can use AWS Lambda and Amazon CloudWatch Events to handle event notifications programmatically. Events are emitted on a best effort basis. For more information, see the [*Amazon CloudWatch Events User Guide*](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/events.html).

The following events are available:

**Note**

No events are emitted for AMI lifecycle policy actions.

- **createSnapshot**—An Amazon EBS event emitted when a `CreateSnapshot` action succeeds or fails. For more information, see [*Amazon CloudWatch Events for Amazon EBS*](https://docs.aws.amazon.com/AmazonEBS/latest/APIReference/API_CreateSnapshot.html).
- **DLM Policy State Change**—An Amazon Data Lifecycle Manager event emitted when a lifecycle policy enters an error state. The event contains a description of what caused the error. The following is an example of an event when the permissions granted by the IAM role are insufficient.

```json
{
   "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"
   ]
}
```
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"
 }
}
```

Monitor your policies using Amazon CloudWatch

You can monitor your Amazon Data Lifecycle Manager lifecycle policies using CloudWatch, which collects raw data and processes it into readable, near real-time metrics. You can use these metrics to see exactly how many Amazon EBS snapshots and EBS-backed AMIs are created, deleted, and copied by your policies over time. You can also set alarms that watch for certain thresholds, and send notifications or take actions when those thresholds are met.

Metrics are kept for a period of 15 months, so that you can access historical information and gain a better understanding of how your lifecycle policies perform over an extended period.

For more information about Amazon CloudWatch, see the Amazon CloudWatch User Guide.

Topics

- Supported metrics (p. 1308)
- View CloudWatch metrics for your policies (p. 1311)
- Graph metrics for your policies (p. 1312)
- Create a CloudWatch alarm for a policy (p. 1313)
- Example use cases (p. 119)
- Managing policies that report failed actions (p. 1314)

Supported metrics

The Data Lifecycle Manager namespace includes the following metrics for Amazon Data Lifecycle Manager lifecycle policies. The supported metrics differ by policy type.

All metrics can be measured on the DLMPolicyId dimension. The most useful statistics are sum and average, and the unit of measure is count.

Choose a tab to view the metrics supported by that policy type.
### EBS snapshot policies

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResourcesTargeted</td>
<td>The number of resources targeted by the tags specified in a snapshot or EBS-backed AMI policy.</td>
</tr>
<tr>
<td>SnapshotsCreateStarted</td>
<td>The number of snapshot create actions initiated by a snapshot policy. Each action is recorded only once, even if there are multiple subsequent retries. If a snapshot create action fails, Amazon Data Lifecycle Manager sends a SnapshotsCreateFailed metric.</td>
</tr>
<tr>
<td>SnapshotsCreateCompleted</td>
<td>The number of snapshots created by a snapshot policy. This includes successful retries within 60 minutes of the scheduled time.</td>
</tr>
<tr>
<td>SnapshotsCreateFailed</td>
<td>The number of snapshots that could not be created by a snapshot policy. This includes unsuccessful retries within 60 minutes from the scheduled time.</td>
</tr>
<tr>
<td>SnapshotsSharedCompleted</td>
<td>The number of snapshots shared across accounts by a snapshot policy.</td>
</tr>
<tr>
<td>SnapshotsDeleteCompleted</td>
<td>The number of snapshots deleted by a snapshot or EBS-backed AMI policy. This metric applies only to snapshots created by the policy. It does not apply to cross-Region snapshot copies created by the policy. This metric includes snapshots that are deleted when an EBS-backed AMI policy deregisters AMIs.</td>
</tr>
<tr>
<td>SnapshotsDeleteFailed</td>
<td>The number of snapshots that could not be deleted by a snapshot or EBS-backed AMI policy. This metric applies only to snapshots created by the policy. This metric includes snapshots that are deleted when an EBS-backed AMI policy deregisters AMIs.</td>
</tr>
<tr>
<td>SnapshotsCopiedRegionStarted</td>
<td>The number of cross-Region snapshot copy actions initiated by a snapshot policy.</td>
</tr>
<tr>
<td>SnapshotsCopiedRegionCompleted</td>
<td>The number of cross-Region snapshot copies created by a snapshot policy. This includes successful retries within 24 hours of the scheduled time.</td>
</tr>
<tr>
<td>SnapshotsCopiedRegionFailed</td>
<td>The number of cross-Region snapshot copies that could not be created by a snapshot policy. This includes unsuccessful retries within 24 hours from the scheduled time.</td>
</tr>
<tr>
<td>SnapshotsCopiedRegionDeleteCompleted</td>
<td>The number of cross-Region snapshot copies deleted, as designated by the retention rule, by a snapshot policy.</td>
</tr>
<tr>
<td>SnapshotsCopiedRegionDeleteFailed</td>
<td>The number of cross-Region snapshot copies that could not be deleted, as designated by the retention rule, by a snapshot policy.</td>
</tr>
</tbody>
</table>

### EBS-backed AMI policies

The following metrics can be used with EBS-backed AMI policies:
<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResourcesTargeted</td>
<td>The number of resources targeted by the tags specified in a snapshot or EBS-backed AMI policy.</td>
</tr>
<tr>
<td>SnapshotsDeleteCompleted</td>
<td>The number of snapshots deleted by a snapshot or EBS-backed AMI policy. This metric applies only to snapshots created by the policy. It does not apply to cross-Region snapshot copies created by the policy. This metric includes snapshots that are deleted when an EBS-backed AMI policy deregisters AMIs.</td>
</tr>
<tr>
<td>SnapshotsDeleteFailed</td>
<td>The number of snapshots that could not be deleted by a snapshot or EBS-backed AMI policy. This metric applies only to snapshots created by the policy. It does not apply to cross-Region snapshot copies created by the policy. This metric includes snapshots that are deleted when an EBS-backed AMI policy deregisters AMIs.</td>
</tr>
<tr>
<td>SnapshotsCopiedRegionDeleteCompleted</td>
<td>The number of cross-Region snapshot copies deleted, as designated by the retention rule, by a snapshot policy.</td>
</tr>
<tr>
<td>SnapshotsCopiedRegionDeleteFailed</td>
<td>The number of cross-Region snapshot copies that could not be deleted, as designated by the retention rule, by a snapshot policy.</td>
</tr>
<tr>
<td>ImagesCreateStarted</td>
<td>The number of CreateImage actions initiated by an EBS-backed AMI policy.</td>
</tr>
<tr>
<td>ImagesCreateCompleted</td>
<td>The number of AMIs created by an EBS-backed AMI policy.</td>
</tr>
<tr>
<td>ImagesCreateFailed</td>
<td>The number of AMIs that could not be created by an EBS-backed AMI policy.</td>
</tr>
<tr>
<td>ImagesDeregisterCompleted</td>
<td>The number of AMIs deregistered by an EBS-backed AMI policy.</td>
</tr>
<tr>
<td>ImagesDeregisterFailed</td>
<td>The number of AMIs that could not be deregistered by an EBS-backed AMI policy.</td>
</tr>
<tr>
<td>ImagesCopiedRegion</td>
<td>The number of cross-Region copy actions initiated by an EBS-backed AMI policy.</td>
</tr>
<tr>
<td>ImagesCopiedRegionCompleted</td>
<td>The number of cross-Region AMI copies created by an EBS-backed AMI policy.</td>
</tr>
<tr>
<td>ImagesCopiedRegionFailed</td>
<td>The number of cross-Region AMI copies that could not be created by an EBS-backed AMI policy.</td>
</tr>
<tr>
<td>ImagesCopiedRegionDeregisterCompleted</td>
<td>The number of cross-Region AMI copies deregistered, as designated by the retention rule, by an EBS-backed AMI policy.</td>
</tr>
<tr>
<td>ImagesCopiedRegionDeregisterFailed</td>
<td>The number of cross-Region AMI copies that could not be deregistered, as designated by the retention rule, by an EBS-backed AMI policy.</td>
</tr>
<tr>
<td>EnableImageDeprecationCompleted</td>
<td>The number of AMIs that were marked for deprecation by an EBS-backed AMI policy.</td>
</tr>
<tr>
<td>EnableImageDeprecationFailed</td>
<td>The number of AMIs that could not be marked for deprecation by an EBS-backed AMI policy.</td>
</tr>
</tbody>
</table>
### Cross-account copy event policies

The following metrics can be used with cross-account copy event policies:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SnapshotsCopiedAccountStarted</td>
<td>The number of cross-account snapshot copy actions initiated by a cross-account copy event policy.</td>
</tr>
<tr>
<td>SnapshotsCopiedAccountCompleted</td>
<td>The number of snapshots copied from another account by a cross-account copy event policy. This includes successful retries within 24 hours of the scheduled time.</td>
</tr>
<tr>
<td>SnapshotsCopiedAccountFailed</td>
<td>The number of snapshots that could not be copied from another account by a cross-account copy event policy. This includes unsuccessful retries within 24 hours of the scheduled time.</td>
</tr>
<tr>
<td>SnapshotsCopiedAccountDeleteCompleted</td>
<td>The number of cross-Region snapshot copies deleted, as designated by the retention rule, by a cross-account copy event policy.</td>
</tr>
<tr>
<td>SnapshotsCopiedAccountDeleteFailed</td>
<td>The number of cross-Region snapshot copies that could not be deleted, as designated by the retention rule, by a cross-account copy event policy.</td>
</tr>
</tbody>
</table>

### View CloudWatch metrics for your policies

You can use the AWS Management Console or the command line tools to list the metrics that Amazon Data Lifecycle Manager sends to Amazon CloudWatch.

**Amazon EC2 console**

**To view metrics using the Amazon EC2 console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Lifecycle Manager**.
3. Select a policy in the grid and then choose the **Monitoring** tab.

**CloudWatch console**

**To view metrics using the Amazon CloudWatch console**

2. In the navigation pane, choose **Metrics**.
3. Select the **EBS** namespace and then select **Data Lifecycle Manager metrics**.

**AWS CLI**

**To list all the available metrics for Amazon Data Lifecycle Manager**
Use the `list-metrics` command.

```
C:\> aws cloudwatch list-metrics --namespace AWS/EBS
```

**To list all the metrics for a specific policy**

Use the `list-metrics` command and specify the `DLMPolicyId` dimension.

```
C:\> aws cloudwatch list-metrics --namespace AWS/EBS --dimensions
    Name=DLMPolicyId,Value=policy-abcdef01234567890
```

**To list a single metric across all policies**

Use the `list-metrics` command and specify the `--metric-name` option.

```
C:\> aws cloudwatch list-metrics --namespace AWS/EBS --metric-name
    SnapshotsCreateCompleted
```

Graph metrics for your policies

After you create a policy, you can open the Amazon EC2 console and view the monitoring graphs for the policy on the **Monitoring** tab. Each graph is based on one of the available Amazon EC2 metrics.

The following graphs metrics are available:

- Resources targeted (based on `ResourcesTargeted`)
- Snapshot creation started (based on `SnapshotsCreateStarted`)
- Snapshot creation completed (based on `SnapshotsCreateCompleted`)
- Snapshot creation failed (based on `SnapshotsCreateFailed`)
- Snapshot sharing completed (based on `SnapshotsSharedCompleted`)
- Snapshot deletion completed (based on `SnapshotsDeleteCompleted`)
- Snapshot deletion failed (based on `SnapshotsDeleteFailed`)
- Snapshot cross-Region copy started (based on `SnapshotsCopiedRegionStarted`)
- Snapshot cross-Region copy completed (based on `SnapshotsCopiedRegionCompleted`)
- Snapshot cross-Region copy failed (based on `SnapshotsCopiedRegionFailed`)
- Snapshot cross-Region copy deletion completed (based on `SnapshotsCopiedRegionDeleteCompleted`)
- Snapshot cross-Region copy deletion failed (based on `SnapshotsCopiedRegionDeleteFailed`)
- Snapshot cross-account copy started (based on `SnapshotsCopiedAccountStarted`)
- Snapshot cross-account copy completed (based on `SnapshotsCopiedAccountCompleted`)
- Snapshot cross-account copy failed (based on `SnapshotsCopiedAccountFailed`)
- Snapshot cross-account copy deletion completed (based on `SnapshotsCopiedAccountDeleteCompleted`)
- Snapshot cross-account copy deletion failed (based on `SnapshotsCopiedAccountDeleteFailed`)
- AMI creation started (based on `ImagesCreateStarted`)
- AMI creation completed (based on `ImagesCreateCompleted`)
- AMI creation failed (based on `ImagesCreateFailed`)
- AMI deregistration completed (based on `ImagesDeregisterCompleted`)
- AMI deregistration failed (based on `ImagesDeregisterFailed`)
- AMI cross-Region copy started (based on `ImagesCopiedRegionStarted`)
• AMI cross-Region copy completed (based on ImagesCopiedRegionCompleted)
• AMI cross-Region copy failed (based on ImagesCopiedRegionFailed)
• AMI cross-Region copy deregistration completed (based on ImagesCopiedRegionDeregisterCompleted)
• AMI cross-Region copy deregister failed (based on ImagesCopiedRegionDeregisteredFailed)
• AMI enable deprecation completed (based on EnableImageDeprecationCompleted)
• AMI enable deprecation failed (based on EnableImageDeprecationFailed)
• AMI cross-Region copy enable deprecation completed (based on EnableCopiedImageDeprecationCompleted)
• AMI cross-Region copy enable deprecation failed (based on EnableCopiedImageDeprecationFailed)

Create a CloudWatch alarm for a policy

You can create a CloudWatch alarm that monitors CloudWatch metrics for your policies. CloudWatch will automatically send you a notification when the metric reaches a threshold that you specify. You can create a CloudWatch alarm using the CloudWatch console.

For more information about creating alarms using the CloudWatch console, see the following topic in the Amazon CloudWatch User Guide.

• Create a CloudWatch Alarm Based on a Static Threshold
• Create a CloudWatch Alarm Based on Anomaly Detection

Example use cases

The following are example use cases.

Topics
• Example 1: ResourcesTargeted metric (p. 1313)
• Example 2: SnapshotDeleteFailed metric (p. 1314)
• Example 3: SnapshotsCopiedRegionFailed metric (p. 1314)

Example 1: ResourcesTargeted metric

You can use the ResourcesTargeted metric to monitor the total number of resources that are targeted by a specific policy each time it is run. This enables you to trigger an alarm when the number of targeted resources is below or above an expected threshold.

For example, if you expect your daily policy to create backups of no more than 50 volumes, you can create an alarm that sends an email notification when the sum for ResourcesTargeted is greater than 50 over a 1 hour period. In this way, you can ensure that no snapshots have been unexpectedly created from volumes that have been incorrectly tagged.

You can use the following command to create this alarm:

C:\> aws cloudwatch put-metric-alarm \ 
--alarm-name resource-targeted-monitor \ 
--alarm-description "Alarm when policy targets more than 50 resources" \ 
--metric-name ResourcesTargeted \ 
--namespace AWS/EBS \ 
--statistic Sum \ 
--period 3600 \ 
--threshold 50 \ 
--comparison-operator GreaterThanThreshold \ 

1313
Example 2: SnapshotDeleteFailed metric

You can use the SnapshotDeleteFailed metric to monitor for failures to delete snapshots as per the policy's snapshot retention rule.

For example, if you've created a policy that should automatically delete snapshots every twelve hours, you can create an alarm that notifies your engineering team when the sum of SnapshotDeletionFailed is greater than 0 over a 1 hour period. This could help to investigate improper snapshot retention and to ensure that your storage costs are not increased by unnecessary snapshots.

You can use the following command to create this alarm:

```bash
C:\> aws cloudwatch put-metric-alarm \
--alarm-name snapshot-deletion-failed-monitor \
--alarm-description "Alarm when snapshot deletions fail" \
--metric-name SnapshotsDeleteFailed \
--namespace AWS/EBS \
--statistic Sum \
--period 3600 \
--threshold 0 \
--comparison-operator GreaterThanThreshold \
--dimensions "Name=DLMPolicyId,Value=policy_id" \
--evaluation-periods 1 \
--alarm-actions sns_topic_arn
```

Example 3: SnapshotsCopiedRegionFailed metric

Use the SnapshotsCopiedRegionFailed metric to identify when your policies fail to copy snapshots to other Regions.

For example, if your policy copies snapshots across Regions daily, you can create an alarm that sends an SMS to your engineering team when the sum of SnapshotCrossRegionCopyFailed is greater than 0 over a 1 hour period. This can be useful for verifying whether subsequent snapshots in the lineage were successfully copied by the policy.

You can use the following command to create this alarm:

```bash
C:\> aws cloudwatch put-metric-alarm \
--alarm-name snapshot-copy-region-failed-monitor \
--alarm-description "Alarm when snapshot copy fails" \
--metric-name SnapshotsCopiedRegionFailed \
--namespace AWS/EBS \
--statistic Sum \
--period 3600 \
--threshold 0 \
--comparison-operator GreaterThanThreshold \
--dimensions "Name=DLMPolicyId,Value=policy_id" \
--evaluation-periods 1 \
--alarm-actions sns_topic_arn
```

Managing policies that report failed actions

For more information about what to do when one of your policies reports an unexpected non-zero value for a failed action metric, see the What should I do if Amazon Data Lifecycle Manager reports failed actions in CloudWatch metrics? AWS Knowledge Center article.
Amazon EBS data services

Amazon EBS provides the following data services.

Data services
- Amazon EBS Elastic Volumes (p. 1315)
- Amazon EBS encryption (p. 1327)
- Amazon EBS fast snapshot restore (p. 1339)

Amazon EBS Elastic Volumes

With Amazon EBS Elastic Volumes, you can increase the volume size, change the volume type, or adjust the performance of your EBS volumes. If your instance supports Elastic Volumes, you can do so without detaching the volume or restarting the instance. This enables you to continue using your application while the changes take effect.

There is no charge to modify the configuration of a volume. You are charged for the new volume configuration after volume modification starts. For more information, see the Amazon EBS Pricing page.

Contents
- Requirements when modifying volumes (p. 1315)
- Request modifications to your EBS volumes (p. 1317)
- Monitor the progress of volume modifications (p. 1320)
- Extend a Windows file system after resizing a volume (p. 1323)

Requirements when modifying volumes

The following requirements and limitations apply when you modify an Amazon EBS volume. To learn more about the general requirements for EBS volumes, see Constraints on the size and configuration of an EBS volume (p. 1181).

Topics
- Supported instance types (p. 1315)
- Requirements for Windows volumes (p. 1315)
- Limitations (p. 1316)

Supported instance types

Elastic Volumes are supported on the following instances:
- All current-generation instances (p. 142)
- The following previous-generation instances: C1, C3, CC2, CR1, G2, I2, M1, M3, and R3

If your instance type does not support Elastic Volumes, see Modify an EBS volume if Elastic Volumes is not supported (p. 1320).

Requirements 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 requirements affecting Linux volumes, see Requirements 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.

**Limitations**

- There are limits to the maximum aggregated storage that can be requested across volume modifications. For more information, see Amazon EBS service quotas in the *Amazon Web Services General Reference*.

- After modifying a volume, you must wait at least six hours and ensure that the volume is in the *in-use* or *available* state before you can modify the same volume. This is sometimes referred to as a cooldown period.

- If the volume was attached before November 3, 2016 23:40 UTC, you must initialize Elastic Volumes support. For more information, see Initializing Elastic Volumes Support (p. 1318).

- If you encounter an error message while attempting to modify an EBS volume, or if you are modifying an EBS volume attached to a previous-generation instance type, take one of the following steps:
  
  - For a non-root volume, detach the volume from the instance, apply the modifications, and then re-attach the volume.
  
  - For a root volume, stop the instance, apply the modifications, and then restart the instance.

- Modification time is increased for volumes that are not fully initialized. For more information see Initialize Amazon EBS volumes (p. 1366).

- The new volume size can't exceed the supported capacity of its file system and partitioning scheme. For more information, see Constraints on the size and configuration of an EBS volume (p. 1181).

- If you modify the volume type of a volume, the size and performance must be within the limits of the target volume type. For more information, see Amazon EBS volume types (p. 1163)

- You can't decrease the size of an EBS volume. However, you can create a smaller volume and then migrate your data to it using an application-level tool such as robocopy.

- After provisioning over 32,000 IOPS on an existing io1 or io2 volume, you might need to detach and re-attach the volume, or restart the instance to see the full performance improvements.

- For io2 volumes, you can't increase the size beyond 16 TiB or the IOPS beyond 64,000 while the volume is attached to an instance type that does not support io2 Block Express volumes. Currently,
only R5b instances support io2 Block Express volumes. For more information, see io2 Block Express volumes (p. 1172)

- You can't modify the size or provisioned IOPS of an io2 volume that is attached to an R5B instance.
- You can't modify the volume type of Multi-Attach enabled io2 volumes.
- You can't modify the volume type, size, or Provisioned IOPS of Multi-Attach enabled io1 volumes.
- A gp2 volume that is attached to an instance as a root volume can't be modified to an st1 or sc1 volume. If detached and modified to st1 or sc1, it can't be re-attached to an instance as the root volume.
- While m3.medium instances fully support volume modification, m3.large, m3.xlarge, and m3.2xlarge instances might not support all volume modification features.

Request modifications to your EBS volumes

With Elastic Volumes, you can dynamically increase the size, increase or decrease the performance, and change the volume type of your Amazon EBS volumes without detaching them.

Use the following process when modifying a volume:

1. (Optional) Before modifying a volume that contains valuable data, it is a best practice to create a snapshot of the volume in case you need to roll back your changes. For more information, see Create Amazon EBS snapshots (p. 1211).

2. Request the volume modification.

3. Monitor the progress of the volume modification. For more information, see Monitor the progress of volume modifications (p. 1320).

4. If the size of the volume was modified, extend the volume's file system to take advantage of the increased storage capacity. For more information, see Extend a Windows file system after resizing a volume (p. 1323).

Contents

- Modify an EBS volume using Elastic Volumes (p. 1317)
- Initialize Elastic Volumes support (if needed) (p. 1318)
- Modify an EBS volume if Elastic Volumes is not supported (p. 1320)

Modify an EBS volume using Elastic Volumes

You can only increase volume size. You can increase or decrease volume performance. If you are not changing the volume type, then volume size and performance modifications must be within the limits of the current volume type. If you are changing the volume type, then volume size and performance modifications must be within the limits of the target volume type.

To modify an EBS volume, use one of the following methods.

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, IOPS, and throughput. Set new configuration values as follows:
   - To modify the type, choose a value for Volume Type.
• To modify the size, enter a new value for **Size**.
• To modify the IOPS, if the volume type is gp3, io1, or io2, enter a new value for **IOPS**.
• To modify the throughput, if the volume type is gp3, enter a new value for **Throughput**.

4. After you have finished changing the volume settings, choose **Modify**. When prompted for confirmation, choose **Yes**.

5. Modifying volume size has no practical effect until you also extend the volume's file system to make use of the new storage capacity. For more information, see **Extend a Windows file system after resizing a volume (p. 1323)**.

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. 538)**.

**AWS CLI**

**To modify an EBS volume using the AWS CLI**

Use the `modify-volume` command to modify one or more configuration settings for a volume. For example, if you have a volume of type gp2 with a size of 100 GiB, the following command changes its configuration to a volume of type io1 with 10,000 IOPS and a size of 200 GiB.

```
aws ec2 modify-volume --volume-type io1 --iops 10000 --size 200 --volume-id vol-11111111111111111
```

The following is example output:

```
{
    "VolumeModification": {
        "TargetSize": 200,
        "TargetVolumeType": "io1",
        "ModificationState": "modifying",
        "VolumeId": "vol-11111111111111111",
        "TargetIops": 10000,
        "StartTime": "2017-01-19T22:21:02.959Z",
        "Progress": 0,
        "OriginalVolumeType": "gp2",
        "OriginalIops": 300,
        "OriginalSize": 100
    }
}
```

Modifying volume size has no practical effect until you also extend the volume's file system to make use of the new storage capacity. For more information, see **Extend a Windows file system after resizing a volume (p. 1323)**.

**Initialize Elastic Volumes support (if needed)**

Before you can modify a volume that was attached to an instance before November 3, 2016 23:40 UTC, you must initialize volume modification support using one of the following actions:

• Detach and attach the volume
• Stop and start the instance

Use one of the following procedures to determine whether your instances are ready for volume modification.
To determine whether your instances are ready using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, choose Instances.
3. Choose the Show/Hide Columns icon (the gear). Select the Launch time attribute column and then choose Confirm.
4. Sort the list of instances by the Launch Time column. For each instance that was started before the cutoff date, choose the Storage tab and check the Attachment time column to see when its volumes were attached.

To determine whether your instances are ready using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation pane, choose Instances.
3. Choose the Show/Hide Columns icon (the gear). Select the Launch Time and Block Devices attributes and then choose Close.
4. Sort the list of instances by the Launch Time column. For instances that were started before the cutoff date, check when the devices were attached. In the following example, you must initialize volume modification for the first instance because it was started before the cutoff date and its root volume was attached before the cutoff date. The other instances are ready because they were started after the cutoff date.

AWS CLI

To determine whether your instances are ready using the CLI

Use the following describe-instances command to determine whether the volume was attached before November 3, 2016 23:40 UTC.

```bash
aws ec2 describe-instances --query "Reservations[*].Instances[*].[InstanceId,LaunchTime<='2016-11-01',BlockDeviceMappings[*] [Ebs.AttachTime<='2016-11-01']]" --output text
```

The first line of the output for each instance shows its ID and whether it was started before the cutoff date (True or False). The first line is followed by one or more lines that show whether each EBS volume was attached before the cutoff date (True or False). In the following example output, you must initialize volume modification for the first instance because it was started before the cutoff date and its root volume was attached before the cutoff date. The other instances are ready because they were started after the cutoff date.

```
i-e905622e True
i-719f99a8 False
i-006b02c1b78381e57 False
i-006b02c1b78381e57 False
```
Modify an EBS volume if Elastic Volumes is not supported

If you are using a supported instance type, you can use Elastic Volumes to dynamically modify the size, performance, and volume type of your Amazon EBS volumes without detaching them.

If you cannot use Elastic Volumes but you need to modify the root (boot) volume, you must stop the instance, modify the volume, and then restart the instance.

After the instance has started, you can check the file system size to see if your instance recognizes the larger volume space.

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 Extend a Windows file system after resizing a volume (p. 1323).

You may have to bring the volume online in order to use it. For more information, see Make an Amazon EBS volume available for use on Windows (p. 1187). You do not need to reformat the volume.

Monitor the progress of volume modifications

When you modify an EBS volume, it goes through a sequence of states. The volume enters the modifying state, the optimizing state, and finally the completed state. At this point, the volume is ready to be further modified.

Note
Rarely, a transient AWS fault can result in a failed state. This is not an indication of volume health; it merely indicates that the modification to the volume failed. If this occurs, retry the volume modification.

While the volume is in the optimizing state, your volume performance is in between the source and target configuration specifications. Transitional volume performance will be no less than the source volume performance. If you are downgrading IOPS, transitional volume performance is no less than the target volume performance.

Volume modification changes take effect as follows:

- Size changes usually take a few seconds to complete and take effect after the volume has transitioned to the Optimizing state.
- Performance (IOPS) changes can take from a few minutes to a few hours to complete and are dependent on the configuration change being made.
- It might take up to 24 hours for a new configuration to take effect, and in some cases more, such as when the volume has not been fully initialized. Typically, a fully used 1-TiB volume takes about 6 hours to migrate to a new performance configuration.

To monitor the progress of a volume modification, use one of the following methods.

Amazon EC2 console

To monitor progress of a modification using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Volumes.
3. Select the volume.
4. The State column and the State field in the details pane contain information in the following format: volume-state - modification-state (progress%). The possible volume states are creating, completed, and failed.
available, in-use, deleting, deleted, and error. The possible modification states are modifying, optimizing, and completed. Shortly after the volume modification is completed, we remove the modification state and progress, leaving only the volume state.

In this example, the modification state of the selected volume is optimizing. The modification state of the next volume is modifying.

5. Choose the text in the State field in the details pane to display information about the most recent modification action, as shown in the previous step.

AWS CLI

To monitor progress of a modification using the AWS CLI

Use the describe-volumes-modifications command to view the progress of one or more volume modifications. The following example describes the volume modifications for two volumes.

```
aws ec2 describe-volumes-modifications --volume-ids vol-11111111111111111 vol-22222222222222222
```

In the following example output, the volume modifications are still in the modifying state. Progress is reported as a percentage.

```
{
   "VolumesModifications": [
   {
      "TargetSize": 200,
      "TargetVolumeType": "io1",
      "ModificationState": "modifying",
      "VolumeId": "vol-11111111111111111",
      "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-22222222222222222"
   }
   
```

1321
The next example describes all volumes with a modification state of either optimizing or completed, and then filters and formats the results to show only modifications that were initiated on or after February 1, 2017:

```
aws ec2 describe-volumes-modifications --filters Name=modification-state,Values="optimizing","completed" --query "VolumesModifications[?StartTime>='2017-02-01'].{ID:VolumeId,STATE:ModificationState}"
```

The following is example output with information about two volumes:

```
[
  {
    "STATE": "optimizing",
    "ID": "vol-06397e7a0eEXAMPLE"
  },
  {
    "STATE": "completed",
    "ID": "vol-ba74e18c2aEXAMPLE"
  }
]
```

CloudWatch Events console

With CloudWatch Events, you can create a notification rule for volume modification events. You can use your rule to generate a notification message using Amazon SNS or to invoke a Lambda function in response to matching events. Events are emitted on a best effort basis.

To monitor progress of a modification using CloudWatch Events

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/
2. Choose Events, Create rule.
3. For Build event pattern to match events by service, choose Custom event pattern.
4. For Build custom event pattern, replace the contents with the following and choose Save.

```
{
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "EBS Volume Notification"
  ],
  "detail": {
    "event": [
      "modifyVolume"
    ]
  }
}
```

The following is example event data:
Extend a Windows file system after resizing a volume

After you increase the size of an EBS volume, use the Windows Disk Management utility or PowerShell 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. For more information about this utility, see Extend a basic volume on the Microsoft Docs website.

For more information about extending a file system on Linux, see Extend a Linux file system after resizing a volume in the Amazon EC2 User Guide for Linux Instances.

Contents

- Extend a Windows file system using the Disk Management utility (p. 1323)
- Extend a Windows file system using PowerShell (p. 1326)

Extend a Windows file system using the Disk Management utility

Use the following procedure to extend a Windows file system using Disk Management.

To extend a file system using Disk Management

1. 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 more information, see Create Amazon EBS snapshots (p. 1211).
2. Log in to your Windows instance using Remote Desktop.
3. In the Run dialog, enter diskmgmt.msc and press Enter. The Disk Management utility opens.
4. On the **Disk Management** menu, choose **Action, Rescan Disks**.

5. Open the context (right-click) menu for the expanded drive and choose **Extend Volume**.

**Note**

**Extend Volume** might be disabled (grayed out) if:

- The unallocated space is not adjacent to the drive. The unallocated space must be adjacent to the right side of the drive you want to extend.
- The volume uses the Master Boot Record (MBR) partition style and it is already 2TB in size. Volumes that use MBR cannot exceed 2TB in size.
6. 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. Generally, you specify 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.
7. To extend a file system using PowerShell, you must reboot the instance to enable Windows to see the new volume size. For more information about installing the AWS NVMe driver, see AWS NVMe drivers for Windows instances (p. 538).

**Extend a Windows file system using PowerShell**

Use the following procedure to extend a Windows file system using PowerShell.

**To extend a file system using PowerShell**

1. 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 more information, see Create Amazon EBS snapshots (p. 1211).

2. Log in to your Windows instance using Remote Desktop.

3. Run PowerShell as an administrator.

4. Run the Get-Partition command. PowerShell returns the corresponding partition number for each partition, the drive letter, offset, size, and type. Note the drive letter of the partition to extend.

5. Run the following command to rescan the disk.
"rescan" | diskpart

6. Run the following command, using the drive letter you noted in step 4 in place of `<drive-letter>`. PowerShell returns the minimum and maximum size of the partition allowed, in bytes.

```
Get-PartitionSupportedSize -DriveLetter <drive-letter>
```

7. To extend the partition, run the following command, entering the new size of the volume in place of `<size>`. You can enter the size in KB, MB, and GB; for example, 24GB.

```
Resize-Partition -DriveLetter <drive-letter> -Size <size>
```

The following shows the complete command and response flow for extending a file system using PowerShell.

Amazon EBS encryption

Use Amazon EBS encryption as a straight-forward encryption solution for your EBS resources associated with your EC2 instances. With Amazon EBS encryption, you aren't required to build, maintain, and secure your own key management infrastructure. Amazon EBS encryption uses AWS KMS keys when creating encrypted volumes and snapshots.
Encryption operations occur on the servers that host EC2 instances, ensuring the security of both data-at-rest and data-in-transit between an instance and its attached EBS storage.

You can attach both encrypted and unencrypted volumes to an instance simultaneously.

Contents
- How EBS encryption works (p. 1328)
- Requirements (p. 1329)
- Default KMS key for EBS encryption (p. 1330)
- Encryption by default (p. 1331)
- Encrypt EBS resources (p. 1332)
- Encryption scenarios (p. 1333)
- Set encryption defaults using the API and CLI (p. 1338)

How EBS encryption works

You can encrypt both the boot and data volumes of an EC2 instance.

When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

- Data at rest inside the volume
- All data moving between the volume and the instance
- All snapshots created from the volume
- All volumes created from those snapshots

EBS encrypts your volume with a data key using the industry-standard AES-256 algorithm. Your data key is stored on disk with your encrypted data, but not before EBS encrypts it with your KMS key. Your data key never appears on disk in plaintext. The same data key is shared by snapshots of the volume and any subsequent volumes created from those snapshots. For more information, see Data keys in the AWS Key Management Service Developer Guide.

Amazon EC2 works with AWS KMS to encrypt and decrypt your EBS volumes in slightly different ways depending on whether the snapshot from which you create an encrypted volume is encrypted or unencrypted.

How EBS encryption works when the snapshot is encrypted

When you create an encrypted volume from an encrypted snapshot that you own, Amazon EC2 works with AWS KMS to encrypt and decrypt your EBS volumes as follows:

1. Amazon EC2 sends a GenerateDataKeyWithoutPlaintext request to AWS KMS, specifying the KMS key that you chose for volume encryption.
2. AWS KMS generates a new data key, encrypts it under the KMS key that you chose for volume encryption, and sends the encrypted data key to Amazon EBS to be stored with the volume metadata.
3. When you attach the encrypted volume to an instance, Amazon EC2 sends a CreateGrant request to AWS KMS so that it can decrypt the data key.
4. AWS KMS decrypts the encrypted data key and sends the decrypted data key to Amazon EC2.
5. Amazon EC2 uses the plaintext data key in hypervisor memory to encrypt disk I/O to the volume. The plaintext data key persists in memory as long as the volume is attached to the instance.
How EBS encryption works when the snapshot is unencrypted

When you create an encrypted volume from unencrypted snapshot, Amazon EC2 works with AWS KMS to encrypt and decrypt your EBS volumes as follows:

1. Amazon EC2 sends a `CreateGrant` request to AWS KMS, so that it can encrypt the volume that is created from the snapshot.
2. Amazon EC2 sends a `GenerateDataKeyWithoutPlaintext` request to AWS KMS, specifying the KMS key that you chose for volume encryption.
3. AWS KMS generates a new data key, encrypts it under the KMS key that you chose for volume encryption, and sends the encrypted data key to Amazon EBS to be stored with the volume metadata.
4. Amazon EC2 sends a `Decrypt` request to AWS KMS to get the encryption key to encrypt the volume data.
5. When you attach the encrypted volume to an instance, Amazon EC2 sends a `CreateGrant` request to AWS KMS, so that it can decrypt the data key.
6. When you attach the encrypted volume to an instance, Amazon EC2 sends a `Decrypt` request to AWS KMS, specifying the encrypted data key.
7. AWS KMS decrypts the encrypted data key and sends the decrypted data key to Amazon EC2.
8. Amazon EC2 uses the plaintext data key in hypervisor memory to encrypt disk I/O to the volume. The plaintext data key persists in memory as long as the volume is attached to the instance.

For more information, see How Amazon Elastic Block Store (Amazon EBS) uses AWS KMS and Amazon EC2 example two in the AWS Key Management Service Developer Guide.

Requirements

Before you begin, verify that the following requirements are met.

Supported volume types

Encryption is supported by all EBS volume types. You can expect the same IOPS performance on encrypted volumes as on unencrypted volumes, with a minimal effect on latency. You can access encrypted volumes the same way that you access unencrypted volumes. Encryption and decryption are handled transparently, and they require no additional action from you or your applications.

Supported instance types

Amazon EBS encryption is available on all current generation (p. 142) instance types and the following previous generation (p. 145) instance types: A1, C3, cr1.8xlarge, G2, I2, M3, and R3.

Permissions for IAM users

When you configure a KMS key as the default key for EBS encryption, the default KMS key policy allows any IAM user with access to the required KMS actions to use this KMS key to encrypt or decrypt EBS resources. You must grant IAM users permission to call the following actions in order to use EBS encryption:

- `kms:CreateGrant`
- `kms:Decrypt`
- `kms:DescribeKey`
- `kms:GenerateDataKeyWithoutPlainText`
- `kms:ReEncrypt`
To follow the principle of least privilege, do not allow full access to `kms:CreateGrant`. Instead, allow the user to create grants on the KMS key only when the grant is created on the user's behalf by an AWS service, as shown in the following example.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kms:CreateGrant",
            "Resource": [
                "arn:aws:kms:us-east-2:123456789012:key/abcd1234-a123-456d-a12b-a123b4cd56ef"
            ],
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
        }
    ]
}
```

For more information, see Allows access to the AWS account and enables IAM policies in the Default key policy section in the AWS Key Management Service Developer Guide.

**Default KMS key for EBS encryption**

Amazon EBS automatically creates a unique AWS managed key in each Region where you store AWS resources. This KMS key has the alias `/alias/aws/ebs`. By default, Amazon EBS uses this KMS key for encryption. Alternatively, you can specify a symmetric customer managed key that you created as the default KMS key for EBS encryption. Using your own KMS key gives you more flexibility, including the ability to create, rotate, and disable KMS keys.

**Important**

Amazon EBS does not support asymmetric KMS keys. For more information, see Using symmetric and asymmetric KMS keys in the AWS Key Management Service Developer Guide.

**New console**

1. Open the Amazon EC2 console at `https://console.aws.amazon.com/ec2/`.
2. From the navigation bar, select the Region.
3. From the navigation pane, select **EC2 Dashboard**.
4. In the upper-right corner of the page, choose **Account Attributes, EBS encryption**.
5. Choose **Manage**.
6. For **Default encryption key**, choose a symmetric customer managed key.
7. Choose **Update EBS encryption**.

**Old console**

1. Open the Amazon EC2 console at `https://console.aws.amazon.com/ec2/`.
2. From the navigation bar, select the Region.
3. From the navigation pane, select EC2 Dashboard.
4. In the upper-right corner of the page, choose Account Attributes, Settings.
5. Choose Change the default key and then choose an available KMS key.
6. Choose Save settings.

Encryption by default

You can configure your AWS account to enforce the encryption of the new EBS volumes and snapshot copies that you create. For example, Amazon EBS encrypts the EBS volumes created when you launch an instance and the snapshots that you copy from an unencrypted snapshot. For examples of transitioning from unencrypted to encrypted EBS resources, see Encrypt unencrypted resources (p. 1332).

Encryption by default has no effect on existing EBS volumes or snapshots.

Considerations

- Encryption by default is a Region-specific setting. If you enable it for a Region, you cannot disable it for individual volumes or snapshots in that Region.
- When you enable encryption by default, you can launch an instance only if the instance type supports EBS encryption. For more information, see Supported instance types (p. 1329).
- If you copy a snapshot and encrypt it to a new KMS key, a complete (non-incremental) copy is created. This results in additional storage costs.
- When migrating servers using AWS Server Migration Service (SMS), do not turn on encryption by default. If encryption by default is already on and you are experiencing delta replication failures, turn off encryption by default. Instead, enable AMI encryption when you create the replication job.

New console

To enable encryption by default for a Region

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region.
3. From the navigation pane, select EC2 Dashboard.
4. In the upper-right corner of the page, choose Account Attributes, EBS encryption.
5. Choose Manage.
6. Select Enable. You keep the AWS managed key with the alias alias/aws/ebs created on your behalf as the default encryption key, or choose a symmetric customer managed key.
7. Choose Update EBS encryption.

Old console

To enable encryption by default for a Region

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region.
3. From the navigation pane, select EC2 Dashboard.
4. In the upper-right corner of the page, choose Account Attributes, Settings.
5. Under EBS Storage, select Always encrypt new EBS volumes.
6. Choose Save settings.
You cannot change the KMS key that is associated with an existing snapshot or encrypted volume. However, you can associate a different KMS key during a snapshot copy operation so that the resulting copied snapshot is encrypted by the new KMS key.

**Encrypt EBS resources**

You encrypt EBS volumes by enabling encryption, either using encryption by default (p. 1331) or by enabling encryption when you create a volume that you want to encrypt.

When you encrypt a volume, you can specify the symmetric KMS key to use to encrypt the volume. If you do not specify a KMS key, the KMS key that is used for encryption depends on the encryption state of the source snapshot and its ownership. For more information, see the encryption outcomes table (p. 1336).

**Note**

If you are using the API or AWS CLI to specify a KMS key, be aware that AWS authenticates the KMS key asynchronously. If you specify a KMS key ID, an alias, or an ARN that is not valid, the action can appear to complete, but it eventually fails.

You cannot change the KMS key that is associated with an existing snapshot or volume. However, you can associate a different KMS key during a snapshot copy operation so that the resulting copied snapshot is encrypted by the new KMS key.

**Encrypt an empty volume on creation**

When you create a new, empty EBS volume, you can encrypt it by enabling encryption for the specific volume creation operation. If you enabled EBS encryption by default, the volume is automatically encrypted using your default KMS key for EBS encryption. Alternatively, you can specify a different symmetric KMS key for the specific volume creation operation. The volume is encrypted by the time it is first available, so your data is always secured. For detailed procedures, see Create an Amazon EBS volume (p. 1183).

By default, the KMS key that you selected when creating a volume encrypts the snapshots that you make from the volume and the volumes that you restore from those encrypted snapshots. You cannot remove encryption from an encrypted volume or snapshot, which means that a volume restored from an encrypted snapshot, or a copy of an encrypted snapshot, is always encrypted.

Public snapshots of encrypted volumes are not supported, but you can share an encrypted snapshot with specific accounts. For detailed directions, see Share an Amazon EBS snapshot (p. 1234).

**Encrypt unencrypted resources**

Although there is no direct way to encrypt an existing unencrypted volume or snapshot, you can encrypt them by creating either a volume or a snapshot. If you enabled encryption by default, Amazon EBS encrypts the resulting new volume or snapshot using your default KMS key for EBS encryption. Even if you have not enabled encryption by default, you can enable encryption when you create an individual volume or snapshot. Whether you enable encryption by default or in individual creation operations, you can override the default KMS key for EBS encryption and select a symmetric customer managed key. For more information, see Create an Amazon EBS volume (p. 1183) and Copy an Amazon EBS snapshot (p. 1229).

To encrypt the snapshot copy to a customer managed key, you must both enable encryption and specify the KMS key, as shown in Copy an unencrypted snapshot (encryption by default not enabled) (p. 1334).

**Important**

Amazon EBS does not support asymmetric KMS keys. For more information, see Using Symmetric and Asymmetric KMS keys in the AWS Key Management Service Developer Guide.

You can also apply new encryption states when launching an instance from an EBS-backed AMI. This is because EBS-backed AMIs include snapshots of EBS volumes that can be encrypted as described. For more information, see Use encryption with EBS-backed AMIs (p. 127).
**Encryption scenarios**

When you create an encrypted EBS resource, it is encrypted by your account's default KMS key for EBS encryption unless you specify a different customer managed key in the volume creation parameters or the block device mapping for the AMI or instance. For more information, see Default KMS key for EBS encryption (p. 1330).

The following examples illustrate how you can manage the encryption state of your volumes and snapshots. For a full list of encryption cases, see the encryption outcomes table (p. 1336).

**Examples**
- Restore an unencrypted volume (encryption by default not enabled) (p. 1333)
- Restore an unencrypted volume (encryption by default enabled) (p. 1333)
- Copy an unencrypted snapshot (encryption by default not enabled) (p. 1334)
- Copy an unencrypted snapshot (encryption by default enabled) (p. 1335)
- Re-encrypt an encrypted volume (p. 1335)
- Re-encrypt an encrypted snapshot (p. 1336)
- Migrate data between encrypted and unencrypted volumes (p. 1336)
- Encryption outcomes (p. 1336)

**Restore an unencrypted volume (encryption by default not enabled)**

Without encryption by default enabled, a volume restored from an unencrypted snapshot is unencrypted by default. However, you can encrypt the resulting volume by setting the Encrypted parameter and, optionally, the KmsKeyId parameter. The following diagram illustrates the process.

![Diagram](image)

If you leave out the KmsKeyId parameter, the resulting volume is encrypted using your default KMS key for EBS encryption. You must specify a KMS key ID to encrypt the volume to a different KMS key.

For more information, see Create a volume from a snapshot (p. 1185).

**Restore an unencrypted volume (encryption by default enabled)**

When you have enabled encryption by default, encryption is mandatory for volumes restored from unencrypted snapshots, and no encryption parameters are required for your default KMS key to be used. The following diagram shows this simple default case:
If you want to encrypt the restored volume to a symmetric customer managed key, you must supply both the `Encrypted` and `KmsKeyId` parameters as shown in Restore an unencrypted volume (encryption by default not enabled) (p. 1333).

**Copy an unencrypted snapshot (encryption by default not enabled)**

Without encryption by default enabled, a copy of an unencrypted snapshot is unencrypted by default. However, you can encrypt the resulting snapshot by setting the `Encrypted` parameter and, optionally, the `KmsKeyId` parameter. If you omit `KmsKeyId`, the resulting snapshot is encrypted by your default KMS key. You must specify a KMS key ID to encrypt the volume to a different symmetric KMS key.

The following diagram illustrates the process.

You can encrypt an EBS volume by copying an unencrypted snapshot to an encrypted snapshot and then creating a volume from the encrypted snapshot. For more information, see Copy an Amazon EBS snapshot (p. 1229).
Copy an unencrypted snapshot (encryption by default enabled)

When you have enabled encryption by default, encryption is mandatory for copies of unencrypted snapshots, and no encryption parameters are required if your default KMS key is used. The following diagram illustrates this default case:

Re-encrypt an encrypted volume

When the CreateVolume action operates on an encrypted snapshot, you have the option of re-encrypting it with a different KMS key. The following diagram illustrates the process. In this example, you own two KMS keys, KMS key A and KMS key B. The source snapshot is encrypted by KMS key A. During volume creation, with the KMS key ID of KMS key B specified as a parameter, the source data is automatically decrypted, then re-encrypted by KMS key B.

For more information, see Create a volume from a snapshot (p. 1185).
Re-encrypt an encrypted snapshot

The ability to encrypt a snapshot during copying allows you to apply a new symmetric KMS key to an already-encrypted snapshot that you own. Volumes restored from the resulting copy are only accessible using the new KMS key. The following diagram illustrates the process. In this example, you own two KMS keys, KMS key A and KMS key B. The source snapshot is encrypted by KMS key A. During copy, with the KMS key ID of KMS key B specified as a parameter, the source data is automatically re-encrypted by KMS key B.

In a related scenario, you can choose to apply new encryption parameters to a copy of a snapshot that has been shared with you. By default, the copy is encrypted with a KMS key shared by the snapshot's owner. However, we recommend that you create a copy of the shared snapshot using a different KMS key that you control. This protects your access to the volume if the original KMS key is compromised, or if the owner revokes the KMS key for any reason. For more information, see Encryption and snapshot copying (p. 1231).

Migrate data between encrypted and unencrypted volumes

When you have access to both an encrypted and unencrypted volume, you can freely transfer data between them. EC2 carries out the encryption and decryption operations transparently.

For example, use the robocopy command to copy the data. In the following command, the source data is located in D: \ and the destination volume is mounted at E: \.

```bash
PS C:\> robocopy D:\sourcefolder E:\destinationfolder /e /copyall /eta
```

We recommend using folders rather than copying an entire volume, as this avoids potential problems with hidden folders.

Encryption outcomes

The following table describes the encryption outcome for each possible combination of settings.
<table>
<thead>
<tr>
<th>Is encryption enabled?</th>
<th>Is encryption by default enabled?</th>
<th>Source of volume</th>
<th>Default (no customer managed key specified)</th>
<th>Custom (customer managed key specified)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>New (empty) volume</td>
<td>Unencrypted</td>
<td>N/A</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Unencrypted snapshot that you own</td>
<td>Unencrypted</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Encrypted snapshot that you own</td>
<td>Encrypted by same key</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Unencrypted snapshot that is shared with you</td>
<td>Unencrypted</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Encrypted snapshot that is shared with you</td>
<td>Encrypted by default customer managed key*</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>New volume</td>
<td>Encrypted by default customer managed key</td>
<td>Encrypted by a specified customer managed key**</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Unencrypted snapshot that you own</td>
<td>Encrypted by default customer managed key</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Encrypted snapshot that you own</td>
<td>Encrypted by same key</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Unencrypted snapshot that is shared with you</td>
<td>Encrypted by default customer managed key</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Encrypted snapshot that is shared with you</td>
<td>Encrypted by default customer managed key</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>New (empty) volume</td>
<td>Encrypted by default customer managed key</td>
<td>N/A</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Unencrypted snapshot that you own</td>
<td>Encrypted by default customer managed key</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Encrypted snapshot that you own</td>
<td>Encrypted by same key</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Unencrypted snapshot that is shared with you</td>
<td>Encrypted by default customer managed key</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Encrypted snapshot that is shared with you</td>
<td>Encrypted by default customer managed key</td>
<td></td>
</tr>
</tbody>
</table>
EBS data services

<table>
<thead>
<tr>
<th>Is encryption enabled?</th>
<th>Is encryption by default enabled?</th>
<th>Source of volume</th>
<th>Default (no customer managed key specified)</th>
<th>Custom (customer managed key specified)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>New volume</td>
<td>Encrypted by default customer managed key</td>
<td>Encrypted by a specified customer managed key</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Unencrypted snapshot that you own</td>
<td>Encrypted by default customer managed key</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Encrypted snapshot that you own</td>
<td>Encrypted by same key</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Unencrypted snapshot that is shared with you</td>
<td>Encrypted by default customer managed key</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Encrypted snapshot that is shared with you</td>
<td>Encrypted by default customer managed key</td>
<td></td>
</tr>
</tbody>
</table>

* This is the default customer managed key used for EBS encryption for the AWS account and Region. By default this is a unique AWS managed key for EBS, or you can specify a customer managed key. For more information, see Default KMS key for EBS encryption (p. 1330).

** This is a customer managed key specified for the volume at launch time. This customer managed key is used instead of the default customer managed key for the AWS account and Region.

Set encryption defaults using the API and CLI

You can manage encryption by default and the default KMS key using the following API actions and CLI commands.

<table>
<thead>
<tr>
<th>API action</th>
<th>CLI command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisableEbsEncryptionByDefault</td>
<td>disable-ebs-encryption-by-default</td>
<td>Disables encryption by default.</td>
</tr>
<tr>
<td>EnableEbsEncryptionByDefault</td>
<td>enable-ebs-encryption-by-default</td>
<td>Enables encryption by default.</td>
</tr>
<tr>
<td>GetEbsDefaultKmsKeyId</td>
<td>get-eb-default-kms-key-id</td>
<td>Describes the default KMS key.</td>
</tr>
<tr>
<td>GetEbsEncryptionByDefault</td>
<td>get-ebs-encryption-by-default</td>
<td>Indicates whether encryption by default is enabled.</td>
</tr>
<tr>
<td>ModifyEbsDefaultKmsKeyId</td>
<td>modify-ebs-default-kms-key-id</td>
<td>Changes the default KMS key used to encrypt EBS volumes.</td>
</tr>
<tr>
<td>ResetEbsDefaultKmsKeyId</td>
<td>reset-ebs-default-kms-key-id</td>
<td>Resets the AWS managed key as the default KMS key used to encrypt EBS volumes.</td>
</tr>
</tbody>
</table>
Amazon EBS fast snapshot restore

Amazon EBS fast snapshot restore enables you to create a volume from a snapshot that is fully initialized at creation. This eliminates the latency of I/O operations on a block when it is accessed for the first time. Volumes that are created using fast snapshot restore instantly deliver all of their provisioned performance.

To get started, enable fast snapshot restore for specific snapshots in specific Availability Zones. Each snapshot and Availability Zone pair refers to one fast snapshot restore. When you create a volume from one of these snapshots in one of its enabled Availability Zones, the volume is restored using fast snapshot restore.

Fast snapshot restore must be explicitly enabled on a per-snapshot basis. If you create a new snapshot from a volume that was restored from a fast snapshot restore-enabled snapshot, the new snapshot is not automatically enabled for fast snapshot restore. You must explicitly enable it for the new snapshot.

You can enable fast snapshot restore for snapshots that you own and for public and private snapshots that are shared with you.

Contents

- Fast snapshot restore quotas (p. 1339)
- Fast snapshot restore states (p. 1339)
- Volume creation credits (p. 1339)
- Manage fast snapshot restore (p. 1340)
- View snapshots with fast snapshot restore enabled (p. 1341)
- View volumes restored using fast snapshot restore (p. 1342)
- Monitor fast snapshot restore (p. 1342)
- Pricing and Billing (p. 1342)

Fast snapshot restore quotas

You can enable up to 50 snapshots for fast snapshot restore per Region. The quota applies to snapshots that you own and snapshots that are shared with you. If you enable fast snapshot restore for a snapshot that is shared with you, it counts towards your fast snapshot restore quota. It does not count towards the snapshot owner's fast snapshot restore quota.

Fast snapshot restore states

After you enable fast snapshot restore for a snapshot, it can be in one of the following states.

- enabling — A request was made to enable fast snapshot restore.
- optimizing — Fast snapshot restore is being enabled. It takes 60 minutes per TiB to optimize a snapshot. Snapshots in this state offer some performance benefit when restoring volumes.
- enabled — Fast snapshot restore is enabled. Snapshots in this state offer the full performance benefit when restoring volumes.
- disabling — A request was made to disable fast snapshot restore, or a request to enable fast snapshot restore failed.
- disabled — Fast snapshot restore is disabled. You can enable fast snapshot restore again as needed.

Volume creation credits

The number of volumes that receive the full performance benefit of fast snapshot restore is determined by the volume creation credits for the snapshot. There is one credit bucket per snapshot per Availability Zone. Each volume that you create from a snapshot with fast snapshot restore enabled consumes one
credit from the credit bucket. If you create a volume but there is less than one credit in the bucket, the volume is created without benefit of fast snapshot restore.

When you enable fast snapshot restore for a snapshot that is shared with you, you get a separate credit bucket for the shared snapshot in your account. If you create volumes from the shared snapshot, the credits are consumed from your credit bucket; they are not consumed from the snapshot owner’s credit bucket.

The size of a credit bucket depends on the size of the snapshot, not the size of the volumes created from the snapshot. The size of the credit bucket for each snapshot is calculated as follows:

\[
\text{MAX (1, MIN (10, FLOOR}(1024/\text{snapshot\_size\_gib}))\]

As you consume credits, the credit bucket is refilled over time. The refill rate for each credit bucket is calculated as follows:

\[
\text{MIN (10, 1024/\text{snapshot\_size\_gib})}
\]

For example, if you enable fast snapshot restore for a snapshot with a size of 100 GiB, the maximum size of its credit bucket is 10 credits and the refill rate is 10 credits per hour. When the credit bucket is full, you can create 10 initialized volumes from this snapshot simultaneously.

You can use Cloudwatch metrics to monitor the size of your credit buckets and the number of credits available in each bucket. For more information, see Fast snapshot restore metrics (p. 1380).

After you create a volume from a snapshot with fast snapshot restore enabled, you can describe the volume using describe-volumes and check the fastRestored field in the output to determine whether the volume was created as an initialized volume using fast snapshot restore.

**Manage fast snapshot restore**

Fast snapshot restore is disabled for a snapshot by default. You can enable or disable fast snapshot restore for snapshots that you own and for snapshots that are shared with you. When you enable or disable fast snapshot restore for a snapshot, the changes apply to your account only.

**Note**

When you enable fast snapshot restore for a snapshot, your account is billed for each minute that fast snapshot restore is enabled in a particular Availability Zone. Charges are pro-rated and have a minimum of one hour.

When you delete a snapshot that you own, fast snapshot restore is automatically disabled for that snapshot in your account. If you enabled fast snapshot restore for a snapshot that is shared with you, and the snapshot owner deletes or unshares it, fast snapshot restore is automatically disabled for the shared snapshot in your account.

If you enabled fast snapshot restore for a snapshot that is shared with you, and it’s encrypted using a custom CMK, fast snapshot restore is not automatically disabled for the snapshot when the snapshot owner revokes your access to the custom CMK. You must manually disable fast snapshot restore for that snapshot.

Use the following procedure to enable or disable fast snapshot restore for a snapshot that you own or for a snapshot that is shared with you.

**To enable or disable fast snapshot restore**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Snapshots**.
3. Select the snapshot.
4. Choose **Actions, Manage Fast Snapshot Restore**.
5. Select or deselect Availability Zones, and then choose **Save**.
6. To track the state of fast snapshot restore as it is enabled, see **Fast Snapshot Restore** on the **Description** tab.

**Note**
After you enable fast snapshot restore for a snapshot, it enters the **optimizing** state. Snapshots that are in the **optimizing** state provide some performance benefits when using them to restore volumes. They start to provide the full performance benefits of fast snapshot restore only after they enter the **enabled** state.

**To manage fast snapshot restore using the AWS CLI**

- `enable-fast-snapshot-restores`
- `disable-fast-snapshot-restores`
- `describe-fast-snapshot-restores`

**View snapshots with fast snapshot restore enabled**

Use the following procedure to view the state of fast snapshot restore for a snapshot that you own or for a snapshot that is shared with you.

**To view the state of fast snapshot restore using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Snapshots**.
3. Select the snapshot.
4. On the **Description** tab, see **Fast Snapshot Restore**, which indicates the state of fast snapshot restore. For example, it might show a state of "2 Availability Zones optimizing" or "2 Availability Zones enabled".

**To view snapshots with fast snapshot restore enabled using the AWS CLI**

Use the `describe-fast-snapshot-restores` command to describe the snapshots that are enabled for fast snapshot restore.

```
aws ec2 describe-fast-snapshot-restores --filters Name=state,Values=enabled
```

The following is example output.

```
{
    "FastSnapshotRestores": [
        {
            "SnapshotId": "snap-0e94653493cb0447",
            "AvailabilityZone": "us-east-2a",
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated - Lifecycle state transition",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.596Z",
            "OptimizingTime": "2020-01-25T23:58:25.573Z",
            "EnabledTime": "2020-01-25T23:59:29.852Z"
        },
        {
            "SnapshotId": "snap-0e94653493cb0447",
            "AvailabilityZone": "us-east-2b",
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated - Lifecycle state transition",
```
View volumes restored using fast snapshot restore

When you create a volume from a snapshot that is enabled for fast snapshot restore in the Availability Zone for the volume, it is restored using fast snapshot restore.

Use the `describe-volumes` command to view volumes that were created from a snapshot that is enabled for fast snapshot restore.

```
aws ec2 describe-volumes --filters Name=fast-restored,Values=true
```

The following is example output.

```
{
  "Volumes": [
    {
      "Attachments": [],
      "AvailabilityZone": "us-east-2a",
      "CreateTime": "2020-01-26T00:34:11.093Z",
      "Encrypted": true,
      "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/8c5b2c63-b9bc-45a3-a87a-5513e232e843",
      "Size": 20,
      "SnapshotId": "snap-0e946653493cb0447",
      "State": "available",
      "VolumeId": "vol-0d371921d4ca797b0",
      "Iops": 100,
      "VolumeType": "gp2",
      "FastRestored": true
    }
  ]
}
```

Monitor fast snapshot restore

Amazon EBS emits Amazon CloudWatch events when the fast snapshot restore state for a snapshot changes. For more information, see EBS fast snapshot restore events (p. 1389).

Pricing and Billing

You are billed for each minute that fast snapshot restore is enabled for a snapshot in a particular Availability Zone. Charges are pro-rated with a minimum of one hour.

For example, if you enable fast snapshot restore for one snapshot in US-East-1a for one month (30 days), you are billed $540 (1 snapshot x 1 AZ x 720 hours x $0.75 per hour). If you enable fast snapshot restore for two snapshots in us-east-1a, us-east-1b, and us-east-1c for the same period, you are billed $3240 (2 snapshots x 3 AZs x 720 hours x $0.75 per hour).

If you enable fast snapshot restore for a public or private snapshot that is shared with you, your account is billed; the snapshot owner is not billed. When a snapshot that is shared with you is deleted or unshared by the snapshot owner, fast snapshot restore is disabled for the snapshot in your account and billing is stopped.

For more information, see Amazon EBS pricing.
Amazon EBS and NVMe on Windows instances

EBS volumes are exposed as NVMe block devices on instances built on the Nitro System (p. 146). When you attach a volume to your instance, you include a device name for the volume. This device name is used by Amazon EC2. The block device driver for the instance assigns the actual volume name when mounting the volume, and the name assigned can be different from the name that Amazon EC2 uses.

The EBS performance guarantees stated in Amazon EBS Product Details are valid regardless of the block-device interface.

Contents

- Install or upgrade the NVMe driver (p. 1343)
- Identify the EBS device (p. 1343)
- Work with NVMe EBS volumes (p. 1344)
- I/O operation timeout (p. 1344)

Install or upgrade the NVMe driver

The AWS Windows AMIs for Windows Server 2008 R2 and later include the AWS NVMe driver. If you are not using the latest AWS Windows AMIs provided by Amazon, see Install or upgrade AWS NVMe drivers (p. 538).

Identify 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. Additionally, the device name assigned by the block device driver can be different from the name specified in the block device mapping.

The following example shows the command and output for a volume attached during instance launch. Note that the NVMe device name does not include the /dev/ prefix.

The following example shows the command and output for a volume attached after instance launch. Note that the NVMe device name includes the /dev/ prefix.

Windows Server 2008 R2 and later

You can also run the ebsnvme-id command to map the NVMe device disk number to an EBS volume ID and device name. By default, all EBS NVMe devices are enumerated. You can pass a disk number to enumerate information for a specific device. Ebsnvme-id is included in the latest AWS provided Windows Server AMIs located in C:\PROGRAMDATA\AMAZON\Tools.

You can also download ebsnvme-id.zip and extract the contents to your Amazon EC2 instance to get access to ebsnvme-id.exe.

```
PS C:\Users\Administrator\Desktop> ebsnvme-id.exe
Disk Number: 0
Volume ID: vol-0d6d7ee9f6e471a7f
Device Name: sda1

Disk Number: 1
Volume ID: vol-03a26248ff39b57cf
Device Name: xvdd

Disk Number: 2
```
Volume ID: vol-038bd1c629aa125e6
Device Name: xvde

Disk Number: 3
Volume ID: vol-034f9d29ec0b64c89
Device Name: xvdb

Disk Number: 4
Volume ID: vol-03e2dbe464b66f0a1
Device Name: xvdc

Work 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. 538).

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 EBS–optimized instances

An Amazon EBS–optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

EBS–optimized instances deliver dedicated bandwidth to Amazon EBS. When attached to an EBS–optimized instance, General Purpose SSD (gp2 and gp3) volumes are designed to deliver their baseline and burst performance 99% of the time, and Provisioned IOPS SSD (io1 and io2) volumes are designed to deliver their provisioned performance 99.9% of the time. Both Throughput Optimized HDD (st1) and Cold HDD (sc1) guarantee performance consistency of 90% of burst throughput 99% of the time. Non-compliant periods are approximately uniformly distributed, targeting 99% of expected total throughput each hour. For more information, see Amazon EBS volume types (p. 1163).

Contents
- Supported instance types (p. 1344)
- Get maximum performance (p. 1358)
- View instances types that support EBS optimization (p. 1359)
- Enable EBS optimization at launch (p. 1360)
- Enable EBS optimization for an existing instance (p. 1360)

Supported instance types

The following tables show which instance types support EBS optimization. They include the dedicated bandwidth to Amazon EBS, the typical maximum aggregate throughput that can be achieved on that
connection with a streaming read workload and 128 KiB I/O size, and the maximum IOPS the instance can support if you are using a 16 KiB I/O size. Choose an EBS–optimized instance that provides more dedicated Amazon EBS throughput than your application needs; otherwise, the connection between Amazon EBS and Amazon EC2 can become a performance bottleneck.

**EBS optimized by default**

The following table lists the instance types that support EBS optimization and EBS optimization is enabled by default. There is no need to enable EBS optimization and no effect if you disable EBS optimization.

*Note*
You can also view this information programatically using the AWS CLI. For more information, see View instances types that support EBS optimization (p. 1359).

<table>
<thead>
<tr>
<th>Instance size</th>
<th>Maximum bandwidth (Mbps)</th>
<th>Maximum throughput (MB/s, 128 KiB I/O)</th>
<th>Maximum IOPS (16 KiB I/O)</th>
</tr>
</thead>
<tbody>
<tr>
<td>c4.large</td>
<td>500</td>
<td>62.5</td>
<td>4,000</td>
</tr>
<tr>
<td>c4.xlarge</td>
<td>750</td>
<td>93.75</td>
<td>6,000</td>
</tr>
<tr>
<td>c4.2xlarge</td>
<td>1,000</td>
<td>125</td>
<td>8,000</td>
</tr>
<tr>
<td>c4.4xlarge</td>
<td>2,000</td>
<td>250</td>
<td>16,000</td>
</tr>
<tr>
<td>c4.8xlarge</td>
<td>4,000</td>
<td>500</td>
<td>32,000</td>
</tr>
<tr>
<td>c5.large *</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>c5.xlarge *</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>c5.2xlarge *</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>c5.4xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>c5.9xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>c5.12xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>c5.18xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>c5.24xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>c5.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>c5a.large *</td>
<td>3,170</td>
<td>396</td>
<td>13,300</td>
</tr>
<tr>
<td>c5a.xlarge *</td>
<td>3,170</td>
<td>396</td>
<td>13,300</td>
</tr>
<tr>
<td>c5a.2xlarge *</td>
<td>3,170</td>
<td>396</td>
<td>13,300</td>
</tr>
<tr>
<td>c5a.4xlarge *</td>
<td>3,170</td>
<td>396</td>
<td>13,300</td>
</tr>
<tr>
<td>c5a.8xlarge</td>
<td>3,170</td>
<td>396</td>
<td>13,300</td>
</tr>
<tr>
<td>c5a.12xlarge</td>
<td>4,750</td>
<td>594</td>
<td>20,000</td>
</tr>
<tr>
<td>c5a.16xlarge</td>
<td>6,300</td>
<td>788</td>
<td>26,700</td>
</tr>
<tr>
<td>c5a.24xlarge</td>
<td>9,500</td>
<td>1,188</td>
<td>40,000</td>
</tr>
<tr>
<td>Instance size</td>
<td>Maximum bandwidth (Mbps)</td>
<td>Maximum throughput (MB/s, 128 KiB I/O)</td>
<td>Maximum IOPS (16 KiB I/O)</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------</td>
<td>---------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>c5ad.large *</td>
<td>3,170</td>
<td>396</td>
<td>13,300</td>
</tr>
<tr>
<td>c5ad.xlarge *</td>
<td>3,170</td>
<td>396</td>
<td>13,300</td>
</tr>
<tr>
<td>c5ad.2xlarge *</td>
<td>3,170</td>
<td>396</td>
<td>13,300</td>
</tr>
<tr>
<td>c5ad.4xlarge *</td>
<td>3,170</td>
<td>396</td>
<td>13,300</td>
</tr>
<tr>
<td>c5ad.8xlarge</td>
<td>3,170</td>
<td>396</td>
<td>13,300</td>
</tr>
<tr>
<td>c5ad.12xlarge</td>
<td>4,750</td>
<td>594</td>
<td>20,000</td>
</tr>
<tr>
<td>c5ad.16xlarge</td>
<td>6,300</td>
<td>788</td>
<td>26,700</td>
</tr>
<tr>
<td>c5ad.24xlarge</td>
<td>9,500</td>
<td>1,188</td>
<td>40,000</td>
</tr>
<tr>
<td>c5d.large *</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>c5d.xlarge *</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>c5d.2xlarge *</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>c5d.4xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>c5d.9xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>c5d.12xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>c5d.18xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>c5d.24xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>c5d.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>c5n.large *</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>c5n.xlarge *</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>c5n.2xlarge *</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>c5n.4xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>c5n.9xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>c5n.18xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>c5n.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>d2.xlarge</td>
<td>750</td>
<td>93.75</td>
<td>6,000</td>
</tr>
<tr>
<td>d2.2xlarge</td>
<td>1,000</td>
<td>125</td>
<td>8,000</td>
</tr>
<tr>
<td>d2.4xlarge</td>
<td>2,000</td>
<td>250</td>
<td>16,000</td>
</tr>
<tr>
<td>d2.8xlarge</td>
<td>4,000</td>
<td>500</td>
<td>32,000</td>
</tr>
<tr>
<td>d3.xlarge *</td>
<td>2,800</td>
<td>350</td>
<td>15,000</td>
</tr>
<tr>
<td>d3.2xlarge *</td>
<td>2,800</td>
<td>350</td>
<td>15,000</td>
</tr>
<tr>
<td>Instance size</td>
<td>Maximum bandwidth (Mbps)</td>
<td>Maximum throughput (MB/s, 128 KiB I/O)</td>
<td>Maximum IOPS (16 KiB I/O)</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------</td>
<td>----------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>d3.4xlarge</td>
<td>2,800</td>
<td>350</td>
<td>15,000</td>
</tr>
<tr>
<td>d3.8xlarge</td>
<td>5,000</td>
<td>625</td>
<td>30,000</td>
</tr>
<tr>
<td>d3en.xlarge *</td>
<td>2,800</td>
<td>350</td>
<td>15,000</td>
</tr>
<tr>
<td>d3en.2xlarge *</td>
<td>2,800</td>
<td>350</td>
<td>15,000</td>
</tr>
<tr>
<td>d3en.4xlarge</td>
<td>2,800</td>
<td>350</td>
<td>15,000</td>
</tr>
<tr>
<td>d3en.8xlarge</td>
<td>5,000</td>
<td>625</td>
<td>30,000</td>
</tr>
<tr>
<td>d3en.12xlarge</td>
<td>7,000</td>
<td>875</td>
<td>40,000</td>
</tr>
<tr>
<td>f1.2xlarge</td>
<td>1,700</td>
<td>212.5</td>
<td>12,000</td>
</tr>
<tr>
<td>f1.4xlarge</td>
<td>3,500</td>
<td>437.5</td>
<td>44,000</td>
</tr>
<tr>
<td>f1.16xlarge</td>
<td>14,000</td>
<td>1,750</td>
<td>75,000</td>
</tr>
<tr>
<td>g3s.xlarge</td>
<td>850</td>
<td>106.25</td>
<td>5,000</td>
</tr>
<tr>
<td>g3.4xlarge</td>
<td>3,500</td>
<td>437.5</td>
<td>20,000</td>
</tr>
<tr>
<td>g3.8xlarge</td>
<td>7,000</td>
<td>875</td>
<td>40,000</td>
</tr>
<tr>
<td>g3.16xlarge</td>
<td>14,000</td>
<td>1,750</td>
<td>80,000</td>
</tr>
<tr>
<td>g4ad.xlarge *</td>
<td>3,170</td>
<td>396.25</td>
<td>13,333</td>
</tr>
<tr>
<td>g4ad.2xlarge *</td>
<td>3,170</td>
<td>396.25</td>
<td>13,333</td>
</tr>
<tr>
<td>g4ad.4xlarge *</td>
<td>3,170</td>
<td>396.25</td>
<td>13,333</td>
</tr>
<tr>
<td>g4ad.8xlarge</td>
<td>3,170</td>
<td>396.25</td>
<td>13,333</td>
</tr>
<tr>
<td>g4ad.16xlarge</td>
<td>6,300</td>
<td>787.5</td>
<td>26,667</td>
</tr>
<tr>
<td>g4dn.xlarge *</td>
<td>3,500</td>
<td>437.5</td>
<td>20,000</td>
</tr>
<tr>
<td>g4dn.2xlarge *</td>
<td>3,500</td>
<td>437.5</td>
<td>20,000</td>
</tr>
<tr>
<td>g4dn.4xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>g4dn.8xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>g4dn.12xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>g4dn.16xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>g4dn.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>h1.2xlarge</td>
<td>1,750</td>
<td>218.75</td>
<td>12,000</td>
</tr>
<tr>
<td>h1.4xlarge</td>
<td>3,500</td>
<td>437.5</td>
<td>20,000</td>
</tr>
<tr>
<td>h1.8xlarge</td>
<td>7,000</td>
<td>875</td>
<td>40,000</td>
</tr>
<tr>
<td>h1.16xlarge</td>
<td>14,000</td>
<td>1,750</td>
<td>80,000</td>
</tr>
<tr>
<td>Instance size</td>
<td>Maximum bandwidth (Mbps)</td>
<td>Maximum throughput (MB/s, 128 KiB I/O)</td>
<td>Maximum IOPS (16 KiB I/O)</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------</td>
<td>--------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>i3.large</td>
<td>425</td>
<td>53.13</td>
<td>3000</td>
</tr>
<tr>
<td>i3.xlarge</td>
<td>850</td>
<td>106.25</td>
<td>6000</td>
</tr>
<tr>
<td>i3.2xlarge</td>
<td>1,700</td>
<td>212.5</td>
<td>12,000</td>
</tr>
<tr>
<td>i3.4xlarge</td>
<td>3,500</td>
<td>437.5</td>
<td>16,000</td>
</tr>
<tr>
<td>i3.8xlarge</td>
<td>7,000</td>
<td>875</td>
<td>32,500</td>
</tr>
<tr>
<td>i3.16xlarge</td>
<td>14,000</td>
<td>1,750</td>
<td>65,000</td>
</tr>
<tr>
<td>i3.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>i3en.large*</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>i3en.xlarge*</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>i3en.2xlarge*</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>i3en.3xlarge*</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>i3en.6xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>i3en.12xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>i3en.24xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>i3en.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>m4.large</td>
<td>450</td>
<td>56.25</td>
<td>3,600</td>
</tr>
<tr>
<td>m4.xlarge</td>
<td>750</td>
<td>93.75</td>
<td>6,000</td>
</tr>
<tr>
<td>m4.2xlarge</td>
<td>1,000</td>
<td>125</td>
<td>8,000</td>
</tr>
<tr>
<td>m4.4xlarge</td>
<td>2,000</td>
<td>250</td>
<td>16,000</td>
</tr>
<tr>
<td>m4.10xlarge</td>
<td>8,000</td>
<td>500</td>
<td>32,000</td>
</tr>
<tr>
<td>m4.16xlarge</td>
<td>10,000</td>
<td>1,250</td>
<td>65,000</td>
</tr>
<tr>
<td>m5.large*</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5.xlarge*</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5.2xlarge*</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5.4xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5.8xlarge</td>
<td>6,800</td>
<td>850</td>
<td>30,000</td>
</tr>
<tr>
<td>m5.12xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>m5.16xlarge</td>
<td>13,600</td>
<td>1,700</td>
<td>60,000</td>
</tr>
<tr>
<td>m5.24xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>m5.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
</tbody>
</table>
### EBS Optimization

<table>
<thead>
<tr>
<th>Instance size</th>
<th>Maximum bandwidth (Mbps)</th>
<th>Maximum throughput (MB/s, 128 KiB I/O)</th>
<th>Maximum IOPS (16 KiB I/O)</th>
</tr>
</thead>
<tbody>
<tr>
<td>m5a.large *</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>m5a.xlarge *</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>m5a.2xlarge *</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>m5a.4xlarge</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>m5a.8xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>m5a.12xlarge</td>
<td>6,780</td>
<td>847.5</td>
<td>30,000</td>
</tr>
<tr>
<td>m5a.16xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>m5a.24xlarge</td>
<td>13,570</td>
<td>1,696.25</td>
<td>60,000</td>
</tr>
<tr>
<td>m5ad.large *</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>m5ad.xlarge *</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>m5ad.2xlarge *</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>m5ad.4xlarge</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>m5ad.8xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>m5ad.12xlarge</td>
<td>6,780</td>
<td>847.5</td>
<td>30,000</td>
</tr>
<tr>
<td>m5ad.16xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>m5ad.24xlarge</td>
<td>13,570</td>
<td>1,696.25</td>
<td>60,000</td>
</tr>
<tr>
<td>m5d.large *</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5d.xlarge *</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5d.2xlarge *</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5d.4xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5d.8xlarge</td>
<td>6,800</td>
<td>850</td>
<td>30,000</td>
</tr>
<tr>
<td>m5d.12xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>m5d.16xlarge</td>
<td>13,600</td>
<td>1,700</td>
<td>60,000</td>
</tr>
<tr>
<td>m5d.24xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>m5d.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>m5dn.large *</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5dn.xlarge *</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5dn.2xlarge *</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5dn.4xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5dn.8xlarge</td>
<td>6,800</td>
<td>850</td>
<td>30,000</td>
</tr>
<tr>
<td>Instance size</td>
<td>Maximum bandwidth (Mbps)</td>
<td>Maximum throughput (MB/s, 128 KiB I/O)</td>
<td>Maximum IOPS (16 KiB I/O)</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------</td>
<td>---------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>m5dn.12xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>m5dn.16xlarge</td>
<td>13,600</td>
<td>1,700</td>
<td>60,000</td>
</tr>
<tr>
<td>m5dn.24xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>m5dn.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>m5n.large*</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5n.xlarge*</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5n.2xlarge*</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5n.4xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>m5n.8xlarge</td>
<td>6,800</td>
<td>850</td>
<td>30,000</td>
</tr>
<tr>
<td>m5n.12xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>m5n.16xlarge</td>
<td>13,600</td>
<td>1,700</td>
<td>60,000</td>
</tr>
<tr>
<td>m5n.24xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>m5n.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>m5zn.large*</td>
<td>3,170</td>
<td>396.25</td>
<td>13,333</td>
</tr>
<tr>
<td>m5zn.xlarge*</td>
<td>3,170</td>
<td>396.25</td>
<td>13,333</td>
</tr>
<tr>
<td>m5zn.2xlarge</td>
<td>3,170</td>
<td>396.25</td>
<td>13,333</td>
</tr>
<tr>
<td>m5zn.3xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>m5zn.6xlarge</td>
<td>9,500</td>
<td>1187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>m5zn.12xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>m5zn.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>m6i.large*</td>
<td>10,000</td>
<td>1,250</td>
<td>40,000</td>
</tr>
<tr>
<td>m6i.xlarge*</td>
<td>10,000</td>
<td>1,250</td>
<td>40,000</td>
</tr>
<tr>
<td>m6i.2xlarge*</td>
<td>10,000</td>
<td>1,250</td>
<td>40,000</td>
</tr>
<tr>
<td>m6i.4xlarge*</td>
<td>10,000</td>
<td>1,250</td>
<td>40,000</td>
</tr>
<tr>
<td>m6i.8xlarge</td>
<td>10,000</td>
<td>1,250</td>
<td>40,000</td>
</tr>
<tr>
<td>m6i.12xlarge</td>
<td>15,000</td>
<td>1,875</td>
<td>60,000</td>
</tr>
<tr>
<td>m6i.16xlarge</td>
<td>20,000</td>
<td>2,500</td>
<td>80,000</td>
</tr>
<tr>
<td>m6i.24xlarge</td>
<td>30,000</td>
<td>3,750</td>
<td>120,000</td>
</tr>
<tr>
<td>m6i.32xlarge</td>
<td>40,000</td>
<td>5,000</td>
<td>160,000</td>
</tr>
<tr>
<td>p2.xlarge</td>
<td>750</td>
<td>93.75</td>
<td>6,000</td>
</tr>
<tr>
<td>Instance size</td>
<td>Maximum bandwidth (Mbps)</td>
<td>Maximum throughput (MB/s, 128 KiB I/O)</td>
<td>Maximum IOPS (16 KiB I/O)</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------</td>
<td>---------------------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>p2.8xlarge</td>
<td>5,000</td>
<td>625</td>
<td>32,500</td>
</tr>
<tr>
<td>p2.16xlarge</td>
<td>10,000</td>
<td>1,250</td>
<td>65,000</td>
</tr>
<tr>
<td>p3.2xlarge</td>
<td>1,750</td>
<td>218.75</td>
<td>10,000</td>
</tr>
<tr>
<td>p3.8xlarge</td>
<td>7,000</td>
<td>875</td>
<td>40,000</td>
</tr>
<tr>
<td>p3.16xlarge</td>
<td>14,000</td>
<td>1,750</td>
<td>80,000</td>
</tr>
<tr>
<td>p3dn.24xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>r4.large</td>
<td>425</td>
<td>53.13</td>
<td>3,000</td>
</tr>
<tr>
<td>r4.xlarge</td>
<td>850</td>
<td>106.25</td>
<td>6,000</td>
</tr>
<tr>
<td>r4.2xlarge</td>
<td>1,700</td>
<td>212.5</td>
<td>12,000</td>
</tr>
<tr>
<td>r4.4xlarge</td>
<td>3,500</td>
<td>437.5</td>
<td>18,750</td>
</tr>
<tr>
<td>r4.8xlarge</td>
<td>7,000</td>
<td>875</td>
<td>37,500</td>
</tr>
<tr>
<td>r4.16xlarge</td>
<td>14,000</td>
<td>1,750</td>
<td>75,000</td>
</tr>
<tr>
<td>r5.large *</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5.xlarge *</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5.2xlarge *</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5.4xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5.8xlarge</td>
<td>6,800</td>
<td>850</td>
<td>30,000</td>
</tr>
<tr>
<td>r5.12xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>r5.16xlarge</td>
<td>13,600</td>
<td>1,700</td>
<td>60,000</td>
</tr>
<tr>
<td>r5.24xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>r5.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>r5a.large *</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>r5a.xlarge *</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>r5a.2xlarge *</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>r5a.4xlarge</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>r5a.8xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>r5a.12xlarge</td>
<td>6,780</td>
<td>847.5</td>
<td>30,000</td>
</tr>
<tr>
<td>r5a.16xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>r5a.24xlarge</td>
<td>13,570</td>
<td>1,696.25</td>
<td>60,000</td>
</tr>
<tr>
<td>r5ad.large *</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>Instance size</td>
<td>Maximum bandwidth (Mbps)</td>
<td>Maximum throughput (MB/s, 128 KiB I/O)</td>
<td>Maximum IOPS (16 KiB I/O)</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------</td>
<td>----------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>r5ad.xlarge</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>r5ad.2xlarge</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>r5ad.4xlarge</td>
<td>2,880</td>
<td>360</td>
<td>16,000</td>
</tr>
<tr>
<td>r5ad.8xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>r5ad.12xlarge</td>
<td>6,780</td>
<td>847.5</td>
<td>30,000</td>
</tr>
<tr>
<td>r5ad.16xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>r5ad.24xlarge</td>
<td>13,570</td>
<td>1,696.25</td>
<td>60,000</td>
</tr>
<tr>
<td>r5b.large</td>
<td>10,000</td>
<td>1,250</td>
<td>43,333</td>
</tr>
<tr>
<td>r5b.xlarge</td>
<td>10,000</td>
<td>1,250</td>
<td>43,333</td>
</tr>
<tr>
<td>r5b.2xlarge</td>
<td>10,000</td>
<td>1,250</td>
<td>43,333</td>
</tr>
<tr>
<td>r5b.4xlarge</td>
<td>10,000</td>
<td>1,250</td>
<td>43,333</td>
</tr>
<tr>
<td>r5b.8xlarge</td>
<td>20,000</td>
<td>2,500</td>
<td>86,667</td>
</tr>
<tr>
<td>r5b.12xlarge</td>
<td>30,000</td>
<td>3,750</td>
<td>130,000</td>
</tr>
<tr>
<td>r5b.16xlarge</td>
<td>40,000</td>
<td>5,000</td>
<td>173,333</td>
</tr>
<tr>
<td>r5b.24xlarge</td>
<td>60,000</td>
<td>7,500</td>
<td>260,000</td>
</tr>
<tr>
<td>r5b.metal</td>
<td>60,000</td>
<td>7,500</td>
<td>260,000</td>
</tr>
<tr>
<td>r5d.large</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5d.xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5d.2xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5d.4xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5d.8xlarge</td>
<td>6,800</td>
<td>850</td>
<td>30,000</td>
</tr>
<tr>
<td>r5d.12xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>r5d.16xlarge</td>
<td>13,600</td>
<td>1,700</td>
<td>60,000</td>
</tr>
<tr>
<td>r5d.24xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>r5d.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>r5dn.large</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5dn.xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5dn.2xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5dn.4xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5dn.8xlarge</td>
<td>6,800</td>
<td>850</td>
<td>30,000</td>
</tr>
<tr>
<td>Instance size</td>
<td>Maximum bandwidth (Mbps)</td>
<td>Maximum throughput (MB/s, 128 KiB I/O)</td>
<td>Maximum IOPS (16 KiB I/O)</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------</td>
<td>---------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>r5dn.12xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>r5dn.16xlarge</td>
<td>13,600</td>
<td>1,700</td>
<td>60,000</td>
</tr>
<tr>
<td>r5dn.24xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>r5dn.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>r5n.large*</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5n.xlarge*</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5n.2xlarge*</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5n.4xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>18,750</td>
</tr>
<tr>
<td>r5n.8xlarge</td>
<td>6,800</td>
<td>850</td>
<td>30,000</td>
</tr>
<tr>
<td>r5n.12xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>r5n.16xlarge</td>
<td>13,600</td>
<td>1,700</td>
<td>60,000</td>
</tr>
<tr>
<td>r5n.24xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>r5n.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>t3.nano*</td>
<td>2,085</td>
<td>260.57</td>
<td>11,800</td>
</tr>
<tr>
<td>t3.micro*</td>
<td>2,085</td>
<td>260.57</td>
<td>11,800</td>
</tr>
<tr>
<td>t3.small*</td>
<td>2,085</td>
<td>260.57</td>
<td>11,800</td>
</tr>
<tr>
<td>t3.medium*</td>
<td>2,085</td>
<td>260.57</td>
<td>11,800</td>
</tr>
<tr>
<td>t3.large*</td>
<td>2,780</td>
<td>347.5</td>
<td>15,700</td>
</tr>
<tr>
<td>t3.xlarge*</td>
<td>2,780</td>
<td>347.5</td>
<td>15,700</td>
</tr>
<tr>
<td>t3.2xlarge*</td>
<td>2,780</td>
<td>347.5</td>
<td>15,700</td>
</tr>
<tr>
<td>t3a.nano*</td>
<td>2,085</td>
<td>260.57</td>
<td>11,800</td>
</tr>
<tr>
<td>t3a.micro*</td>
<td>2,085</td>
<td>260.57</td>
<td>11,800</td>
</tr>
<tr>
<td>t3a.small*</td>
<td>2,085</td>
<td>260.57</td>
<td>11,800</td>
</tr>
<tr>
<td>t3a.medium*</td>
<td>2,085</td>
<td>260.57</td>
<td>11,800</td>
</tr>
<tr>
<td>t3a.large*</td>
<td>2,780</td>
<td>347.5</td>
<td>15,700</td>
</tr>
<tr>
<td>t3a.xlarge*</td>
<td>2,780</td>
<td>347.5</td>
<td>15,700</td>
</tr>
<tr>
<td>t3a.2xlarge*</td>
<td>2,780</td>
<td>347.5</td>
<td>15,700</td>
</tr>
<tr>
<td>u-6tb1.56xlarge</td>
<td>38,000</td>
<td>4,750</td>
<td>160,000</td>
</tr>
<tr>
<td>u-6tb1.112xlarge</td>
<td>38,000</td>
<td>4,750</td>
<td>160,000</td>
</tr>
<tr>
<td>u-6tb1.metal</td>
<td>38,000</td>
<td>4,750</td>
<td>160,000</td>
</tr>
</tbody>
</table>
### EBS optimization

<table>
<thead>
<tr>
<th>Instance size</th>
<th>Maximum bandwidth (Mbps)</th>
<th>Maximum throughput (MB/s, 128 KiB I/O)</th>
<th>Maximum IOPS (16 KiB I/O)</th>
</tr>
</thead>
<tbody>
<tr>
<td>u-9tb1.112xlarge</td>
<td>38,000</td>
<td>4,750</td>
<td>160,000</td>
</tr>
<tr>
<td>u-9tb1.metal</td>
<td>38,000</td>
<td>4,750</td>
<td>160,000</td>
</tr>
<tr>
<td>u-12tb1.112xlarge</td>
<td>38,000</td>
<td>4,750</td>
<td>160,000</td>
</tr>
<tr>
<td>u-12tb1.metal</td>
<td>38,000</td>
<td>4,750</td>
<td>160,000</td>
</tr>
<tr>
<td>u-18tb1.metal</td>
<td>38,000</td>
<td>4,750</td>
<td>160,000</td>
</tr>
<tr>
<td>u-24tb1.metal</td>
<td>38,000</td>
<td>4,750</td>
<td>160,000</td>
</tr>
<tr>
<td>x1.16xlarge</td>
<td>7,000</td>
<td>875</td>
<td>40,000</td>
</tr>
<tr>
<td>x1.32xlarge</td>
<td>14,000</td>
<td>1,750</td>
<td>80,000</td>
</tr>
<tr>
<td>x1e.xlarge</td>
<td>500</td>
<td>62.5</td>
<td>3,700</td>
</tr>
<tr>
<td>x1e.2xlarge</td>
<td>1,000</td>
<td>125</td>
<td>7,400</td>
</tr>
<tr>
<td>x1e.4xlarge</td>
<td>1,750</td>
<td>218.75</td>
<td>10,000</td>
</tr>
<tr>
<td>x1e.8xlarge</td>
<td>3,500</td>
<td>437.5</td>
<td>20,000</td>
</tr>
<tr>
<td>x1e.16xlarge</td>
<td>7,000</td>
<td>875</td>
<td>40,000</td>
</tr>
<tr>
<td>x1e.32xlarge</td>
<td>14,000</td>
<td>1,750</td>
<td>80,000</td>
</tr>
<tr>
<td>z1d.large*</td>
<td>3,170</td>
<td>396.25</td>
<td>13,333</td>
</tr>
<tr>
<td>z1d.xlarge*</td>
<td>3,170</td>
<td>396.25</td>
<td>13,333</td>
</tr>
<tr>
<td>z1d.2xlarge</td>
<td>3,170</td>
<td>396.25</td>
<td>13,333</td>
</tr>
<tr>
<td>z1d.3xlarge</td>
<td>4,750</td>
<td>593.75</td>
<td>20,000</td>
</tr>
<tr>
<td>z1d.6xlarge</td>
<td>9,500</td>
<td>1,187.5</td>
<td>40,000</td>
</tr>
<tr>
<td>z1d.12xlarge</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
<tr>
<td>z1d.metal</td>
<td>19,000</td>
<td>2,375</td>
<td>80,000</td>
</tr>
</tbody>
</table>

* These instance types can support maximum performance for 30 minutes at least once every 24 hours. If you have a workload that requires sustained maximum performance for longer than 30 minutes, select an instance type according to baseline performance as shown in the following table.

<table>
<thead>
<tr>
<th>Instance size</th>
<th>Baseline bandwidth (Mbps)</th>
<th>Baseline throughput (MB/s, 128 KiB I/O)</th>
<th>Baseline IOPS (16 KiB I/O)</th>
</tr>
</thead>
<tbody>
<tr>
<td>c5.large</td>
<td>650</td>
<td>81.25</td>
<td>4,000</td>
</tr>
<tr>
<td>c5.xlarge</td>
<td>1,150</td>
<td>143.75</td>
<td>6,000</td>
</tr>
<tr>
<td>c5.2xlarge</td>
<td>2,300</td>
<td>287.5</td>
<td>10,000</td>
</tr>
<tr>
<td>c5a.large</td>
<td>200</td>
<td>25</td>
<td>800</td>
</tr>
<tr>
<td>c5a.xlarge</td>
<td>400</td>
<td>50</td>
<td>1,600</td>
</tr>
<tr>
<td>Instance size</td>
<td>Baseline bandwidth (Mbps)</td>
<td>Baseline throughput (MB/s, 128 KiB I/O)</td>
<td>Baseline IOPS (16 KiB I/O)</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------</td>
<td>----------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>c5a.2xlarge</td>
<td>800</td>
<td>100</td>
<td>3,200</td>
</tr>
<tr>
<td>c5a.4xlarge</td>
<td>1,580</td>
<td>198</td>
<td>6,600</td>
</tr>
<tr>
<td>c5ad.large</td>
<td>200</td>
<td>25</td>
<td>800</td>
</tr>
<tr>
<td>c5ad.xlarge</td>
<td>400</td>
<td>50</td>
<td>1,600</td>
</tr>
<tr>
<td>c5ad.2xlarge</td>
<td>800</td>
<td>100</td>
<td>3,200</td>
</tr>
<tr>
<td>c5ad.4xlarge</td>
<td>1,580</td>
<td>198</td>
<td>6,600</td>
</tr>
<tr>
<td>c5d.large</td>
<td>650</td>
<td>81.25</td>
<td>4,000</td>
</tr>
<tr>
<td>c5d.xlarge</td>
<td>1,150</td>
<td>143.75</td>
<td>6,000</td>
</tr>
<tr>
<td>c5d.2xlarge</td>
<td>2,300</td>
<td>287.5</td>
<td>10,000</td>
</tr>
<tr>
<td>c5n.large</td>
<td>650</td>
<td>81.25</td>
<td>4,000</td>
</tr>
<tr>
<td>c5n.xlarge</td>
<td>1,150</td>
<td>143.75</td>
<td>6,000</td>
</tr>
<tr>
<td>c5n.2xlarge</td>
<td>2,300</td>
<td>287.5</td>
<td>10,000</td>
</tr>
<tr>
<td>d3.xlarge</td>
<td>850</td>
<td>106.25</td>
<td>5,000</td>
</tr>
<tr>
<td>d3.2xlarge</td>
<td>1,700</td>
<td>212.5</td>
<td>10,000</td>
</tr>
<tr>
<td>d3en.large</td>
<td>425</td>
<td>53.125</td>
<td>2,500</td>
</tr>
<tr>
<td>d3en.xlarge</td>
<td>850</td>
<td>106.25</td>
<td>5,000</td>
</tr>
<tr>
<td>d3en.2xlarge</td>
<td>1,700</td>
<td>212.5</td>
<td>10,000</td>
</tr>
<tr>
<td>g4ad.xlarge</td>
<td>400</td>
<td>50</td>
<td>1,700</td>
</tr>
<tr>
<td>g4ad.2xlarge</td>
<td>800</td>
<td>100</td>
<td>3,400</td>
</tr>
<tr>
<td>g4ad.4xlarge</td>
<td>1,580</td>
<td>197.5</td>
<td>6,700</td>
</tr>
<tr>
<td>g4dn.xlarge</td>
<td>950</td>
<td>118.75</td>
<td>3,000</td>
</tr>
<tr>
<td>g4dn.2xlarge</td>
<td>1,150</td>
<td>143.75</td>
<td>6,000</td>
</tr>
<tr>
<td>i3en.large</td>
<td>577</td>
<td>72.1</td>
<td>3,000</td>
</tr>
<tr>
<td>i3en.xlarge</td>
<td>1,154</td>
<td>144.2</td>
<td>6,000</td>
</tr>
<tr>
<td>i3en.2xlarge</td>
<td>2,307</td>
<td>288.39</td>
<td>12,000</td>
</tr>
<tr>
<td>i3en.3xlarge</td>
<td>3,800</td>
<td>475</td>
<td>15,000</td>
</tr>
<tr>
<td>m5.large</td>
<td>650</td>
<td>81.25</td>
<td>3,600</td>
</tr>
<tr>
<td>m5.xlarge</td>
<td>1,150</td>
<td>143.75</td>
<td>6,000</td>
</tr>
<tr>
<td>m5.2xlarge</td>
<td>2,300</td>
<td>287.5</td>
<td>12,000</td>
</tr>
<tr>
<td>m5a.large</td>
<td>650</td>
<td>81.25</td>
<td>3,600</td>
</tr>
<tr>
<td>Instance size</td>
<td>Baseline bandwidth (Mbps)</td>
<td>Baseline throughput (MB/s, 128 KiB I/O)</td>
<td>Baseline IOPS (16 KiB I/O)</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------</td>
<td>----------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>m5a.xlarge</td>
<td>1,085</td>
<td>135.63</td>
<td>6,000</td>
</tr>
<tr>
<td>m5a.2xlarge</td>
<td>1,580</td>
<td>197.5</td>
<td>8,333</td>
</tr>
<tr>
<td>m5ad.large</td>
<td>650</td>
<td>81.25</td>
<td>3,600</td>
</tr>
<tr>
<td>m5ad.xlarge</td>
<td>1,085</td>
<td>135.63</td>
<td>6,000</td>
</tr>
<tr>
<td>m5ad.2xlarge</td>
<td>1,580</td>
<td>197.5</td>
<td>8,333</td>
</tr>
<tr>
<td>m5d.large</td>
<td>650</td>
<td>81.25</td>
<td>3,600</td>
</tr>
<tr>
<td>m5d.xlarge</td>
<td>1,150</td>
<td>143.75</td>
<td>6,000</td>
</tr>
<tr>
<td>m5d.2xlarge</td>
<td>2,300</td>
<td>287.5</td>
<td>12,000</td>
</tr>
<tr>
<td>m5dn.large</td>
<td>650</td>
<td>81.25</td>
<td>3,600</td>
</tr>
<tr>
<td>m5dn.xlarge</td>
<td>1,150</td>
<td>143.75</td>
<td>6,000</td>
</tr>
<tr>
<td>m5dn.2xlarge</td>
<td>2,300</td>
<td>287.5</td>
<td>12,000</td>
</tr>
<tr>
<td>m5n.large</td>
<td>650</td>
<td>81.25</td>
<td>3,600</td>
</tr>
<tr>
<td>m5n.xlarge</td>
<td>1,150</td>
<td>143.75</td>
<td>6,000</td>
</tr>
<tr>
<td>m5n.2xlarge</td>
<td>2,300</td>
<td>287.5</td>
<td>12,000</td>
</tr>
<tr>
<td>m5zn.large</td>
<td>800</td>
<td>100</td>
<td>3,333</td>
</tr>
<tr>
<td>m5zn.xlarge</td>
<td>1,580</td>
<td>195.5</td>
<td>6,667</td>
</tr>
<tr>
<td>m6i.large</td>
<td>650</td>
<td>81.25</td>
<td>3,600</td>
</tr>
<tr>
<td>m6i.xlarge</td>
<td>1,250</td>
<td>156.25</td>
<td>6,000</td>
</tr>
<tr>
<td>m6i.2xlarge</td>
<td>2,500</td>
<td>312.5</td>
<td>12,000</td>
</tr>
<tr>
<td>m6i.4xlarge</td>
<td>5,000</td>
<td>625</td>
<td>20,000</td>
</tr>
<tr>
<td>r5.large</td>
<td>650</td>
<td>81.25</td>
<td>3,600</td>
</tr>
<tr>
<td>r5.xlarge</td>
<td>1,150</td>
<td>143.75</td>
<td>6,000</td>
</tr>
<tr>
<td>r5.2xlarge</td>
<td>2,300</td>
<td>287.5</td>
<td>12,000</td>
</tr>
<tr>
<td>r5a.large</td>
<td>650</td>
<td>81.25</td>
<td>3,600</td>
</tr>
<tr>
<td>r5a.xlarge</td>
<td>1,085</td>
<td>135.63</td>
<td>6,000</td>
</tr>
<tr>
<td>r5a.2xlarge</td>
<td>1,580</td>
<td>197.5</td>
<td>8,333</td>
</tr>
<tr>
<td>r5ad.large</td>
<td>650</td>
<td>81.25</td>
<td>3,600</td>
</tr>
<tr>
<td>r5ad.xlarge</td>
<td>1,085</td>
<td>135.63</td>
<td>6,000</td>
</tr>
<tr>
<td>r5ad.2xlarge</td>
<td>1,580</td>
<td>197.5</td>
<td>8,333</td>
</tr>
<tr>
<td>r5b.large</td>
<td>1,250</td>
<td>156.25</td>
<td>5,417</td>
</tr>
</tbody>
</table>
### EBS optimization

The following table lists the instance types that support EBS optimization but EBS optimization is not enabled by default. You can enable EBS optimization when you launch these instances or after they are running. Instances must have EBS optimization enabled to achieve the level of performance.

<table>
<thead>
<tr>
<th>Instance size</th>
<th>Baseline bandwidth (Mbps)</th>
<th>Baseline throughput (MB/s, 128 KiB I/O)</th>
<th>Baseline IOPS (16 KiB I/O)</th>
</tr>
</thead>
<tbody>
<tr>
<td>r5b.xlarge</td>
<td>2,500</td>
<td>312.5</td>
<td>10,833</td>
</tr>
<tr>
<td>r5b.2xlarge</td>
<td>5,000</td>
<td>625</td>
<td>21,667</td>
</tr>
<tr>
<td>r5d.large</td>
<td>650</td>
<td>81.25</td>
<td>3,600</td>
</tr>
<tr>
<td>r5d.xlarge</td>
<td>1,150</td>
<td>143.75</td>
<td>6,000</td>
</tr>
<tr>
<td>r5d.2xlarge</td>
<td>2,300</td>
<td>287.5</td>
<td>12,000</td>
</tr>
<tr>
<td>r5dn.large</td>
<td>650</td>
<td>81.25</td>
<td>3,600</td>
</tr>
<tr>
<td>r5dn.xlarge</td>
<td>1,150</td>
<td>143.75</td>
<td>6,000</td>
</tr>
<tr>
<td>r5dn.2xlarge</td>
<td>2,300</td>
<td>287.5</td>
<td>12,000</td>
</tr>
<tr>
<td>r5n.large</td>
<td>650</td>
<td>81.25</td>
<td>3,600</td>
</tr>
<tr>
<td>r5n.xlarge</td>
<td>1,150</td>
<td>143.75</td>
<td>6,000</td>
</tr>
<tr>
<td>r5n.2xlarge</td>
<td>2,300</td>
<td>287.5</td>
<td>12,000</td>
</tr>
<tr>
<td>t3.nano</td>
<td>43</td>
<td>5.43</td>
<td>250</td>
</tr>
<tr>
<td>t3.micro</td>
<td>87</td>
<td>10.86</td>
<td>500</td>
</tr>
<tr>
<td>t3.small</td>
<td>174</td>
<td>21.71</td>
<td>1,000</td>
</tr>
<tr>
<td>t3.medium</td>
<td>347</td>
<td>43.43</td>
<td>2,000</td>
</tr>
<tr>
<td>t3.large</td>
<td>695</td>
<td>86.86</td>
<td>4,000</td>
</tr>
<tr>
<td>t3.xlarge</td>
<td>695</td>
<td>86.86</td>
<td>4,000</td>
</tr>
<tr>
<td>t3.2xlarge</td>
<td>695</td>
<td>86.86</td>
<td>4,000</td>
</tr>
<tr>
<td>t3a.nano</td>
<td>45</td>
<td>5.63</td>
<td>250</td>
</tr>
<tr>
<td>t3a.micro</td>
<td>90</td>
<td>11.25</td>
<td>500</td>
</tr>
<tr>
<td>t3a.small</td>
<td>175</td>
<td>21.88</td>
<td>1,000</td>
</tr>
<tr>
<td>t3a.medium</td>
<td>350</td>
<td>43.75</td>
<td>2,000</td>
</tr>
<tr>
<td>t3a.large</td>
<td>695</td>
<td>86.86</td>
<td>4,000</td>
</tr>
<tr>
<td>t3a.xlarge</td>
<td>695</td>
<td>86.86</td>
<td>4,000</td>
</tr>
<tr>
<td>t3a.2xlarge</td>
<td>695</td>
<td>86.86</td>
<td>4,000</td>
</tr>
<tr>
<td>z1d.large</td>
<td>800</td>
<td>100</td>
<td>3,333</td>
</tr>
<tr>
<td>z1d.xlarge</td>
<td>1,580</td>
<td>197.5</td>
<td>6,667</td>
</tr>
</tbody>
</table>
described. When you enable EBS optimization for an instance that is not EBS-optimized by default, you pay an additional low, hourly fee for the dedicated capacity. For pricing information, see EBS-Optimized Instances on the Amazon EC2 Pricing, On-Demand Pricing page.

**Note**

You can also view this information programatically using the AWS CLI. For more information, see View instances types that support EBS optimization (p. 1359).

<table>
<thead>
<tr>
<th>Instance size</th>
<th>Maximum bandwidth (Mbps)</th>
<th>Maximum throughput (MB/s, 128 KiB I/O)</th>
<th>Maximum IOPS (16 KiB I/O)</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1.xlarge</td>
<td>1,000</td>
<td>125</td>
<td>8,000</td>
</tr>
<tr>
<td>c3.xlarge</td>
<td>500</td>
<td>62.5</td>
<td>4,000</td>
</tr>
<tr>
<td>c3.2xlarge</td>
<td>1,000</td>
<td>125</td>
<td>8,000</td>
</tr>
<tr>
<td>c3.4xlarge</td>
<td>2,000</td>
<td>250</td>
<td>16,000</td>
</tr>
<tr>
<td>g2.2xlarge</td>
<td>1,000</td>
<td>125</td>
<td>8,000</td>
</tr>
<tr>
<td>i2.xlarge</td>
<td>500</td>
<td>62.5</td>
<td>4,000</td>
</tr>
<tr>
<td>i2.2xlarge</td>
<td>1,000</td>
<td>125</td>
<td>8,000</td>
</tr>
<tr>
<td>i2.4xlarge</td>
<td>2,000</td>
<td>250</td>
<td>16,000</td>
</tr>
<tr>
<td>m1.large</td>
<td>500</td>
<td>62.5</td>
<td>4,000</td>
</tr>
<tr>
<td>m1.xlarge</td>
<td>1,000</td>
<td>125</td>
<td>8,000</td>
</tr>
<tr>
<td>m2.2xlarge</td>
<td>500</td>
<td>62.5</td>
<td>4,000</td>
</tr>
<tr>
<td>m2.4xlarge</td>
<td>1,000</td>
<td>125</td>
<td>8,000</td>
</tr>
<tr>
<td>m3.xlarge</td>
<td>500</td>
<td>62.5</td>
<td>4,000</td>
</tr>
<tr>
<td>m3.2xlarge</td>
<td>1,000</td>
<td>125</td>
<td>8,000</td>
</tr>
<tr>
<td>r3.xlarge</td>
<td>500</td>
<td>62.5</td>
<td>4,000</td>
</tr>
<tr>
<td>r3.2xlarge</td>
<td>1,000</td>
<td>125</td>
<td>8,000</td>
</tr>
<tr>
<td>r3.4xlarge</td>
<td>2,000</td>
<td>250</td>
<td>16,000</td>
</tr>
</tbody>
</table>

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.

**Get maximum performance**

You can use the EBSIOBalance% and EBSByteBalance% metrics to help you determine whether your instances are sized correctly. You can view these metrics in the CloudWatch console and set an alarm that is triggered based on a threshold you specify. These metrics are expressed as a percentage. Instances with a consistently low balance percentage are candidates to size up. Instances where the balance percentage never drops below 100% are candidates for downsizing. For more information, see Monitor your instances using CloudWatch (p. 839).

The high memory instances are designed to run large in-memory databases, including production deployments of the SAP HANA in-memory database, in the cloud. To maximize EBS performance,
use high memory instances with an even number of io1 or io2 volumes with identical provisioned performance. For example, for IOPS heavy workloads, use four io1 or io2 volumes with 40,000 provisioned IOPS to get the maximum 160,000 instance IOPS. Similarly, for throughput heavy workloads, use six io1 or io2 volumes with 48,000 provisioned IOPS to get the maximum 4,750 MB/s throughput. For additional recommendations, see Storage Configuration for SAP HANA.

Considerations

- G4dn, i3en, M5a, M5ad, R5a, R5ad, T3, T3a, and Z1d instances launched after February 26, 2020 provide the maximum performance listed in the table above. To get the maximum performance from an instance launched before February 26, 2020, stop and start it.
- C5, C5d, C5n, M5d, M5n, M5dn, R5, R5d, R5n, R5dn, and P3dn instances launched after December 3, 2019 provide the maximum performance listed in the table above. To get the maximum performance from an instance launched before December 3, 2019, stop and start it.
- u-6tb1.metal, u-9tb1.metal, and u-12tb1.metal instances launched after March 12, 2020 provide the performance in the table above. Instances of these types launched before March 12, 2020 might provide lower performance. To get the maximum performance from an instance launched before March 12, 2020, contact your account team to upgrade the instance at no additional cost.

View instances types that support EBS optimization

You can use the AWS CLI to view the instances types in the current Region that support EBS optimization.

To view the instance types that support EBS optimization and that have it enabled by default

Use the following `describe-instance-types` command.

```
C:\> aws ec2 describe-instance-types \
   --filters Name=ebs-info.ebs-optimized-support,Values=default --output=table
```

Example output for eu-west-1:

```
<table>
<thead>
<tr>
<th>DescribeInstanceTypes</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBSOptimized</td>
</tr>
<tr>
<td>default</td>
</tr>
<tr>
<td>default</td>
</tr>
<tr>
<td>default</td>
</tr>
<tr>
<td>default</td>
</tr>
<tr>
<td>default</td>
</tr>
</tbody>
</table>
|                ...
```

To view the instance types that support EBS optimization but do not have it enabled by default
Use the following `describe-instance-types` command.

```bash
C:\> aws ec2 describe-instance-types \
   --filters Name=ebs-info.ebs-optimized-support,Values=supported --output=table
```

Example output for `eu-west-1`:

```
<table>
<thead>
<tr>
<th>EBSOptimized</th>
<th>InstanceType</th>
<th>MaxBandwidth(Mb/s)</th>
<th>MaxIOPS</th>
<th>MaxThroughput(MB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>supported</td>
<td>m2.4xlarge</td>
<td>1000</td>
<td>8000</td>
<td>125.0</td>
</tr>
<tr>
<td>supported</td>
<td>i2.2xlarge</td>
<td>1000</td>
<td>8000</td>
<td>125.0</td>
</tr>
<tr>
<td>supported</td>
<td>r3.4xlarge</td>
<td>2000</td>
<td>16000</td>
<td>250.0</td>
</tr>
<tr>
<td>supported</td>
<td>m3.xlarge</td>
<td>500</td>
<td>4000</td>
<td>62.5</td>
</tr>
<tr>
<td>supported</td>
<td>r3.2xlarge</td>
<td>1000</td>
<td>8000</td>
<td>125.0</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Enable EBS optimization at launch**

You can enable optimization for an instance by setting its attribute for EBS optimization.

**To enable Amazon EBS optimization when launching an instance using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. Choose Launch Instance.
3. In Step 1: Choose an Amazon Machine Image (AMI), select an AMI.
4. In Step 2: Choose an Instance Type, select an instance type that is listed as supporting Amazon EBS optimization.
5. In Step 3: Configure Instance Details, complete the fields that you need and choose **Launch as EBS-optimized instance**. If the instance type that you selected in the previous step doesn't support Amazon EBS optimization, this option is not present. If the instance type that you selected is Amazon EBS-optimized by default, this option is selected and you can't deselect it.
6. Follow the directions to complete the wizard and launch your instance.

**To enable EBS optimization when launching an instance using the command line**

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

- `run-instances` with `--ebs-optimized` (AWS CLI)
- `New-EC2Instance` with `-EbsOptimized` (AWS Tools for Windows PowerShell)

**Enable EBS optimization for an existing instance**

You can enable or disable optimization for an existing instance by modifying its Amazon EBS-optimized instance attribute. If the instance is running, you must stop it first.

**Warning**

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.
To enable EBS optimization for an existing instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances, and select the instance.
3. To stop the instance, choose Actions, Instance state, Stop instance. It can take a few minutes for the instance to stop.
4. With the instance still selected, choose Actions, Instance settings, Change instance type.
5. For Change Instance Type, do one of the following:
   - If the instance type of your instance is Amazon EBS–optimized by default, EBS-optimized is selected and you can't change it. You can choose Cancel, because Amazon EBS optimization is already enabled for the instance.
   - If the instance type of your instance supports Amazon EBS optimization, choose EBS-optimized and then choose Apply.
   - If the instance type of your instance does not support Amazon EBS optimization, you can't choose EBS-optimized. You can select an instance type from Instance type that supports Amazon EBS optimization, choose EBS-optimized, and then choose Apply.
6. Choose Instance state, Start instance.

To enable EBS optimization for an existing instance using the command line

1. If the instance is running, use one of the following commands to stop it:
   - stop-instances (AWS CLI)
   - Stop-EC2Instance (AWS Tools for Windows PowerShell)
2. To enable EBS optimization, use one of the following commands with the corresponding option:
   - modify-instance-attribute with --ebs-optimized (AWS CLI)
   - Edit-EC2InstanceAttribute with -EbsOptimized (AWS Tools for Windows PowerShell)

Amazon EBS volume performance on 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.

AWS updates to the performance of EBS volume types might not immediately take effect on your existing volumes. To see full performance on an older volume, you might first need to perform a ModifyVolume action on it. For more information, see Modifying the Size, IOPS, or Type of an EBS Volume on Windows.

Contents

- Amazon EBS performance tips (p. 1362)
- I/O characteristics and monitoring (p. 1363)
Amazon EBS performance tips

These tips represent best practices for getting optimal performance from your EBS volumes in a variety of user scenarios.

**Use EBS-optimized instances**

On instances without support for EBS-optimized throughput, network traffic can contend with traffic between your instance and your EBS volumes; on EBS-optimized instances, the two types of traffic are kept separate. Some EBS-optimized instance configurations incur an extra cost (such as C3, R3, and M3), while others are always EBS-optimized at no extra cost (such as M4, C4, C5, and D2). For more information, see Amazon EBS–optimized instances (p. 1344).

**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. 1363).

**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 created from a snapshot. You can avoid this performance hit using one of the following options:

- Access each block prior to putting the volume into production. This process is called initialization (formerly known as pre-warming). For more information, see Initialize Amazon EBS volumes (p. 1366).
- Enable fast snapshot restore on a snapshot to ensure that the EBS volumes created from it are fully-initialized at creation and instantly deliver all of their provisioned performance. For more information, see Amazon EBS fast snapshot restore (p. 1339).

**Factors that can degrade HDD performance**

When you create a snapshot of a Throughput Optimized HDD (st1) or Cold HDD (sc1) volume, performance may drop as far as the volume's baseline value while the snapshot is in progress. This behavior is specific to these volume types. Other factors that can limit performance include driving more throughput than the instance can support, the performance penalty encountered while initializing volumes created from a snapshot, and excessive amounts of small, random I/O on the volume. For more information about calculating throughput for HDD volumes, see Amazon EBS volume types (p. 1163).

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. 1363)

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 volumes together in a RAID 0 configuration to use the available bandwidth for these instances. For more information, see RAID configuration on Windows (p. 1368).

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 Monitor the status of your volumes (p. 1196).

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 gp3) and Provisioned IOPS SSD (io1 and io2)—deliver consistent performance whether an I/O operation is random or sequential. HDD-backed volumes—Throughput Optimized HDD (st1) and Cold HDD (sc1)—deliver optimal performance only when I/O operations are large and sequential. To understand how SSD and HDD volumes will perform in your application, it is important to know the connection between demand on the volume, the quantity of IOPS available to it, the time it takes for an I/O operation to complete, and the volume's throughput limits.

Topics

- IOPS (p. 1363)
- Volume queue length and latency (p. 1364)
- I/O size and volume throughput limits (p. 1365)
- Monitor I/O characteristics using CloudWatch (p. 1365)
- Related resources (p. 1366)

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 sequential, Amazon EBS attempts to merge them into a single I/O operation up to the maximum I/O size. Similarly, when I/O operations are larger than the maximum I/O size, Amazon EBS attempts to split them into smaller I/O operations. The following table shows some examples.

<table>
<thead>
<tr>
<th>Volume type</th>
<th>Maximum I/O size</th>
<th>I/O operations from your application</th>
<th>Number of IOPS</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSD</td>
<td>256 KiB</td>
<td>1 x 1024 KiB I/O operation</td>
<td>4 (1,024+256=4)</td>
<td>Amazon EBS splits the 1,024 I/O operation into four smaller 256 KiB operations.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 x sequential 32 KiB I/O operations</td>
<td>1 (8x32=256)</td>
<td>Amazon EBS merges the eight</td>
</tr>
</tbody>
</table>
### EBS Performance

<table>
<thead>
<tr>
<th>Volume type</th>
<th>Maximum I/O size</th>
<th>I/O operations from your application</th>
<th>Number of IOPS</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>8 random 32 KiB I/O operations</td>
<td>8</td>
<td>Amazon EBS counts random I/O operations separately.</td>
</tr>
<tr>
<td></td>
<td>1,024 KiB</td>
<td>1 x 1024 KiB I/O operation</td>
<td>1</td>
<td>The I/O operation is already equal to the maximum I/O size. It is not merged or split.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 x sequential 128 KiB I/O operations</td>
<td>1 (8x128=1,024)</td>
<td>Amazon EBS merges the eight sequential 128 KiB I/O operations into a single 1,024 KiB I/O operation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 random 32 KiB I/O operations</td>
<td>8</td>
<td>Amazon EBS counts random I/O operations separately.</td>
</tr>
</tbody>
</table>

Consequently, when you create an SSD-backed volume supporting 3,000 IOPS (either by provisioning a Provisioned IOPS SSD volume at 3,000 IOPS or by sizing a General Purpose SSD volume at 1,000 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 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 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 1,000 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 x 256 KiB = 250 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. 1163).

For smaller I/O operations, you may see a higher-than-provisioned IOPS value as measured from inside your instance. This happens when the instance operating system merges small I/O operations into a larger operation before passing them to Amazon EBS.

If your workload uses sequential I/Os on HDD-backed st1 and sc1 volumes, you may experience a higher than expected number of IOPS as measured from inside your instance. This happens when the instance operating system merges sequential I/Os and counts them in 1,024 KiB-sized units. If your workload uses small or random I/Os, you may experience a lower throughput than you expect. This is because we count each random, non-sequential I/O toward the total IOPS count, which can cause you to hit the volume's IOPS limit sooner than expected.

Whatever your EBS volume type, if you are not experiencing the IOPS or throughput you expect in your configuration, ensure that your EC2 instance bandwidth is not the limiting factor. You should always use a current-generation, EBS-optimized instance (or one that includes 10 Gb/s network connectivity) for optimal performance. For more information, see Amazon EBS–optimized instances (p. 1344). Another possible cause for not experiencing the expected IOPS is that you are not driving enough I/O to the EBS volumes.

Monitor I/O characteristics using CloudWatch

You can monitor these I/O characteristics with each volume's CloudWatch volume metrics (p. 1376). Important metrics to consider include the following:

- **BurstBalance**
- **VolumeReadBytes**
- **VolumeWriteBytes**
- **VolumeReadOps**
- **VolumeWriteOps**
- **VolumeQueueLength**

*BurstBalance* displays the burst bucket balance for gp2, st1, and sc1 volumes as a percentage of the remaining balance. When your burst bucket is depleted, volume I/O (for gp2 volumes) or volume throughput (for st1 and sc1 volumes) is throttled to the baseline. Check the *BurstBalance* value to determine whether your volume is being throttled for this reason. For a complete list of the available Amazon EBS metrics, see Amazon EBS metrics (p. 1375) and Amazon EBS metrics for Nitro-based instances (p. 846).

HDD-backed st1 and sc1 volumes are designed to perform best with workloads that take advantage of the 1,024 KiB maximum I/O size. To determine your volume's average I/O size, divide VolumeWriteBytes by VolumeWriteOps. The same calculation applies to read operations. If average I/O size is below 64 KiB, increasing the size of the I/O operations sent to an st1 or sc1 volume should improve performance.

**Note**

If average I/O size is at or near 44 KiB, you might be using an instance or kernel without support for indirect descriptors. Any Linux kernel 3.8 and above has this support, as well as any current-generation instance.
If your I/O latency is higher than you require, check `VolumeQueueLength` to make sure your application is not trying to drive more IOPS than you have provisioned. If your application requires a greater number of IOPS than your volume can provide, you should consider using a larger gp2 volume with a higher base performance level or an io1 or io2 volume with more provisioned IOPS to achieve faster latencies.

**Related resources**

For more information about Amazon EBS I/O characteristics, see the following re:Invent presentation: Amazon EBS: Designing for Performance.

**Initialize Amazon EBS volumes**

Empty EBS volumes receive their maximum performance the moment that they are created and do not require initialization (formerly known as pre-warming).

For volumes that were created from snapshots, the storage blocks must be pulled down from Amazon S3 and written to the volume before you can access them. This preliminary action takes time and can cause a significant increase in the latency of I/O operations the first time each block is accessed. Volume performance is achieved after all blocks have been downloaded and written to the volume.

**Important**

While initializing Provisioned IOPS SSD volumes that were created from snapshots, the performance of the volume may drop below 50 percent of its expected level, which causes the volume to display a warning state in the I/O Performance status check. This is expected, and you can ignore the warning state on Provisioned IOPS SSD volumes while you are initializing them. For more information, see EBS volume status checks (p. 1197).

For most applications, amortizing the initialization cost over the lifetime of the volume is acceptable. To avoid this initial performance hit in a production environment, you can use one of the following options:

- Force the immediate initialization of the entire volume. For more information, see Initialize Amazon EBS volumes on Windows (p. 1366).
- Enable fast snapshot restore on a snapshot to ensure that the EBS volumes created from it are fully-initialized at creation and instantly deliver all of their provisioned performance. For more information, see Amazon EBS fast snapshot restore (p. 1339).

**Initialize 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 created 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.

For information about initializing Amazon EBS volumes on Linux, see Initializing Amazon EBS volumes on Linux.

Before using either tool, gather information about the disks on your system as follows:

**To gather information about the system disks**

1. Use the `wmic` command to list the available disks on your system:

   ```bash
   wmic diskdrive get size,deviceid
   ```

   The following is example output:

<table>
<thead>
<tr>
<th>DeviceID</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>.\PHYSICALDRIVE2</td>
<td>80517265920</td>
</tr>
</tbody>
</table>
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.

**Use dd**

Complete the following procedures to install and use `dd` to initialize a volume.

**Important considerations**

- Initializing a volume takes from several minutes up to several hours, depending on your EC2 instance bandwidth, the IOPS provisioned for the volume, and the size of the volume.
- Incorrect use of `dd` can easily destroy a volume's data. Be sure to follow this procedure precisely.

**To 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 enables you to initialize Amazon EBS volumes that have been created from snapshots. The most recent beta versions support the `/dev/null` virtual device. If you install an earlier version, you can use the `nul` virtual device instead. Full documentation is available at [http://www.chrysocome.net/dd](http://www.chrysocome.net/dd).

1. Download the most recent binary version of `dd` for Windows from [http://www.chrysocome.net/dd](http://www.chrysocome.net/dd).
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 run the programs in that folder from anywhere.
   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.
5. Open a new command prompt window. The previous step doesn't update the environment variables in your current command prompt windows. The command prompt windows that you open now that you completed the previous step are updated.

**To initialize a volume using dd for Windows**

Run 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.

```
dd if=\\.PHYSICALDRIVE0 of=/dev/null bs=1M --progress --size
```

You might get an error if `dd` attempts to read beyond the end of the volume. You can safely ignore this error.
If you used an earlier version of the `dd` command, it does not support the `/dev/null` device. Instead, you can use the `nul` device as follows.

```
dd if=\\.\PHYSICALDRIVE\n of=nul bs=1M --progress --size
```

**Use fio**

Complete the following procedures to install and use `fio` to initialize a volume.

**To 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 created from snapshots. For more information, see https://github.com/axboe/fio.

1. Download the `fio MSI` installer (select the latest x86 or x64 build, then select Artifacts).
2. Install `fio`.

**To initialize a volume using fio for Windows**

1. Run a command similar to the following to initialize a volume:

```
fio --filename=\\.\PHYSICALDRIVE\n --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 Make an Amazon EBS volume available for use on Windows (p. 1187).

**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.

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. 1368)
- Create a RAID 0 array on Windows (p. 1369)
- Create snapshots of volumes in a RAID array (p. 1372)

**RAID configuration options**

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. Use RAID 0 when I/O performance is of the utmost
importance. With RAID 0, I/O is distributed across the volumes in a stripe. If you add a volume, you get the straight addition of throughput and IOPS. However, keep in mind that performance of the stripe is limited to the worst performing volume in the set, and that the loss of a single volume in the set results in a complete data loss for the array.

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. For example, two 500 GiB io1 volumes with 4,000 provisioned IOPS each create a 1000 GiB RAID 0 array with an available bandwidth of 8,000 IOPS and 1,000 MiB/s of throughput.

**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.

RAID 1 is also not recommended for use with Amazon EBS. RAID 1 requires more Amazon EC2 to Amazon EBS bandwidth than non-RAID configurations because the data is written to multiple volumes simultaneously. In addition, RAID 1 does not provide any write performance improvement.

**Create a RAID 0 array on Windows**

This documentation provides a basic RAID 0 setup example.

Before you perform this procedure, you need to decide how large your RAID 0 array should be and how many IOPS you want to provision.

Use the following procedure to create the RAID 0 array. Note that you can get directions for Linux instances from *Create a RAID 0 array on Linux* in the *Amazon EC2 User Guide for Linux Instances*.

**To create a RAID 0 array on Windows**

1. Create the Amazon EBS volumes for your array. For more information, see *Create an Amazon EBS volume* (p. 1183).

   **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.

2. Attach the Amazon EBS volumes to the instance that you want to host the array. For more information, see *Attach an Amazon EBS volume to an instance* (p. 1186).

3. Connect to your Windows instance. For more information, see *Connect to your Windows instance* (p. 413).

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
   ```
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
   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.
ii. Delete the volume.

```bash
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. 1370) for each disk you want to use in your array.

7. Verify that the disks you want to use are now dynamic. In this case, we’re using disks 1 and 2 for the RAID volume.

```bash
DISKPART> list disk

<table>
<thead>
<tr>
<th>Disk ###</th>
<th>Status</th>
<th>Size</th>
<th>Free</th>
<th>Dyn</th>
<th>Gpt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk 0</td>
<td>Online</td>
<td>30 GB</td>
<td>0 B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disk 1</td>
<td>Online</td>
<td>8 GB</td>
<td>0 B</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Disk 2</td>
<td>Online</td>
<td>8 GB</td>
<td>0 B</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>
```

8. Create your raid array. On Windows, a RAID 0 volume is referred to as a striped volume.

To create a striped volume array on disks 1 and 2, use the following command (note the stripe option to stripe the array):

```bash
DISKPART> create volume stripe disk=1,2
DiskPart successfully created the volume.
```


```bash
DISKPART> list volume

<table>
<thead>
<tr>
<th>Volume ###</th>
<th>Ltr</th>
<th>Label</th>
<th>Fs</th>
<th>Type</th>
<th>Size</th>
<th>Status</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume 0</td>
<td>C</td>
<td>NTFS</td>
<td>Partition</td>
<td>29 GB</td>
<td>Healthy</td>
<td>System</td>
<td></td>
</tr>
<tr>
<td>Volume 1</td>
<td>RAW</td>
<td>Stripe</td>
<td>15 GB</td>
<td>Healthy</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Note that the Type column now indicates that Volume 1 is 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.

```bash
DISKPART> select volume n
Volume n is the selected volume.
```

b. Format the volume.

**Note**

To perform a full format, omit the quick option.

```bash
DISKPART> format quick recommended label="My new volume"
100 percent completed
```
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.

**Create snapshots of volumes in a RAID array**

If you want to back up the data on the EBS volumes in a RAID array using snapshots, you must ensure that the snapshots are consistent. This is because the snapshots of these volumes are created independently. To restore EBS volumes in a RAID array from snapshots that are out of sync would degrade the integrity of the array.

To create a consistent set of snapshots for your RAID array, use EBS multi-volume snapshots. Multi-volume snapshots allow you to take point-in-time, data coordinated, and crash-consistent snapshots across multiple EBS volumes attached to an EC2 instance. You do not have to stop your instance to coordinate between volumes to ensure consistency because snapshots are automatically taken across multiple EBS volumes. For more information, see the steps for creating multi-volume snapshots under **Creating Amazon EBS snapshots**.

**Benchmark EBS volumes**

You can test the performance of Amazon EBS volumes by simulating I/O workloads. The process is as follows:

1. Launch an EBS-optimized instance.
2. Create new EBS volumes.
3. Attach the volumes to your EBS-optimized instance.
4. Configure and mount the block device.
5. Install a tool to benchmark I/O performance.
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. 1344)**.

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
Set up Provisioned IOPS SSD or General Purpose SSD volumes

To create Provisioned IOPS SSD (io1 and io2) or General Purpose SSD (gp2 and gp3) volumes using the Amazon EC2 console, for **Volume type**, choose **Provisioned IOPS SSD (io1)**, **Provisioned IOPS SSD (io2)**, **General Purpose SSD (gp2)**, or **General Purpose SSD (gp3)**. At the command line, specify `io1`, `io2`, `gp2`, or `gp3` for the `--volume-type` parameter. For `io1`, `io2`, and `gp3` volumes, specify the number of I/O operations per second (IOPS) for the `--iops` parameter. For more information, see Amazon EBS volume types (p. 1163) and Create an Amazon EBS volume (p. 1183).

Set 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 Create an Amazon EBS volume (p. 1183). For information about attaching these volumes to your instance, see Attach an Amazon EBS volume to an instance (p. 1186).

Install benchmark tools

The following table lists some of the possible tools you can use to benchmark the performance of EBS volumes.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiskSpd</td>
<td>DiskSpd is a storage performance tool from the Windows, Windows Server, and Cloud Server Infrastructure engineering teams at Microsoft. It is available for download at <a href="https://github.com/Microsoft/diskspd/releases">https://github.com/Microsoft/diskspd/releases</a>. After you download the diskspd.exe executable file, open a command prompt with administrative rights (by choosing &quot;Run as Administrator&quot;), and then navigate to the directory where you copied the diskspd.exe file. Copy the desired diskspd.exe executable file from the appropriate executable folder (amd64fre, armfre or x86fre) to a short, simple path like C:\DiskSpd. In most cases you will want the 64-bit version of DiskSpd from the amd64fre folder. The source code for DiskSpd is hosted on GitHub at: <a href="https://github.com/Microsoft/diskspd">https://github.com/Microsoft/diskspd</a>.</td>
</tr>
<tr>
<td>CrystalDiskMark</td>
<td>CrystalDiskMark is a simple disk benchmark software. It is available for download at <a href="https://crystalmark.info/en/software/crystaldiskmark/">https://crystalmark.info/en/software/crystaldiskmark/</a>.</td>
</tr>
</tbody>
</table>

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.

Choose the volume queue length

Choosing the best volume queue length based on your workload and volume type.

Queue length on SSD-backed volumes

To determine the optimal queue length for your workload on SSD-backed volumes, we recommend that you target a queue length of 1 for every 1000 IOPS available (baseline for General Purpose SSD volumes
and the provisioned amount for Provisioned IOPS SSD volumes). Then you can monitor your application performance and tune that value based on your application requirements.

Increasing the queue length is beneficial until you achieve the provisioned IOPS, throughput or optimal system queue length value, which is currently set to 32. For example, a volume with 3,000 provisioned IOPS should target a queue length of 3. You should experiment with tuning these values up or down to see what performs best for your application.

**Queue length on HDD-backed volumes**

To determine the optimal queue length for your workload on HDD-backed volumes, we recommend that you target a queue length of at least 4 while performing 1MiB sequential I/Os. Then you can monitor your application performance and tune that value based on your application requirements. For example, a 2 TiB st1 volume with burst throughput of 500 MiB/s and IOPS of 500 should target a queue length of 4, 8, or 16 while performing 1,024 KiB, 512 KiB, or 256 KiB sequential I/Os respectively. You should experiment with tuning these values 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.

**Disable C-states on Windows**

You can disable C-states on Windows as follows:

1. In PowerShell, get the current active power scheme.

   ```powershell
   $current_scheme = powercfg /getactivescheme
   ```

2. Get the power scheme GUID.

   ```powershell
   (Get-WmiObject -class Win32_PowerPlan -Namespace "root\cimv2\power" -Filter "ElementName='High performance'").InstanceID
   ```

3. Get the power setting GUID.

   ```powershell
   (Get-WmiObject -class Win32_PowerSetting -Namespace "root\cimv2\power" -Filter "ElementName='Processor idle disable'").InstanceID
   ```

4. Get the power setting subgroup GUID.

   ```powershell
   (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.

   ```powershell
   powercfg /setacvalueindex <power_scheme_guid> <power_setting_subgroup_guid> <power_setting_guid> 1
   ```

6. Set active scheme to ensure the settings are saved.

   ```powershell
   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 created from snapshots, be sure to initialize them before benchmarking. For more information, see Initialize Amazon EBS volumes (p. 1366).

When you are finished testing your volumes, see the following topics for help cleaning up: Delete an Amazon EBS volume (p. 1206) and Terminate your instance (p. 441).

Benchmark Provisioned IOPS SSD and General Purpose SSD volumes

Run DiskSpd on the volume that you created.

The following command will run a 30 second random I/O test using a 20GB test file located on the C: drive, with a 25% write and 75% read ratio, and an 8K block size. It will use eight worker threads, each with four outstanding I/Os, and a write entropy value seed of 1GB. The results of the test will be saved to a text file called DiskSpeedResults.txt. These parameters simulate a SQL Server OLTP workload.

```
diskspd -b8K -d30 -o4 -t8 -h -r -w25 -L -Z1G -c20G C:\iotest.dat > DiskSpeedResults.txt
```

For more information about interpreting the results, see this tutorial: Inspecting disk IO performance with DiskSpd.

Amazon CloudWatch metrics for Amazon EBS

Amazon CloudWatch metrics are statistical data that you can use to view, analyze, and set alarms on the operational behavior of your volumes.

Data is available automatically in 1-minute periods at no charge.

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 (1-minute periods). We recommend that you specify a period in your request that is equal to or greater 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.

Topics

- Amazon EBS metrics (p. 1375)
- Dimensions for Amazon EBS metrics (p. 1380)
- Graphs in the Amazon EC2 console (p. 1380)

Amazon EBS metrics

Amazon Elastic Block Store (Amazon EBS) sends data points to CloudWatch for several metrics. All Amazon EBS volume types automatically send 1-minute metrics to CloudWatch, but only when the volume is attached to an instance.
Volume metrics for volumes attached to all instance types

The AWS/EBS namespace includes the following metrics for EBS volumes that are attached to all instance types. To get information about the available disk space from the operating system on an instance, see View free disk space (p. 1192).

**Note**

- Some metrics have differences on instances that are built on the Nitro System. For a list of these instance types, see Instances built on the Nitro System (p. 146).
- The AWS/EC2 namespace includes additional Amazon EBS metrics for volumes that are attached to Nitro-based instances that are not bare metal instances. For more information about these metrics see, Amazon EBS metrics for Nitro-based instances (p. 846).

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VolumeReadBytes</td>
<td>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. The <code>Minimum</code> and <code>Maximum</code> statistics on this metric are supported only by volumes attached to Nitro-based instances. Units: Bytes</td>
</tr>
<tr>
<td>VolumeWriteBytes</td>
<td>Provides information on the write 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 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 <code>SampleCount</code> 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. The <code>Minimum</code> and <code>Maximum</code> statistics on this metric are supported only by volumes attached to Nitro-based instances. Units: Bytes</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>VolumeReadOps</td>
<td>The total number of read operations in a specified period of time. Note: read operations are counted on completion. 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. The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances. Units: Count</td>
</tr>
<tr>
<td>VolumeWriteOps</td>
<td>The total number of write operations in a specified period of time. Note: write operations are counted on completion. 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. The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances. Units: Count</td>
</tr>
<tr>
<td>VolumeTotalReadTime</td>
<td><strong>Note</strong> This metric is not supported with Multi-Attach enabled volumes. 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 1 minutes (60 seconds): if 150 operations completed during that period, and each operation took 1 second, the value would be 150 seconds. For Xen instances, data is reported only when there is read activity on the volume. The Average statistic on this metric is not relevant for volumes attached to Nitro-based instances. The Minimum and Maximum statistics on this metric are supported only by volumes attached to Nitro-based instances. Units: Seconds</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| VolumeTotalWriteTime    | **Note** This metric is not supported with Multi-Attach enabled volumes.  
                          | 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 1 minute (60 seconds): if 150 operations completed during that period, and each operation took 1 second, the value would be 150 seconds. For Xen instances, data is reported only when there is write activity on the volume.  
                          | The **Average** statistic on this metric is not relevant for volumes attached to Nitro-based instances.  
                          | The **Minimum** and **Maximum** statistics on this metric are supported only by volumes attached to Nitro-based instances.  
                          | Units: Seconds                                                                                                                             |
| VolumeIdleTime          | **Note** This metric is not supported with Multi-Attach enabled volumes.  
                          | The total number of seconds in a specified period of time when no read or write operations were submitted.  
                          | The **Average** statistic on this metric is not relevant for volumes attached to Nitro-based instances.  
                          | The **Minimum** and **Maximum** statistics on this metric are supported only by volumes attached to Nitro-based instances.  
                          | Units: Seconds                                                                                                                             |
| VolumeQueueLength       | The number of read and write operation requests waiting to be completed in a specified period of time.  
                          | The **Sum** statistic on this metric is not relevant for volumes attached to Nitro-based instances.  
                          | The **Minimum** and **Maximum** statistics on this metric are supported only by volumes attached to Nitro-based instances.  
<pre><code>                      | Units: Count                                                                                                                              |
</code></pre>
<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
</table>
| **VolumeThroughputPercentage**     | **Note**
This metric is not supported with Multi-Attach enabled volumes.

Used with Provisioned IOPS SSD volumes only. The percentage of I/O operations per second (IOPS) delivered of the total IOPS provisioned for an Amazon EBS volume. Provisioned IOPS SSD volumes deliver their provisioned performance 99.9 percent of the time.

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).

Units: Percent

| **VolumeConsumedReadWriteOps**     | 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.

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.

Units: Count

| **BurstBalance**                   | 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.

The **Sum** statistic on this metric is not relevant for volumes attached to instances built on the Nitro System.

If the baseline performance of the volume exceeds the maximum burst performance, credits are never spent. If the volume is attached to an instance built on the Nitro System, the burst balance is not reported. For other instances, the reported burst balance is 100%. For more information, see [I/O Credits and burst performance](p. 1166).

Units: Percent

**Volume metrics for volumes attached to Nitro-based instance types**

The **AWS/EC2** namespace includes additional Amazon EBS metrics for volumes that are attached to Nitro-based instances that are not bare metal instances. For more information about these metrics see, [Amazon EBS metrics for Nitro-based instances](p. 846).
Fast snapshot restore metrics

AWS/EBS namespace includes the following metrics for fast snapshot restore (p. 1339).

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastSnapshotRestoreCreditsBucketSize</td>
<td>The maximum number of volume create credits that can be accumulated. This metric is reported per snapshot per Availability Zone. The most meaningful statistic is <strong>Average</strong>. The results for the <strong>Minimum</strong> and <strong>Maximum</strong> statistics are the same as for <strong>Average</strong> and could be used instead.</td>
</tr>
<tr>
<td>FastSnapshotRestoreCreditsBalance</td>
<td>The number of volume create credits available. This metric is reported per snapshot per Availability Zone. The most meaningful statistic is <strong>Average</strong>. The results for the <strong>Minimum</strong> and <strong>Maximum</strong> statistics are the same as for <strong>Average</strong> and could be used instead.</td>
</tr>
</tbody>
</table>

Dimensions for Amazon EBS metrics

The supported dimension is the volume ID (**VolumeId**). All available statistics are filtered by volume ID.

For the **volume metrics** (p. 1376), the supported dimension is the volume ID (**VolumeId**). All available statistics are filtered by volume ID.

For the **fast snapshot restore metrics** (p. 1380), the supported dimensions are the snapshot ID (**SnapshotId**) and the Availability Zone (**AvailabilityZone**).

Graphs in the Amazon EC2 console

After you create a volume, you can view the volume's monitoring graphs in the Amazon EC2 console. Select a volume on the **Volumes** page in the console and choose **Monitoring**. The following table lists the graphs that are displayed. The column on the right describes how the raw data metrics from the CloudWatch API are used to produce each graph. The period for all the graphs is 5 minutes.

<table>
<thead>
<tr>
<th>Graph</th>
<th>Description using raw metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read Bandwidth (KiB/s)</td>
<td>Sum(VolumeReadBytes) / Period / 1024</td>
</tr>
<tr>
<td>Write Bandwidth (KiB/s)</td>
<td>Sum(VolumeWriteBytes) / Period / 1024</td>
</tr>
<tr>
<td>Read Throughput (IOPS)</td>
<td>Sum(VolumeReadOps) / Period</td>
</tr>
<tr>
<td>Write Throughput (IOPS)</td>
<td>Sum(VolumeWriteOps) / Period</td>
</tr>
<tr>
<td>Avg Queue Length (Operations)</td>
<td>Avg(VolumeQueueLength)</td>
</tr>
<tr>
<td>% Time Spent Idle</td>
<td>Sum(VolumeIdleTime) / Period × 100</td>
</tr>
</tbody>
</table>
| Avg Read Size (KiB/Operation) | Avg(VolumeReadBytes) / 1024  
For Nitro-based instances, the following formula derives Average Read Size using **CloudWatch Metric Math**:
Graph | Description using raw metrics
--- | ---
 | 
| (Sum(VolumeReadBytes) / Sum(VolumeReadOps)) / 1024  
The VolumeReadBytes and VolumeReadOps metrics are available in the EBS CloudWatch console.
|  
| Avg Write Size (KiB/Operation) | Avg(VolumeWriteBytes) / 1024  
For Nitro-based instances, the following formula derives Average Write Size using CloudWatch Metric Math:  
(Sum(VolumeWriteBytes) / Sum(VolumeWriteOps)) / 1024  
The VolumeWriteBytes and VolumeWriteOps metrics are available in the EBS CloudWatch console.
|  
| Avg Read Latency (ms/Operation) | Avg(VolumeTotalReadTime) × 1000  
For Nitro-based instances, the following formula derives Average Read Latency using CloudWatch Metric Math:  
(Sum(VolumeTotalReadTime) / Sum(VolumeReadOps)) × 1000  
The VolumeTotalReadTime and VolumeReadOps metrics are available in the EBS CloudWatch console.
|  
| Avg Write Latency (ms/Operation) | Avg(VolumeTotalWriteTime) × 1000  
For Nitro-based instances, the following formula derives Average Write Latency using CloudWatch Metric Math:  
(Sum(VolumeTotalWriteTime) / Sum(VolumeWriteOps)) × 1000  
The VolumeTotalWriteTime and VolumeWriteOps metrics are available in the EBS CloudWatch console.

For the average latency graphs and average size graphs, the average is calculated over the total number of operations (read or write, whichever is applicable to the graph) that completed during the period.

### Amazon CloudWatch Events for Amazon EBS

Amazon EBS emits notifications based on Amazon CloudWatch Events for a variety of volume, snapshot, and encryption status changes. With CloudWatch Events, you can establish rules that trigger programmatic actions in response to a change in volume, snapshot, or encryption key state. For example, when a snapshot is created, you can trigger an AWS Lambda function to share the completed snapshot with another account or copy it to another Region for disaster-recovery purposes.

Events in CloudWatch are represented as JSON objects. The fields that are unique to the event are contained in the "detail" section of the JSON object. The "event" field contains the event name. The "result" field contains the completed status of the action that triggered the event. For more information, see Event Patterns in CloudWatch Events in the Amazon CloudWatch Events User Guide.

For more information, see Using Events in the Amazon CloudWatch User Guide.
EBS CloudWatch events

Contents

- EBS volume events (p. 1382)
- EBS snapshot events (p. 1385)
- EBS volume modification events (p. 1388)
- EBS fast snapshot restore events (p. 1389)
- Using AWS Lambda to handle CloudWatch events (p. 1390)

EBS volume events

Amazon EBS sends events to CloudWatch Events when the following volume events occur.

Events

- Create volume (createVolume) (p. 1382)
- Delete volume (deleteVolume) (p. 1383)
- Volume attach or reattach (attachVolume, reattachVolume) (p. 1384)

Create volume (createVolume)

The createVolume event is sent to your AWS account when an action to create a volume completes. However it is not saved, logged, or archived. This event can have a result of either available or failed. Creation will fail if an invalid AWS 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": [
    ],
    "detail": {
        "result": "available",
        "cause": "",
        "event": "createVolume",
        "request-id": "012345670123-0123-0123-012345678901"
    }
}
```

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": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:012345678901:volume/vol-01234567"
    ],
    "detail": {
        "result": "failed",
        "cause": "EBS KMS Key was disabled",
        "event": "createVolume",
        "request-id": "012345670123-0123-0123-0123456789ab"
    }
}
```
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": "01234567-0123-0123-0123-012345678901",
  "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-1:012345678901:volume/vol-012345678901"],
  "detail": {
    "event": "createVolume",
    "result": "failed",
    "cause": "arn:aws:kms:sa-east-1:012345678901:key/01234567-0123-0123-0123-012345678901 is pending import.",
    "request-id": "01234567-0123-0123-0123-012345678901",
  }
}
```

Delete volume (deleteVolume)

The `deleteVolume` event is sent to your AWS account when an action to delete a volume completes. However it is not saved, logged, or archived. This event has the result `deleted`. If the deletion does not complete, the event is never sent.

Event data

The listing below is an example of a JSON object emitted by EBS for a successful `deleteVolume` event.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Volume Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": ["arn:aws:ec2:us-east-1:012345678901:volume/vol-012345678901"],
  "detail": {
    "result": "deleted",
    "cause": "",
    "event": "deleteVolume",
  }
}
```

Volume attach or reattach (attachVolume, reattachVolume)

The attachVolume or reattachVolume event is sent to your AWS account if a volume fails to attach or reattach to an instance. However it is not saved, logged, or archived. If you use a KMS key to encrypt an EBS volume and the KMS key becomes invalid, EBS will emit an event if that KMS 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.

```
{  
  "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-0123456789ab"  
  ],  
  "detail": {  
    "event": "attachVolume",  
    "result": "failed",  
    "cause": "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123456789ab 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-0123456789ab"  
  ],  
  "detail": {  
    "event": "reattachVolume",  
    "result": "failed",  
    "cause": "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123456789ab is pending deletion.",  
    "request-id": ""  
  }  
}
```
EBS snapshot events

Amazon EBS sends events to CloudWatch Events when the following volume events occur.

Events

- Create snapshot (createSnapshot) (p. 1385)
- Create snapshots (createSnapshots) (p. 1385)
- Copy snapshot (copySnapshot) (p. 1387)
- Share snapshot (shareSnapshot) (p. 1388)

Create snapshot (createSnapshot)

The createSnapshot event is sent to your AWS account when an action to create a snapshot completes. However it is not saved, logged, or archived. This event can have a result of either succeeded or failed.

Event data

The listing below is an example of a JSON object emitted by EBS for a successful createSnapshot event. In the detail section, the source field contains the ARN of the source volume. The startTime and endTime fields indicate when creation of the snapshot started and completed.

```json
{
    "version": "0",
    "id": "01234567-0123-0123-0123-012345678901",
    "detail-type": "EBS Snapshot Notification",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyyMMdd\THHmmssZ",
    "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": "yyyyMMdd\THHmmssZ",
        "endTime": "yyyyMMdd\THHmmssZ"
    }
}
```

Create snapshots (createSnapshots)

The createSnapshots event is sent to your AWS account when an action to create a multi-volume snapshot completes. This event can have a result of either succeeded or failed.

Event data

The listing below is an example of a JSON object emitted by EBS for a successful createSnapshots event. In the detail section, the source field contains the ARNs of the source volumes of the multi-
volume snapshot set. The startTime and endTime fields indicate when creation of the snapshot started and completed.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-012345678901",
    "detail-type": "EBS Multi-Volume Snapshots Completion Status",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
        "arn:aws:ec2::us-east-1:snapshot/snap-012345678"
    ],
    "detail": {
        "event": "createSnapshots",
        "result": "succeeded",
        "cause": ",",
        "request-id": ",",
        "startTime": "yyyy-mm-ddThh:mm:ssZ",
        "endTime": "yyyy-mm-ddThh:mm:ssZ",
        "snapshots": [
            {
                "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
                "source": "arn:aws:ec2::us-east-1:volume/vol-01234567",
                "status": "completed"
            },
            {
                "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-012345678",
                "source": "arn:aws:ec2::us-east-1:volume/vol-012345678",
                "status": "completed"
            }
        ]
    }
}
```

The listing below is an example of a JSON object emitted by EBS after a failed createSnapshots event. The cause for the failure was one or more snapshots for the multi-volume snapshot set failed to complete. The values of snapshot_id are the ARNs of the failed snapshots. startTime and endTime represent when the create-snapshots action started and ended.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-012345678901",
    "detail-type": "EBS Multi-Volume Snapshots Completion Status",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
        "arn:aws:ec2::us-east-1:snapshot/snap-012345678"
    ],
    "detail": {
        "event": "createSnapshots",
        "result": "failed",
        "cause": "Snapshot snap-01234567 is in status error",
        "request-id": ",",
        "startTime": "yyyy-mm-ddThh:mm:ssZ",
        "endTime": "yyyy-mm-ddThh:mm:ssZ",
        "snapshots": [
            {
            
            }
        ]
    }
}
```
Copy snapshot (copySnapshot)

The copySnapshot event is sent to your AWS account when an action to copy a snapshot completes. However it is not saved, logged, or archived. This event can have a result of either succeeded or failed.

Event data

The listing below is an example of a JSON object emitted by EBS after a successful copySnapshot event. The value of snapshot_id is the ARN of the newly created snapshot. In the detail section, the value of source is the ARN of the source snapshot. startTime and endTime represent when the copy-snapshot action started and ended.

```json
{
  "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",
    "arn:aws:ec2:eu-west-1::snapshot/snap-76543210",
    "arn:aws:ec2:us-east-1::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.

```json
{
  "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": [
  ],
  "detail": {
    "event": "copySnapshot",
    "result": "failed",
    "cause": "error",
    "request-id": "",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
  }
}
```
"region": "us-east-1",
"resources": [
   "arn:aws:ec2:us-west-2::snapshot/snap-01234567"
],
"detail": {
   "event": "copySnapshot",
   "result": "failed",
   "cause": "Source snapshot ID is not valid",
   "request-id": "",
   "snapshot_id": "arn:aws:ec2:us-west-2::snapshot/snap-01234567",
   "source": "arn:aws:ec2:eu-west-1::snapshot/snap-76543210",
   "startTime": "yyyy-mm-ddThh:mm:ssZ",
   "endTime": "yyyy-mm-ddThh:mm:ssZ"
}
}

Share snapshot (shareSnapshot)

The shareSnapshot event is sent to your AWS account when another account shares a snapshot with it. However it is not saved, logged, or archived. The result is always succeeded.

Event data

The following is an example of a JSON object emitted by EBS after a completed shareSnapshot event. In the detail section, the value of source is the AWS account number of the user that shared the snapshot with you. startTime and endTime represent when the share-snapshot action started and ended. The shareSnapshot event is emitted only when a private snapshot is shared with another user. Sharing a public snapshot does not trigger the event.

{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Snapshot Notification",
   "source": "aws.ec2",
   "account": "012345678901",
   "time": "yyyy-mm-ddThh:mm:ssZ",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-west-2::snapshot/snap-01234567"
   ],
   "detail": {
      "event": "shareSnapshot",
      "result": "succeeded",
      "cause": "",
      "request-id": "",
      "snapshot_id": "arn:aws:ec2:us-west-2::snapshot/snap-01234567",
      "source": "012345678901",
      "startTime": "yyyy-mm-ddThh:mm:ssZ",
      "endTime": "yyyy-mm-ddThh:mm:ssZ"
   }
}

EBS volume modification events

Amazon EBS sends modifyVolume events to CloudWatch Events when a volume is modified. However it is not saved, logged, or archived.

{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Volume Notification",
   "source": "aws.ec2",
   "account": "012345678901",
   "time": "yyyy-mm-ddThh:mm:ssZ",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-west-2::snapshot/snap-01234567"
   ],
   "detail": {
      "event": "modifyVolume",
      "result": "succeeded",
      "cause": "",
      "request-id": "",
      "volume_id": "arn:aws:ec2:us-west-2::volume/vol-01234567",
      "source": "012345678901",
      "startTime": "yyyy-mm-ddThh:mm:ssZ",
      "endTime": "yyyy-mm-ddThh:mm:ssZ"
   }
}
EBS fast snapshot restore events

Amazon EBS sends events to CloudWatch Events when the state of fast snapshot restore for a snapshot changes. Events are emitted on a best effort basis.

The following is example data for this event.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Fast Snapshot Restore State-change Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1::snapshot/snap-03a55cf6513fa1b6"
  ],
  "detail": {
    "snapshot-id": "snap-1234567890abcdef0",
    "state": "optimizing",
    "zone": "us-east-1a",
    "message": "Client.UserInitiated - Lifecycle state transition",
  }
}
```

The possible values for state are enabling, optimizing, enabled, disabling, and disabled.

The possible values for message are as follows:

Client.InvalidSnapshot.InvalidState - The requested snapshot transitioned to an invalid state (Error)

A request to enable fast snapshot restore failed and the state transitioned to disabling or disabled. Fast snapshot restore cannot be enabled for this snapshot.

Client.UserInitiated

The state successfully transitioned to enabling or disabling.

Client.UserInitiated - Lifecycle state transition

The state successfully transitioned to optimizing, enabled, or disabled.

Server.InsufficientCapacity - There was insufficient capacity available to satisfy the request

A request to enable fast snapshot restore failed due to insufficient capacity, and the state transitioned to disabling or disabled. Wait and then try again.
Server.InternalError - An internal error caused the operation to fail

A request to enable fast snapshot restore failed due to an internal error, and the state transitioned to disabling or disabled. Wait and then try again.

Client.InvalidSnapshot.InvalidState - The requested snapshot was deleted or access permissions were revoked

The fast snapshot restore state for the snapshot has transitioned to disabling or disabled because the snapshot was deleted or unshared by the snapshot owner. Fast snapshot restore cannot be enabled for a snapshot that has been deleted or is no longer shared with you.

Using AWS Lambda to handle CloudWatch events

You can use Amazon EBS and CloudWatch Events to automate your data-backup workflow. This requires you to create an IAM policy, a AWS Lambda function to handle the event, and an Amazon CloudWatch Events rule that matches incoming events and routes them to the Lambda function.

The following procedure uses the createSnapshot event to automatically copy a completed snapshot to another Region for disaster recovery.

To copy a completed snapshot to another Region

1. Create an IAM policy, such as the one shown in the following example, to provide permissions to use the CopySnapshot action and write to the CloudWatch Events log. Assign the policy to the IAM user that will handle the CloudWatch event.

```json
{
    "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.

```javascript
// 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';
```
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.

4. Choose Events, Create rule, Select event source, and Amazon EBS Snapshots.
5. For Specific Event(s), choose createSnapshot and for Specific Result(s), choose succeeded.
6. For Rule target, find and choose the sample function that you previously created.
7. Choose Target, Add Target.
8. For Lambda function, select the Lambda function that you previously created and choose Configure details.
9. On the Configure rule details page, type values for Name and Description. Select the State check box to activate the function (setting it to Enabled).
10. Choose Create rule.

Your rule should now appear on the Rules tab. In the example shown, the event that you configured should be emitted by EBS the next time you copy a snapshot.
Amazon EBS quotas

To view the quotas for your Amazon EBS resources, open the Service Quotas console at https://console.aws.amazon.com/servicequotas/. In the navigation pane, choose AWS services, and select Amazon Elastic Block Store (Amazon EBS).

For a list of Amazon EBS service quotas, see Amazon Elastic Block Store endpoints and quotas in the AWS General Reference.

Amazon EC2 instance store

An instance store provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.

An instance store consists of one or more instance store volumes exposed as block devices. The size of an instance store as well as the number of devices available varies by instance type.

The virtual devices for instance store volumes are ephemeral[0–23]. Instance types that support one instance store volume have ephemeral0. Instance types that support two instance store volumes have ephemeral0 and ephemeral1, and so on.

Contents
- Instance store lifetime (p. 1393)
- Instance store volumes (p. 1393)
- Add instance store volumes to your EC2 instance (p. 1400)
- SSD instance store volumes (p. 1403)
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 hibernates
- 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, hibernate, or terminate an instance, every block of storage in the instance store is reset. Therefore, your data cannot be accessed through the instance store of another instance.

If you create an AMI from an instance, the data on its instance store volumes isn't preserved and isn't present on the instance store volumes of the instances that you launch from the AMI.

If you change the instance type, an instance store will not be attached to the new instance type. For more information, see Change the instance type (p. 231).

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. 1400).

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. 1403).

The data on NVMe instance store volumes and some HDD instance store volumes is encrypted at rest. For more information, see Data protection in Amazon EC2 (p. 1056).

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.

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Instance store volumes</th>
<th>Type</th>
<th>Needs initialization*</th>
<th>TRIM support**</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1.medium</td>
<td>1 x 350 GB</td>
<td>HDD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c1.xlarge</td>
<td>4 x 420 GB (1.6 TB)</td>
<td>HDD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c3.large</td>
<td>2 x 16 GB (32 GB)</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c3.xlarge</td>
<td>2 x 40 GB (80 GB)</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Instance type</td>
<td>Instance store volumes</td>
<td>Type</td>
<td>Needs initialization*</td>
<td>TRIM support**</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------</td>
<td>---------</td>
<td>------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>c3.2xlarge</td>
<td>2 x 80 GB (160 GB)</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c3.4xlarge</td>
<td>2 x 160 GB (320 GB)</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c3.8xlarge</td>
<td>2 x 320 GB (640 GB)</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5ad.large</td>
<td>1 x 75 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5ad.xlarge</td>
<td>1 x 150 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5ad.2xlarge</td>
<td>1 x 300 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5ad.4xlarge</td>
<td>2 x 300 GB (600 GB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5ad.8xlarge</td>
<td>2 x 600 GB (1.2 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5ad.12xlarge</td>
<td>2 x 900 GB (1.8 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5ad.16xlarge</td>
<td>2 x 1,200 GB (2.4 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5ad.24xlarge</td>
<td>2 x 1,900 GB (3.8 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5d.large</td>
<td>1 x 50 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5d.xlarge</td>
<td>1 x 100 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5d.2xlarge</td>
<td>1 x 200 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5d.4xlarge</td>
<td>1 x 400 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5d.9xlarge</td>
<td>1 x 900 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5d.12xlarge</td>
<td>2 x 900 GB (1.8 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5d.18xlarge</td>
<td>2 x 900 GB (1.8 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5d.24xlarge</td>
<td>4 x 900 GB (3.6 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>c5d.metal</td>
<td>4 x 900 GB (3.6 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>cc2.8xlarge</td>
<td>4 x 840 GB (3.36 TB)</td>
<td>HDD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>cr1.8xlarge</td>
<td>2 x 120 GB (240 GB)</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>d2.xlarge</td>
<td>3 x 2,000 GB (6 TB)</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d2.2xlarge</td>
<td>6 x 2,000 GB (12 TB)</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d2.4xlarge</td>
<td>12 x 2,000 GB (24 TB)</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d2.8xlarge</td>
<td>24 x 2,000 GB (48 TB)</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d3.xlarge</td>
<td>3 x 1,980 GB</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d3.2xlarge</td>
<td>6 x 1,980 GB</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d3.4xlarge</td>
<td>12 x 1,980 GB</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d3.8xlarge</td>
<td>24 x 1,980 GB</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instance type</td>
<td>Instance store volumes</td>
<td>Type</td>
<td>Needs initialization*</td>
<td>TRIM support**</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------</td>
<td>--------</td>
<td>------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>d3en.large</td>
<td>1 x 13,980 GB</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d3en.xlarge</td>
<td>2 x 13,980 GB</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d3en.2xlarge</td>
<td>4 x 13,980 GB</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d3en.4xlarge</td>
<td>8 x 13,980 GB</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d3en.6xlarge</td>
<td>12 x 13,980 GB</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d3en.8xlarge</td>
<td>16 x 13,980 GB</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d3en.12xlarge</td>
<td>24 x 13,980 GB</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>f1.2xlarge</td>
<td>1 x 470 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>f1.4xlarge</td>
<td>1 x 940 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>f1.16xlarge</td>
<td>4 x 940 GB (3.76 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>g2.2xlarge</td>
<td>1 x 60 GB</td>
<td>SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>g2.8xlarge</td>
<td>2 x 120 GB (240 GB)</td>
<td>SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>g4ad.xlarge</td>
<td>1 x 150 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>g4ad.2xlarge</td>
<td>1 x 300 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>g4ad.4xlarge</td>
<td>1 x 600 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>g4ad.8xlarge</td>
<td>1 x 1,200 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>g4ad.16xlarge</td>
<td>2 x 1,200 GB (2.4 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>g4dn.xlarge</td>
<td>1 x 125 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>g4dn.2xlarge</td>
<td>1 x 225 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>g4dn.4xlarge</td>
<td>1 x 225 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>g4dn.8xlarge</td>
<td>1 x 900 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>g4dn.12xlarge</td>
<td>1 x 900 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>g4dn.16xlarge</td>
<td>1 x 900 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>g4dn.metal</td>
<td>2 x 900 GB (1.8 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>h1.2xlarge</td>
<td>1 x 2,000 GB (2 TB)</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h1.4xlarge</td>
<td>2 x 2,000 GB (4 TB)</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h1.8xlarge</td>
<td>4 x 2,000 GB (8 TB)</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h1.16xlarge</td>
<td>8 x 2,000 GB (16 TB)</td>
<td>HDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hs1.8xlarge</td>
<td>24 x 2,000 GB (48 TB)</td>
<td>HDD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>i2.xlarge</td>
<td>1 x 800 GB</td>
<td>SSD</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Instance type</td>
<td>Instance store volumes</td>
<td>Type</td>
<td>Needs initialization*</td>
<td>TRIM support**</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------</td>
<td>--------</td>
<td>------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>i2.2xlarge</td>
<td>2 x 800 GB (1.6 TB)</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i2.4xlarge</td>
<td>4 x 800 GB (3.2 TB)</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i2.8xlarge</td>
<td>8 x 800 GB (6.4 TB)</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3.large</td>
<td>1 x 475 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3.xlarge</td>
<td>1 x 950 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3.2xlarge</td>
<td>1 x 1,900 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3.4xlarge</td>
<td>2 x 1,900 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3.8xlarge</td>
<td>4 x 1,900 GB (7.6 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3.16xlarge</td>
<td>8 x 1,900 GB (15.2 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3.metal</td>
<td>8 x 1,900 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3en.large</td>
<td>1 x 1,250 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3en.xlarge</td>
<td>1 x 2,500 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3en.2xlarge</td>
<td>2 x 2,500 GB (5 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3en.3xlarge</td>
<td>1 x 7,500 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3en.6xlarge</td>
<td>2 x 7,500 GB (15 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3en.12xlarge</td>
<td>4 x 7,500 GB (30 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3en.24xlarge</td>
<td>8 x 7,500 GB (60 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>i3en.metal</td>
<td>8 x 7,500 GB (60 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m1.small</td>
<td>1 x 160 GB</td>
<td>HDD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m1.medium</td>
<td>1 x 410 GB</td>
<td>HDD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m1.large</td>
<td>2 x 420 GB (840 GB)</td>
<td>HDD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m1.xlarge</td>
<td>4 x 420 GB (1.6 TB)</td>
<td>HDD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m2.xlarge</td>
<td>1 x 420 GB</td>
<td>HDD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m2.2xlarge</td>
<td>1 x 850 GB</td>
<td>HDD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m2.4xlarge</td>
<td>2 x 840 GB (1.68 TB)</td>
<td>HDD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m3.medium</td>
<td>1 x 4 GB</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m3.large</td>
<td>1 x 32 GB</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m3.xlarge</td>
<td>2 x 40 GB (80 GB)</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m3.2xlarge</td>
<td>2 x 80 GB (160 GB)</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5ad.large</td>
<td>1 x 75 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Instance type</td>
<td>Instance store volumes</td>
<td>Type</td>
<td>Needs initialization*</td>
<td>TRIM support**</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------</td>
<td>------------</td>
<td>------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>m5ad.xlarge</td>
<td>1 x 150 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5ad.2xlarge</td>
<td>1 x 300 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5ad.4xlarge</td>
<td>2 x 300 GB (600 GB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5ad.8xlarge</td>
<td>2 x 600 GB (1.2 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5ad.12xlarge</td>
<td>2 x 900 GB (1.8 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5ad.16xlarge</td>
<td>4 x 600 GB (2.4 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5ad.24xlarge</td>
<td>4 x 900 GB (3.6 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5d.large</td>
<td>1 x 75 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5d.xlarge</td>
<td>1 x 150 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5d.2xlarge</td>
<td>1 x 300 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5d.4xlarge</td>
<td>2 x 300 GB (600 GB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5d.8xlarge</td>
<td>2 x 600 GB (1.2 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5d.12xlarge</td>
<td>2 x 900 GB (1.8 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5d.16xlarge</td>
<td>4 x 600 GB (2.4 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5d.24xlarge</td>
<td>4 x 900 GB (3.6 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5d.metal</td>
<td>4 x 900 GB (3.6 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5dn.large</td>
<td>1 x 75 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5dn.xlarge</td>
<td>1 x 150 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5dn.2xlarge</td>
<td>1 x 300 GB</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5dn.4xlarge</td>
<td>2 x 300 GB (600 GB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5dn.8xlarge</td>
<td>2 x 600 GB (1.2 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5dn.12xlarge</td>
<td>2 x 900 GB (1.8 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5dn.16xlarge</td>
<td>4 x 600 GB (2.4 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5dn.24xlarge</td>
<td>4 x 900 GB (3.6 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>m5dn.metal</td>
<td>4 x 900 GB (3.6 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>p3dn.24xlarge</td>
<td>2 x 900 GB (1.8 TB)</td>
<td>NVMe SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>r3.large</td>
<td>1 x 32 GB</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>r3.xlarge</td>
<td>1 x 80 GB</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>r3.2xlarge</td>
<td>1 x 160 GB</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>r3.4xlarge</td>
<td>1 x 320 GB</td>
<td>SSD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Instance type</td>
<td>Instance store volumes</td>
<td>Type</td>
<td>Needs initialization*</td>
<td>TRIM support**</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------</td>
<td>--------------</td>
<td>------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>r3.8xlarge</td>
<td>2 x 320 GB (640 GB)</td>
<td>SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5ad.large</td>
<td>1 x 75 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5ad.xlarge</td>
<td>1 x 150 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5ad.2xlarge</td>
<td>1 x 300 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5ad.4xlarge</td>
<td>2 x 300 GB (600 GB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5ad.8xlarge</td>
<td>2 x 600 GB (1.2 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5ad.12xlarge</td>
<td>2 x 900 GB (1.8 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5ad.16xlarge</td>
<td>4 x 600 GB (2.4 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5ad.24xlarge</td>
<td>4 x 900 GB (3.6 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5d.large</td>
<td>1 x 75 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5d.xlarge</td>
<td>1 x 150 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5d.2xlarge</td>
<td>1 x 300 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5d.4xlarge</td>
<td>2 x 300 GB (600 GB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5d.8xlarge</td>
<td>2 x 600 GB (1.2 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5d.12xlarge</td>
<td>2 x 900 GB (1.8 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5d.16xlarge</td>
<td>4 x 600 GB (2.4 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5d.24xlarge</td>
<td>4 x 900 GB (3.6 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5d.metal</td>
<td>4 x 900 GB (3.6 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5dn.large</td>
<td>1 x 75 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5dn.xlarge</td>
<td>1 x 150 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5dn.2xlarge</td>
<td>1 x 300 GB</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5dn.4xlarge</td>
<td>2 x 300 GB (600 GB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5dn.8xlarge</td>
<td>2 x 600 GB (1.2 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5dn.12xlarge</td>
<td>2 x 900 GB (1.8 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5dn.16xlarge</td>
<td>4 x 600 GB (2.4 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5dn.24xlarge</td>
<td>4 x 900 GB (3.6 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>r5dn.metal</td>
<td>4 x 900 GB (3.6 TB)</td>
<td>NVMe SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>x1.16xlarge</td>
<td>1 x 1,920 GB</td>
<td>SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>x1.32xlarge</td>
<td>2 x 1,920 GB (3.84 TB)</td>
<td>SSD</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>x1e.xlarge</td>
<td>1 x 120 GB</td>
<td>SSD</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>
Instance store volumes

<table>
<thead>
<tr>
<th>Instance type</th>
<th>Instance store volumes</th>
<th>Type</th>
<th>Needs initialization*</th>
<th>TRIM support**</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1e.2xlarge</td>
<td>1 x 240 GB</td>
<td>SSD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1e.4xlarge</td>
<td>1 x 480 GB</td>
<td>SSD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1e.8xlarge</td>
<td>1 x 960 GB</td>
<td>SSD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1e.16xlarge</td>
<td>1 x 1,920 GB</td>
<td>SSD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1e.32xlarge</td>
<td>2 x 1,920 GB (3.84 TB)</td>
<td>SSD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z1d.large</td>
<td>1 x 75 GB</td>
<td>NVMe SSD</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>z1d.xlarge</td>
<td>1 x 150 GB</td>
<td>NVMe SSD</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>z1d.2xlarge</td>
<td>1 x 300 GB</td>
<td>NVMe SSD</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>z1d.3xlarge</td>
<td>1 x 450 GB</td>
<td>NVMe SSD</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>z1d.6xlarge</td>
<td>1 x 900 GB</td>
<td>NVMe SSD</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>z1d.12xlarge</td>
<td>2 x 900 GB (1.8 TB)</td>
<td>NVMe SSD</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>z1d.metal</td>
<td>2 x 900 GB (1.8 TB)</td>
<td>NVMe SSD</td>
<td></td>
<td>✔</td>
</tr>
</tbody>
</table>

* Volumes attached to certain instances suffer a first-write penalty unless initialized.

** For more information, see Instance store volume TRIM support (p. 1404).

To query instance store volume information using the AWS CLI

You can use the describe-instance-types AWS CLI command to display information about an instance type, such as its instance store volumes. The following example displays the total size of instance storage for all R5 instances with instance store volumes.

```
aws ec2 describe-instance-types --filters "Name=instance-type,Values=r5*" "Name=instance-storage-supported,Values=true" --query "InstanceTypes[].InstanceStorageInfo.TotalSizeInGB" --output table
```

```
--------------------------------
<table>
<thead>
<tr>
<th>DescribeInstanceTypes</th>
</tr>
</thead>
<tbody>
<tr>
<td>r5ad.24xlarge</td>
</tr>
<tr>
<td>r5ad.12xlarge</td>
</tr>
<tr>
<td>r5dn.8xlarge</td>
</tr>
<tr>
<td>r5ad.8xlarge</td>
</tr>
<tr>
<td>r5ad.large</td>
</tr>
<tr>
<td>r5d.4xlarge</td>
</tr>
<tr>
<td>. . .</td>
</tr>
<tr>
<td>r5dn.2xlarge</td>
</tr>
<tr>
<td>r5d.12xlarge</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
</tbody>
</table>
```

The following example displays the complete instance storage details for the specified instance type.

```
aws ec2 describe-instance-types --filters "Name=instance-type,Values=r5d.4xlarge" --query "InstanceTypes[].InstanceStorageInfo"
```

The example output shows that this instance type has two 300 GB NVMe SSD volumes, for a total of 600 GB of instance storage.
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 mappings (p. 1413).

A block device mapping always specifies the root volume for the instance. The root volume is mounted automatically. For Windows instances, the root volume must be an Amazon EBS volume; instance store is not supported for the root volume.

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. 1160).

You can specify the instance store volumes for your instance only when you launch it. You can't attach instance store volumes to an instance after you've launched it.

If you change the instance type, an instance store will not be attached to the new instance type. For more information, see Change the instance type (p. 231).

The number and size of available instance store volumes for your instance varies by instance type. Some instance types do not support instance store volumes. If the number of instance store volumes in a block device mapping exceeds the number of instance store volumes available to an instance, the additional volumes are ignored. For more information about the instance store volumes supported by each instance type, see Instance store volumes (p. 1393).

If the instance type you choose for your instance supports non-NVMe instance store volumes, you must add them to the block device mapping for the instance when you launch it. NVMe instance store volumes are available by default. After you launch an instance, you must ensure that the instance store volumes for your instance are formatted and mounted before you can use them. The root volume of an instance store-backed instance is mounted automatically.

Contents

- Add instance store volumes to an AMI (p. 1401)
- Add instance store volumes to an instance (p. 1402)
- Make instance store volumes available on your instance (p. 1402)
Add instance store volumes to an AMI

You can create an AMI with a block device mapping that includes instance store volumes. If you launch an instance with an instance type that supports instance store volumes and an AMI that specifies instance store volumes in its block device mapping, the instance includes these instance store volumes. If the number of instance store volumes in the block device mapping exceeds the number of instance store volumes available to the instance, the additional instance store volumes are ignored.

Considerations

- For M3 instances, specify instance store volumes in the block device mapping of the instance, not the AMI. Amazon EC2 might ignore instance store volumes that are specified only in the block device mapping of the AMI.
- When you launch an instance, you can omit non-NVMe instance store volumes specified in the AMI block device mapping or add instance store volumes.

New console

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 and templates, Create image.
4. On the Create image page, enter a meaningful name and description for your image.
5. For each instance store volume to add, choose Add volume, from Volume type select an instance store volume, and from Device select a device name. (For more information, see Device names on Windows instances (p. 1412).) 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.

Old console

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.
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 names on Windows instances (p. 1412).) 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 Access Amazon EC2 (p. 3).
Add instance store volumes

When you launch an instance, the default block device mapping is provided by the specified AMI. If you need additional instance store volumes, you must add them to the instance as you launch it. You can also omit devices specified in the AMI block device mapping.

Considerations

- For M3 instances, you might receive instance store volumes even if you do not specify them in the block device mapping for the instance.
- For HS1 instances, no matter how many instance store volumes you specify in the block device mapping of an AMI, the block device mapping for an instance launched from the AMI automatically includes the maximum number of supported instance store volumes. You must explicitly remove the instance store volumes that you don't want from the block device mapping for the instance before you launch it.

To update the block device mapping for an instance using the console

1. Open the Amazon EC2 console.
2. From the dashboard, choose Launch instance.
3. In Step 1: Choose an Amazon Machine Image (AMI), select the AMI to use and choose Select.
4. Follow the wizard to complete Step 1: Choose an Amazon Machine Image (AMI), Step 2: Choose an Instance Type, and Step 3: Configure Instance Details.
5. In Step 4: Add Storage, modify the existing entries as needed. For each instance store volume to add, choose Add New Volume, from Volume Type select an instance store volume, and from Device select a device name. The number of available instance store volumes depends on the instance type.
6. Complete the wizard and launch the instance.
7. (Optional) To view the instance store volumes available on your instance, open Windows Disk Management.

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 Access Amazon EC2 (p. 3).

- --block-device-mappings with run-instances (AWS CLI)
- -BlockDeviceMapping with New-EC2Instance (AWS Tools for Windows PowerShell)

Make 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. 1404). 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 View the instance block device mapping for instance store volumes (p. 1422).

For Windows instances, you can also view the instance store volumes using Windows Disk Management. For more information, see List disks using Disk Management (p. 1427).

To manually mount an instance store volume

1. Choose Start, enter Computer Management, and then press Enter.
2. In left-hand panel, choose Disk Management.
3. If you are prompted to initialize the volume, choose the volume to initialize, select the required partition type depending on your use case, and then choose OK.
4. In the list of volumes, right-click the volume to mount, and then choose New Simple Volume.
5. On the wizard, choose Next.
6. On the Specify Volume Size screen, choose Next to use the maximum volume size. Alternatively, choose a volume size that is between the minimum and maximum disk space.
7. On the Assign a Drive Letter or Path screen, do one of the following, and choose Next.
   
   - To mount the volume with a drive letter, choose Assign the following drive letter and then choose the drive letter to use.
   - To mount the volume as a folder, choose Mount in the following empty NTFS folder and then choose Browse to create or select the folder to use.
   - To mount the volume without a drive letter or path, choose Do not assign a drive letter or drive path.
8. On the Format Partition screen, specify whether or not to format the volume. If you choose to format the volume, choose the required file system and unit size, and specify a volume label.

SSD instance store volumes

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. 1400).

NVMe SSD volumes

Some instances offer non-volatile memory express (NVMe) solid state drives (SSD) instance store volumes. For more information about the type of instance store volume supported by each instance type, see Instance store volumes (p. 1393).

The latest AWS Windows AMIs for the following operating systems contain the AWS NVMe drivers used to interact with SSD instance store volumes that are exposed as NVMe block devices for better performance:

- Windows Server 2019
- 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**.

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 (p. 538).

The data on NVMe instance storage is encrypted using an XTS-AES-256 block cipher implemented in a hardware module on the instance. The encryption keys are generated using the hardware module and are unique to each NVMe instance storage device. All encryption keys are destroyed when the instance is stopped or terminated and cannot be recovered. You cannot disable this encryption and you cannot provide your own encryption key.

**Non-NVMe SSD volumes**

The following instances support instance store volumes that use non-NVMe SSDs to deliver high random I/O performance: C3, G2, I2, M3, R3, and X1. For more information about the instance store volumes supported by each instance type, see Instance store volumes (p. 1393).

**Instance store volume TRIM support**

Some instance types support SSD volumes with TRIM. For more information, see Instance store volumes (p. 1393).

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. On Windows, to temporarily disable TRIM support during initial formatting, use the`fsutil behavior set DisableDeleteNotify 1` command. After formatting is complete, re-enable TRIM support by using`fsutil behavior set DisableDeleteNotify 0`.

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 the`fsutil behavior set DisableDeleteNotify 0` command to ensure TRIM support is enabled during normal operation.

**File storage**

Cloud file storage is a method for storing data in the cloud that provides servers and applications access to data through shared file systems. This compatibility makes cloud file storage ideal for workloads that rely on shared file systems and provides simple integration without code changes.

There are many file storage solutions that exist, ranging from a single node file server on a compute instance using block storage as the underpinnings with no scalability or few redundancies to protect the data, to a do-it-yourself clustered solution, to a fully-managed solution. The following content introduces some of the storage services provided by AWS for use with Windows.

**Contents**

- Use Amazon S3 with Amazon EC2 (p. 1405)
- Use Amazon EFS with Amazon EC2 (p. 1406)
- Use Amazon FSx for Windows File Server with Amazon EC2 (p. 1406)
Use Amazon S3 with Amazon EC2

Amazon S3 is a repository for internet data. Amazon S3 provides access to reliable, fast, and inexpensive data storage infrastructure. It is designed to make web-scale computing easier by enabling you to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web. Amazon S3 stores data objects redundantly on multiple devices across multiple facilities and allows concurrent read or write access to these data objects by many separate clients or application threads. You can use the redundant data stored in Amazon S3 to recover quickly and reliably from instance or application failures.

Amazon EC2 uses Amazon S3 for storing Amazon Machine Images (AMIs). You use AMIs for launching EC2 instances. In case of instance failure, you can use the stored AMI to immediately launch another instance, thereby allowing for fast recovery and business continuity.

Amazon EC2 also uses Amazon S3 to store snapshots (backup copies) of the data volumes. You can use snapshots for recovering data quickly and reliably in case of application or system failures. You can also use snapshots as a baseline to create multiple new data volumes, expand the size of an existing data volume, or move data volumes across multiple Availability Zones, thereby making your data usage highly scalable. For more information about using data volumes and snapshots, see Amazon Elastic Block Store (p. 1159).

Objects are the fundamental entities stored in Amazon S3. Every object stored in Amazon S3 is contained in a bucket. Buckets organize the Amazon S3 namespace at the highest level and identify the account responsible for that storage. Amazon S3 buckets are similar to internet domain names. Objects stored in the buckets have a unique key value and are retrieved using a URL. For example, if an object with a key value /photos/mygarden.jpg is stored in the DOC-EXAMPLE-BUCKET1 bucket, then it is addressable using the URL https://DOC-EXAMPLE-BUCKET1.s3.amazonaws.com/photos/mygarden.jpg.

For more information about the features of Amazon S3, see the Amazon S3 product page.

Usage examples

Given the benefits of Amazon S3 for storage, you might decide to use this service to store files and data sets for use with EC2 instances. There are several ways to move data to and from Amazon S3 to your instances. In addition to the examples discussed below, there are a variety of tools that people have written that you can use to access your data in Amazon S3 from your computer or your instance. Some of the common ones are discussed in the AWS forums.

If you have permission, you can copy a file to or from Amazon S3 and your instance using one of the following methods.

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.

```sh
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.

```sh
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.

```sh
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.

**Use Amazon EFS with Amazon EC2**

Amazon EFS provides scalable file storage for use with Amazon EC2. You can use an EFS file system as a common data source for workloads and applications running on multiple instances. For more information, see the Amazon Elastic File System product page.

**Important**

Amazon EFS is not supported on Windows instances.

To use Amazon EFS with a Linux instance, see Amazon Elastic File System (Amazon EFS) in the Amazon EC2 User Guide for Linux Instances.

**Use Amazon FSx for Windows File Server with Amazon EC2**

Amazon FSx for Windows File Server provides fully managed Windows file servers, backed by a fully-native Windows file system with the features, performance, and compatibility to easily lift and shift enterprise applications to AWS.

Amazon FSx supports a broad set of enterprise Windows workloads with fully managed file storage built on Microsoft Windows Server. Amazon FSx has native support for Windows file system features and for the industry-standard Server Message Block (SMB) protocol to access file storage over a network. Amazon FSx is optimized for enterprise applications in the AWS Cloud, with native Windows compatibility, enterprise performance and features, and consistent sub-millisecond latencies.

With file storage on Amazon FSx, the code, applications, and tools that Windows developers and administrators use today can continue to work unchanged. The Windows applications and workloads that are ideal for Amazon FSx include business applications, home directories, web serving, content management, data analytics, software build setups, and media processing workloads.
As a fully managed service, Amazon FSx for Windows File Server eliminates the administrative overhead of setting up and provisioning file servers and storage volumes. Additionally, it keeps Windows software up to date, detects and addresses hardware failures, and performs backups. It also provides rich integration with other AWS services, including AWS Directory Service for Microsoft Active Directory, Amazon WorkSpaces, AWS Key Management Service, and AWS CloudTrail.

For more information, see the Amazon FSx for Windows File Server User Guide. For pricing information, see Amazon FSx for Windows File Server Pricing.

Instance volume limits

The maximum number of volumes that your instance can have depends on the operating system and instance type. When considering how many volumes to add to your instance, you should consider whether you need increased I/O bandwidth or increased storage capacity.

Contents

- Nitro System volume limits (p. 1407)
- Windows-specific volume limits (p. 1407)
- Bandwidth versus capacity (p. 1408)

Nitro System volume limits

Instances built on the Nitro System (p. 146) support a maximum number of attachments, which are shared between network interfaces, EBS volumes, and NVMe instance store volumes. Every instance has at least one network interface attachment. NVMe instance store volumes are automatically attached. For more information, see Elastic network interfaces (p. 934) and Instance store volumes (p. 1393).

Most of these instances support a maximum of 28 attachments. For example, if you have no additional network interface attachments on an EBS-only instance, you can attach up to 27 EBS volumes to it. If you have one additional network interface on an instance with 2 NVMe instance store volumes, you can attach 24 EBS volumes to it.

For other instances, the following limits apply:

- d3.8xlarge and d3en.12xlarge instances support a maximum of 3 EBS volumes.
- Most bare metal instances support a maximum of 31 EBS volumes.
- High memory virtualized instances support a maximum of 27 EBS volumes.
- High memory bare metal instances support a maximum of 19 EBS volumes.

If you launched a u-6tb1.metal, u-9tb1.metal, or u-12tb1.metal high memory bare metal instance before March 12, 2020, it supports a maximum of 14 EBS volumes. To attach up to 19 EBS volumes to these instances, contact your account team to upgrade the instance at no additional cost.

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.
<table>
<thead>
<tr>
<th>Driver</th>
<th>Volume Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS PV</td>
<td>26</td>
</tr>
<tr>
<td>Citrix PV</td>
<td>26</td>
</tr>
<tr>
<td>Red Hat PV</td>
<td>17</td>
</tr>
</tbody>
</table>

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 Upgrade PV drivers on Windows instances (p. 524).

For more information about how device names related to volumes, see Map disks to volumes on your Windows instance (p. 1423).

Bandwidth versus capacity

For consistent and predictable bandwidth use cases, use EBS-optimized or 10 Gigabit network connectivity instances and General Purpose SSD or Provisioned IOPS SSD volumes. Follow the guidance in Amazon EBS–optimized instances (p. 1344) 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.

Amazon EC2 instance 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.

Topics
- Configure the root volume to persist (p. 1408)
- Confirm that a root volume is configured to persist (p. 1410)
- Change the initial size of the root volume (p. 1411)

Configure the root volume to persist

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 an existing instance, you must use the command line.

Topics
- Configure the root volume to persist during instance launch (p. 1408)
- Configure the root volume to persist for an existing instance (p. 1409)

Configure the root volume to persist during instance launch

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

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

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

AWS CLI

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

Use the run-instances command and include a block device mapping that sets the DeleteOnTermination attribute to false.

```
C:\> aws ec2 run-instances --block-device-mappings file://mapping.json ...
parameters...
```

Specify the following in mapping.json.

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

Tools for Windows PowerShell

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

Use the New-EC2Instance command and include a block device mapping that sets the DeleteOnTermination attribute to false.

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

Configure the root volume to persist for an existing instance

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

AWS CLI

To configure the root volume to persist for an existing instance using the AWS CLI
Confirm that a root volume is configured to persist

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

New console

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

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

Old console

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

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

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

Use the describe-instances command and verify that the DeleteOnTermination attribute in the BlockDeviceMappings response element is set to false.

C:\> aws ec2 describe-instances --instance-id i-1234567890abcdef0

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

Tools for Windows PowerShell

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

Use the Get-EC2Instance and verify that the DeleteOnTermination attribute in the BlockDeviceMappings response element is set to false.

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

Change the initial size of the root volume

By default, the size of the root volume is determined by the size of the snapshot. You can increase the initial size of the root volume using the block device mapping of the instance as follows.

1. Determine the device name of the root volume specified in the AMI, as described in View the EBS volumes in an AMI block device mapping (p. 1418).
2. Confirm the size of the snapshot specified in the AMI block device mapping, as described in View Amazon EBS snapshot information (p. 1233).
3. Override the size of the root volume using the instance block device mapping, as described in Update the block device mapping when launching an instance (p. 1419), specifying a volume size that is larger than the snapshot size.

For example, the following entry for the instance block device mapping increases the size of the root volume, /dev/xvda, to 100 GiB. You can omit the snapshot ID in the instance block device mapping because the snapshot ID is already specified in the AMI block device mapping.

```json
{
  "DeviceName": "/dev/xvda",
  "Ebs": {
    "VolumeSize": 100
  }
}
```
Device names 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.

The number of volumes that your instance can support is determined by the operating system. For more information, see Instance volume limits (p. 1407).

Contents
• Available device names (p. 1412)
• Device name considerations (p. 1413)

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

Windows AMIs use one of the following sets of drivers to permit access to virtualized hardware: AWS PV, Citrix PV, and RedHat PV. For more information, see Paravirtual drivers for Windows instances (p. 519).

The following table lists the available device names that you can specify in a block device mapping or when attaching an EBS volume.

<table>
<thead>
<tr>
<th>Driver type</th>
<th>Available</th>
<th>Reserved for root</th>
<th>Recommended for EBS volumes</th>
<th>Instance store volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS PV, Citrix PV</td>
<td>xvd[b-z]</td>
<td>/dev/sda1</td>
<td>xvd[f-z] *</td>
<td>xvdc[a-x]</td>
</tr>
<tr>
<td></td>
<td>xvd[b-c][a-z]</td>
<td></td>
<td>xvd[a-e]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/dev/sda1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>/dev/sd[b-e]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red Hat PV</td>
<td>xvd[a-z]</td>
<td>/dev/sda1</td>
<td>xvd[f-p]</td>
<td>xvdc[a-x]</td>
</tr>
<tr>
<td></td>
<td>xvd[b-c][a-z]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>/dev/sda1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>/dev/sd[b-e]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* For Citrix PV and Red Hat PV, if you map an EBS volume with the name xvd[a], Windows does not recognize the volume (the volume is visible for AWS PV or AWS NVMe).

** 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. 1392). For more information about NVMe EBS volumes (Nitro-based instances), including how to identify the EBS device, see Amazon EBS and NVMe on Windows instances (p. 1343).
Device name considerations

Keep the following in mind when selecting a device name:

- Although you can attach your EBS volumes using the device names used to attach instance store volumes, we strongly recommend that you don't because the behavior can be unpredictable.
- The number of NVMe instance store volumes for an instance depends on the size of the instance. NVMe instance store volumes are automatically enumerated and assigned a Windows drive letter.
- 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 and later). After the devices have been mapped to drives, they are initialized and mounted. The root drive is initialized and mounted as C: \. 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 instance store volumes, the default depends on the driver. AWS PV drivers and 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:. For more information, see Configure a Windows instance using the EC2Config service (p. 493), Configure a Windows instance using EC2Launch (p. 486), and Map disks to volumes on your Windows instance (p. 1423).

Block device mappings

Each instance that you launch has an associated root device volume, which is either an Amazon EBS volume or an instance store volume. You can use block device mapping to specify additional EBS volumes or instance store volumes to attach to an instance when it's launched. You can also attach additional EBS volumes to a running instance; see Attach an Amazon EBS volume to an instance (p. 1186). However, the only way to attach instance store volumes to an instance is to use block device mapping to attach the volumes as the instance is launched.

For more information about root device volumes, see Amazon EC2 instance root device volume (p. 1408).

Contents

- Block device mapping concepts (p. 1413)
- AMI block device mapping (p. 1417)
- Instance block device mapping (p. 1419)

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. 1414)
• Block device mapping instance store caveats (p. 1414)
• Example block device mapping (p. 1415)
• How devices are made available in the operating system (p. 1416)

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 names on Windows instances (p. 1412).

For Instance store volumes, you also specify the following information:

• The virtual device: ephemeral[0-23]. Note that the number and size of available instance store volumes for your instance varies by instance type.

For NVMe instance store volumes, the following information also applies:

• These volumes are automatically enumerated and assigned a device name; including them in your block device mapping has no effect.

For EBS volumes, you also specify the following information:

• The ID of the snapshot to use to create the block device (snap-xxxxxxxx). This value is optional as long as you specify a volume size.
• The size of the volume, in GiB. The specified size must be greater than or equal to the size of the specified snapshot.
• Whether to delete the volume on instance termination (true or false). The default value is true for the root device volume and false for attached volumes. When you create an AMI, its block device mapping inherits this setting from the instance. When you launch an instance, it inherits this setting from the AMI.
• The volume type, which can be gp2 and gp3 for General Purpose SSD, io1 and io2 for Provisioned IOPS SSD, st1 for Throughput Optimized HDD, sc1 for Cold HDD, or standard for Magnetic. The default value is gp2.
• The number of input/output operations per second (IOPS) that the volume supports. (Used only with io1 and io2 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 Specify a block device mapping for an AMI (p. 1417) and Update the block device mapping when launching an instance (p. 1419).

How devices are made available in the operating system

Device names like \dev\sda1 and xvdh are used by Amazon EC2 to describe block devices. The block device mapping is used by Amazon EC2 to specify the block devices to attach to an EC2 instance. After a block device is attached to an instance, it must be mounted by the operating system before you can access the storage device. When a block device is detached from an instance, it is unmounted by the operating system and you can no longer access the storage device.

With a 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 Configure a Windows instance using the EC2Config service (p. 493).
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
- Specify a block device mapping for an AMI (p. 1417)
- View the EBS volumes in an AMI block device mapping (p. 1418)

Specify 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 and templates, Create image.
4. Enter a name and a description for the image.
5. The instance volumes appear under Instance volumes. To add another volume, choose Add volume.
6. For Volume type, choose the volume type. For Device choose the device name. For an EBS volume, you can specify additional details, such as a snapshot, volume size, volume type, IOPS, and encryption state.
7. Choose Create image.

To add volumes to an AMI using the command line
Use the create-image AWS CLI command to specify a block device mapping for an EBS-backed AMI. Use the register-image AWS CLI command to specify a block device mapping for an instance store-backed AMI.

Specify the block device mapping using the --block-device-mappings parameter. Arguments encoded in JSON can be supplied either directly on the command line or by reference to a file:

```bash
--block-device-mappings [mapping, ...]
--block-device-mappings [file://mapping.json]
```
To add an instance store volume, use the following mapping.

```
{
    "DeviceName": "xvdb",
    "VirtualName": "ephemeral0"
}
```

To add an empty 100 GiB gp2 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`

### View the EBS volumes in an AMI block device mapping

You can easily enumerate the EBS volumes in the block device mapping for an AMI.

**To view the EBS volumes for an AMI using the console**

1. Open the Amazon EC2 console.
2. In the navigation pane, choose AMIs.
3. Choose **EBS images** from the **Filter** list to get a list of EBS-backed AMIs.
4. Select the desired AMI, and look at the **Details** tab. At a minimum, the following information is available for the root device:
   - **Root Device Type** (ebs)
   - **Root Device Name** (for example, /dev/sda1)
   - **Block Devices** (for example, /dev/sda1=snap-1234567890abcdef0:8:true)
If the AMI was created with additional EBS volumes using a block device mapping, the **Block Devices** field displays the mapping for those additional volumes as well. (This screen doesn't display instance store volumes.)

**To view the EBS volumes for an AMI using the command line**

Use the `describe-images` (AWS CLI) command or `Get-EC2Image` (AWS Tools for Windows PowerShell) command to enumerate the EBS volumes in the block device mapping for an AMI.

**Instance block device mapping**

By default, an instance that you launch includes any storage devices specified in the block device mapping of the AMI from which you launched the instance. You can specify changes to the block device mapping for an instance when you launch it, and these updates overwrite or merge with the block device mapping of the AMI.

**Limitations**

- For the root volume, you can only modify the following: volume size, volume type, and the **Delete on Termination** flag.
- When you modify an EBS volume, you can't decrease its size. Therefore, you must specify a snapshot whose size is equal to or greater than the size of the snapshot specified in the block device mapping of the AMI.

**Contents**

- Update the block device mapping when launching an instance (p. 1419)
- Update the block device mapping of a running instance (p. 1421)
- View the EBS volumes in an instance block device mapping (p. 1421)
- View the instance block device mapping for instance store volumes (p. 1422)

**Update the block device mapping when launching an instance**

You can add EBS volumes and instance store volumes to an instance when you launch it. Note that updating the block device mapping for an instance doesn't make a permanent change to the block device mapping of the AMI from which it was launched.

**To add volumes to an instance using the console**

1. Open the Amazon EC2 console.
2. From the dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, select the AMI to use and choose **Select**.
4. Follow the wizard to complete the **Choose an Instance Type** and **Configure Instance Details** pages.
5. On the **Add Storage** page, you can modify the root volume, EBS volumes, and instance store volumes as follows:
   - To change the size of the root volume, locate the **Root** volume under the **Type** column, and change its **Size** field.
   - To suppress an EBS volume specified by the block device mapping of the AMI used to launch the instance, locate the volume and click its **Delete** icon.
   - To add an EBS volume, choose **Add New Volume**, choose **EBS** from the **Type** list, and fill in the fields (**Device**, **Snapshot**, and so on).
• To suppress an instance store volume specified by the block device mapping of the AMI used to launch the instance, locate the volume, and choose its **Delete** icon.

• To add an instance store volume, choose **Add New Volume**, select **Instance Store** from the **Type** list, and select a device name from **Device**.

6. Complete the remaining wizard pages, and choose **Launch**.

**To add volumes to an instance using the AWS CLI**

Use the `run-instances` AWS CLI command with the `--block-device-mappings` option to specify a block device mapping for an instance at launch.

For example, suppose that an EBS-backed AMI specifies the following block device mapping:

- xvdb=ephemeral0
- xvdh=snap-1234567890abcdef0
- xvdj=:100

To prevent `xvdj` from attaching to an instance launched from this AMI, use the following mapping.

```json
{
  "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.

```json
{
  "DeviceName": "xvdh",
  "Ebs": {
    "VolumeSize": 300
  }
}
```

To increase the size of the root volume at instance launch, first call `describe-images` with the ID of the AMI to verify the device name of the root volume. For example, "RootDeviceName": "/dev/xvda". To override the size of the root volume, specify the device name of the root device used by the AMI and the new volume size.

```json
{
  "DeviceName": "/dev/xvda",
  "Ebs": {
    "VolumeSize": 100
  }
}
```

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. If the instance supports NVMe instance store volumes, they are automatically enumerated and assigned an NVMe device name.

```json
{
  "DeviceName": "xvdc",
  "VirtualName": "ephemeral1"
}
```

To add volumes to an instance using the AWS Tools for Windows PowerShell
Use the \texttt{-BlockDeviceMapping} parameter with the \texttt{New-EC2Instance} command (AWS Tools for Windows PowerShell).

**Update the block device mapping of a running instance**

You can use the \texttt{modify-instance-attribute} AWS CLI command to update the block device mapping of a running instance. You do not need to stop the instance before changing this attribute.

\begin{verbatim}
aws ec2 modify-instance-attribute --instance-id i-1a2b3c4d --block-device-mappings file://mapping.json
\end{verbatim}

For example, to preserve the root volume at instance termination, specify the following in \texttt{mapping.json}.

\begin{verbatim}
[
  {
    "DeviceName": "/dev/sda1",
    "Ebs": {
      "DeleteOnTermination": false
    }
  }
]
\end{verbatim}

Alternatively, you can use the \texttt{-BlockDeviceMapping} parameter with the \texttt{Edit-EC2InstanceAttribute} command (AWS Tools for Windows PowerShell).

**View the EBS volumes in an instance block device mapping**

You can easily enumerate the EBS volumes mapped to an instance.

\textbf{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 \textbf{Instances}.
3. In the search box, enter \textbf{Root device type}, and then choose \textbf{EBS}. This displays a list of EBS-backed instances.
4. Select the desired instance and look at the details displayed in the \textbf{Storage} tab. At a minimum, the following information is available for the root device:
   - \textbf{Root device type} (for example, \texttt{EBS})
   - \textbf{Root device name} (for example, \texttt{/dev/xvda})
   - \textbf{Block devices} (for example, \texttt{/dev/xvda}, \texttt{xvdf}, and \texttt{xvdj})

   If the instance was launched with additional EBS volumes using a block device mapping, they appear under \textbf{Block devices}. Any instance store volumes do not appear on this tab.
5. To display additional information about an EBS volume, choose its volume ID to go to the volume page. For more information, see \textit{View information about an Amazon EBS volume (p. 1191)}.

**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.

**View 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. The method you use to view the instance store volumes for your instance depends on the volume type.

**NVMe instance store volumes**

You can use the NVMe command line package, `nvme-cli`, to query the NVMe instance store volumes in the block device mapping. Download and install the package on your instance, and then run the following command.

```
[ec2-user ~]$ sudo nvme list
```

The following is example output for an instance. The text in the Model column indicates whether the volume is an EBS volume or an instance store volume. In this example, both `/dev/nvme1n1` and `/dev/nvme2n1` are instance store volumes.

<table>
<thead>
<tr>
<th>Node</th>
<th>SN</th>
<th>Model</th>
<th>Namespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/nvme0n1</td>
<td>vol06ac3f1b715b759a</td>
<td>Amazon Elastic Block Store</td>
<td>1</td>
</tr>
<tr>
<td>/dev/nvme1n1</td>
<td>AWS2C1436F5159EB661</td>
<td>Amazon EC2 NVMe Instance Storage</td>
<td>1</td>
</tr>
<tr>
<td>/dev/nvme2n1</td>
<td>AWSB1P4FF0C0A6C281E</td>
<td>Amazon EC2 NVMe Instance Storage</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**HDD or SSD instance store volumes**

You can use instance metadata to query the HDD or SSD instance store volumes in the block device mapping. NVMe instance store volumes are not included.

The base URI for all requests for instance metadata is `http://169.254.169.254/latest/`. For more information, see Instance metadata and user data (p. 579).

First, connect to your running instance. From the instance, use this query to get its 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.
Map disks to volumes

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. 1407).

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 Make an Amazon EBS volume available for use on Windows (p. 1187).

When you add a volume to your instance, you specify the device name that Amazon EC2 uses. For more information, see Device names on Windows instances (p. 1412). 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 Upgrade PV drivers on Windows instances (p. 524).

Contents

- List NVMe volumes (p. 1424)
  - List NVMe disks using Disk Management (p. 1424)
  - List NVMe disks using PowerShell (p. 1425)
  - Map NVMe EBS volumes (p. 1426)
- List volumes (p. 1427)
  - List disks using Disk Management (p. 1427)
  - Map disk devices to device names (p. 1429)
List NVMe volumes

You can find the disks on your Windows instance using Disk Management or Powershell.

List NVMe disks using Disk Management

You can find the disks on your Windows instance using Disk Management.

To find the disks on your Windows instance

1. Log in to your Windows instance using Remote Desktop. For more information, see Connect to your Windows instance (p. 413).
2. Start the Disk Management utility.
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 r5d.4xlarge instance with two additional EBS volumes.
List NVMe disks using PowerShell

The following PowerShell script lists each disk and its corresponding device name and volume. It is intended for use with instances build on the Nitro System (p. 146), which use NVMe EBS and instance store volumes.

Connect to your Windows instance and run the following command to enable PowerShell script execution.

```
Set-ExecutionPolicy RemoteSigned
```

Copy the following script and save it as `mapping.ps1` on your Windows instance.

```powershell
# List the disks for NVMe volumes
function Get-EC2InstanceMetadata {
    param([string]$Path)
    (Invoke-WebRequest -Uri "http://169.254.169.254/latest/$Path").Content
}

function GetEBSVolumeId {
    param($Path)
    $SerialNumber = (Get-Disk -Path $Path).SerialNumber
    if($SerialNumber -like 'vol*'){
        $EbsVolumeId = $SerialNumber.Substring(0,20).Replace("vol","vol-")
    } else {
        $EbsVolumeId = $SerialNumber.Substring(0,20).Replace("AWS","AWS-")
    }
    return $EbsVolumeId
}

function GetDeviceName{
    param($EbsVolumeId)
    if($EbsVolumeId -like 'vol*'){
        $VolumeName = ""
    } else {
        $Device = "Ephemeral"
        $VolumeName = "Temporary Storage"
    }
    return $Device,$VolumeName
}

function GetDriveLetter{
    param($Path)
    $DiskNumber =  (Get-Disk -Path $Path).Number
    if($DiskNumber -eq 0){
        $VirtualDevice = "root"
        $DriveLetter = "C"
        $PartitionNumber = (Get-Partition -DriveLetter C).PartitionNumber
    } else {
        $VirtualDevice = "N/A"
        $DriveLetter = (Get-Partition -DiskNumber $DiskNumber).DriveLetter
        if(!$DriveLetter)
            {
                $DriveLetter = ((Get-Partition -DiskId $Path).AccessPaths).Split(",")[0]
            }
```
# List NVMe volumes

```powershell
$PartitionNumber = (Get-Partition -DiskId $Path).PartitionNumber

return $DriveLetter,$VirtualDevice,$PartitionNumber

$Report = @()
foreach($Path in (Get-Disk).Path)
{
    $Disk_ID = (Get-Partition -DiskId $Path).DiskId
    $Disk = (Get-Disk -Path $Path).Number
    $EbsVolumeId  = GetEBSVolumeId($Path)
    $Size = (Get-Disk -Path $Path).Size
    $DriveLetter,$VirtualDevice, $Partition = (GetDriveLetter($Path))
    $Device,$VolumeName = GetDeviceName($EbsVolumeId)
    $Disk = New-Object PSObject -Property @{
        Disk          = $Disk
        Partitions    = $Partition
        DriveLetter   = $DriveLetter
        EbsVolumeId   = $EbsVolumeId
        Device        = $Device
        VirtualDevice = $VirtualDevice
        VolumeName= $VolumeName
    }
    $Report += $Disk
}

$Report | Sort-Object Disk | Format-Table -AutoSize -Property Disk, Partitions, DriveLetter, EbsVolumeId, Device, VirtualDevice, VolumeName
```

Run the script as follows:

```
PS C:\> .\mapping.ps1
```

The following is example output for an instance with a root volume, two EBS volumes, and two instance store volumes.

```
Disk Partitions DriveLetter EbsVolumeId Device VirtualDevice VolumeName
---- ---------- ----------- -----------           ------    ------------- ----------
0          1 C           vol-03683f1d861744bc7 /dev/sda1 root
1          1 D           vol-082b07051043174b9 xvdb      N/A
2          1 E           vol-0a4b046d39e5f534a2 xvdc      N/A
3          1 F           AWS-6AAD8C2AEE1193F0 Ephemeral N/A           Temporary Storage
4          1 G           AWS-13B7299C2BD031A28 Ephemeral N/A           Temporary Storage
```

If you did not provide your credentials on the Windows instance, the script cannot get the EBS volume ID and uses N/A in the EbsVolumeId column.

## Map NVMe EBS volumes

With instances built on the Nitro System (p. 146), 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 `Identify the EBS device (p. 1343)`.

```
PS C:\> Get-Disk
Number Friendly Name Serial Number                   HealthStatus
OperationalStatus Total Size Partition
               Style
```
You can also run the `ebsnvme-id` command to map NVMe disk numbers to EBS volume IDs and device names.

```
PS C:\> C:\PROGRAMDATA\Amazon\Tools\ebsnvme-id.exe
Disk Number: 0
Volume ID: vol-03683f1d861f744bc7
Device Name: sda1

Disk Number: 1
Volume ID: vol-082b07051043174b9
Device Name: xvdb

Disk Number: 2
Volume ID: vol-0a4064b39e5f534a2
Device Name: xvdc
```

## List volumes

You can find the disks on your Windows instance using Disk Management or Powershell.

### List disks using Disk Management

You can find the disks on your Windows instance using Disk Management.

**To find the disks on your Windows instance**

1. Log in to your Windows instance using Remote Desktop. For more information, see Connect to your Windows instance (p. 413).
2. Start the Disk Management utility.
   
   On Windows Server 2012 and later, 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 [Map disk devices to device names (p. 1429)](#). 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 `xvdj`. 
Map disk devices to device names

The block device driver for the instance assigns the actual volume names when mounting volumes.

Mappings

- Instance store volumes (p. 1429)
- EBS volumes (p. 1430)

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. 1393).

<table>
<thead>
<tr>
<th>Location</th>
<th>Device name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus Number 0, Target ID 78, LUN 0</td>
<td>xvdca</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 79, LUN 0</td>
<td>xvdcb</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 80, LUN 0</td>
<td>xvdcc</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 81, LUN 0</td>
<td>xvdcd</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 82, LUN 0</td>
<td>xvdce</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 83, LUN 0</td>
<td>xvdcf</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 84, LUN 0</td>
<td>xvdcg</td>
</tr>
</tbody>
</table>
## EBS volumes

The following table describes how the Citrix PV and AWS PV drivers map non-NVME EBS volumes to Windows volumes.

<table>
<thead>
<tr>
<th>Location</th>
<th>Device name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus Number 0, Target ID 0, LUN 0</td>
<td>/dev/sda1</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 1, LUN 0</td>
<td>xvdb</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 2, LUN 0</td>
<td>xvdcb</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 3, LUN 0</td>
<td>xvdd</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 4, LUN 0</td>
<td>xvde</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 5, LUN 0</td>
<td>xvdf</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 6, LUN 0</td>
<td>xvdg</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 7, LUN 0</td>
<td>xvdh</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 8, LUN 0</td>
<td>xvd</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 9, LUN 0</td>
<td>xvjd</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 10, LUN 0</td>
<td>xvdk</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 11, LUN 0</td>
<td>xvd</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 12, LUN 0</td>
<td>xvdm</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 13, LUN 0</td>
<td>xvdn</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 14, LUN 0</td>
<td>xvd</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 15, LUN 0</td>
<td>xvdp</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 16, LUN 0</td>
<td>xvdq</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 17, LUN 0</td>
<td>xvd</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 18, LUN 0</td>
<td>xvds</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 19, LUN 0</td>
<td>xvd</td>
</tr>
<tr>
<td>Bus Number 0, Target ID 20, LUN 0</td>
<td>xvdu</td>
</tr>
</tbody>
</table>
List disks using PowerShell

The following PowerShell script lists each disk and its corresponding device name and volume.

Requirements and limitations

- Requires Windows Server 2012 or later.
- Requires credentials to get the EBS volume ID. You can configure a profile using the Tools for PowerShell, or attach an IAM role to the instance.
- Does not support NVMe volumes.
- Does not support dynamic disks.

Connect to your Windows instance and run the following command to enable PowerShell script execution.

```
Set-ExecutionPolicy RemoteSigned
```

Copy the following script and save it as `mapping.ps1` on your Windows instance.

```
# List the disks

function Get-EC2InstanceMetadata {
    param([string]$Path)
    (Invoke-WebRequest -Uri "http://169.254.169.254/latest/$Path").Content
}

function Convert-SCSITargetIdToDeviceName {
    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 {
    $DriveLetter = $null
    $VolumeName = $null
    $DiskDrive = $_
    $Disk = $_.Number
    $Partitions = $_.NumberOfPartitions
    $EbsVolumeID = $_.SerialNumber -replace "_\[^ \]*" -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)
        $BlockDevice = $null
        $VirtualDevice = If ($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 { $EbsVolumeID };
        Device = If ($Device -eq $null) { "N/A" } Else { $Device };
        VirtualDevice = If ($VirtualDevice -eq $null) { "N/A" } Else { $VirtualDevice };
        VolumeName = If ($VolumeName -eq $null) { "N/A" } Else { $VolumeName };
    }
Run the script as follows:

```
PS C:\> .\mapping.ps1
```

The following is example output.

<table>
<thead>
<tr>
<th>Disk</th>
<th>Partitions</th>
<th>DriveLetter</th>
<th>EbsVolumeId</th>
<th>Device</th>
<th>VirtualDevice</th>
<th>VolumeName</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>Z</td>
<td>N/A</td>
<td>xvdca</td>
<td>ephemeral0</td>
<td>N/A</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Y</td>
<td>N/A</td>
<td>xvdcb</td>
<td>ephemeral1</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>C</td>
<td>vol-0064aexamplec838a /dev/sda1</td>
<td>root</td>
<td>Windows</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>D</td>
<td>vol-02256example8a4a3 xvdf</td>
<td>ebs2</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

If you did not provide your credentials on the Windows instance, the script cannot get the EBS volume ID and uses N/A in the EbsVolumeId column.

**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](https://docs.microsoft.com/en-us/windows-server/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. 1392) 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. 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. 443) on these instances. You can also configure IAM policies to allow users to only restart nodes from the AWS Management 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. 1435)
- Step 2: Install and Configure Instance Prerequisites (p. 1437)
- Step 3: Create Failover Cluster (p. 1438)
- Step 4: Enable S2D (p. 1439)
- Step 5: Provision Storage (p. 1439)
Step 1: Launch and Domain Join Instances

All Nitro instances support Storage Spaces Direct using EBS and/or NVMe. All current generation Xen-based instances support Storage Spaces Direct with installation of AWS PV driver 8.2.3 and later. 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
Amazon Elastic Compute Cloud  
User Guide for Windows Instances  
Step 1: Launch and Domain Join Instances

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-c49c0d4c -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-xxxxxxxx, subnet-xxxxxxxx
```

We use the following S2D-node1 network interface configuration:

<table>
<thead>
<tr>
<th>Device</th>
<th>Network Interface</th>
<th>Subnet</th>
<th>Primary IP</th>
<th>Secondary IP addresses</th>
<th>IPv6 IPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0</td>
<td>New network interface</td>
<td>subnet-9850a3ef</td>
<td>172.16.1.199</td>
<td>172.16.1.200</td>
<td>Add IP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>172.16.1.201</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Add IP</td>
<td></td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Server NetBIOS Name</th>
<th>IP Address</th>
<th>Subnets</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2D-Node1</td>
<td>172.16.1.199 (Primary)</td>
<td>AZ1 (e.g., eu-west-1a) – private subnet 1</td>
</tr>
<tr>
<td></td>
<td>172.16.1.200 (secondary which will used for the cluster VIP)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>172.16.1.201 (secondary which will be used later for a role such as SQL FCI)</td>
<td></td>
</tr>
<tr>
<td>S2D-Node2</td>
<td>172.16.3.199 (Primary)</td>
<td>AZ1 (e.g., eu-west-1a) – private subnet 2</td>
</tr>
</tbody>
</table>
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.

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 AWS Systems Manager 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.

---

<table>
<thead>
<tr>
<th>Server NetBIOS Name</th>
<th>IP Address</th>
<th>Subnets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>172.16.3.200 (secondary which will be used for the cluster VIP)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>172.16.3.201 (secondary which will be used later for a role such as SQL FCI)</td>
<td></td>
</tr>
<tr>
<td>ADM01</td>
<td>Not specified</td>
<td>AZ1 (e.g., eu-west-1a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– public subnet</td>
</tr>
</tbody>
</table>

---

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 AWS Systems Manager 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.

  ```powershell
  $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

1. Enable multichannel and set the RSS Connection Count.

  ```powershell
  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 = @(
      ```
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.

```powershell
$report = Test-Cluster -Node $nodes -Include 'Storage Spaces Direct', 'Inventory', 'Network', 'System Configuration'
```

2. Review the test results.

```powershell
$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 Management Console to each respective node.

```powershell
$vips = "172.16.1.200", "172.16.3.200"
New-Cluster -Name S2D -Node $nodes -StaticAddress $vips -NoStorage
```

4. Configure a file share witness.

```powershell
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.

```powershell
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:

```powershell
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.”

After 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.

```powershell
$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.

```powershell
$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.

```
PSC:\> Get-StorageSubSystem *cluster* | Get-StorageHealthReport
```
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. 1442)
- Resource IDs (p. 1443)
- List and filter your resources (p. 1444)
- Tag your Amazon EC2 resources (p. 1450)
- Amazon EC2 service quotas (p. 1463)
- Amazon EC2 usage reports (p. 1464)

Resource locations

Some resources can be used in all AWS Regions (global), and some resources are specific to the AWS Region or Availability Zone in which they reside.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS account</td>
<td>Global</td>
<td>You can use the same AWS account in all Regions.</td>
</tr>
<tr>
<td>Key pairs</td>
<td>Global or Regional</td>
<td>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. For more information, see Amazon EC2 key pairs and Windows instances (p. 1127).</td>
</tr>
<tr>
<td>Amazon EC2 resource identifiers</td>
<td>Regional</td>
<td>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.</td>
</tr>
<tr>
<td>User-supplied resource names</td>
<td>Regional</td>
<td>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.</td>
</tr>
<tr>
<td>AMIs</td>
<td>Regional</td>
<td>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 Copy an AMI (p. 113).</td>
</tr>
<tr>
<td>Resource</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Elastic IP addresses</td>
<td>Regional</td>
<td>An Elastic IP address is tied to a Region and can be associated only with an instance in the same Region.</td>
</tr>
<tr>
<td>Security groups</td>
<td>Regional</td>
<td>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.</td>
</tr>
<tr>
<td>EBS snapshots</td>
<td>Regional</td>
<td>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 Copy an Amazon EBS snapshot (p. 1229).</td>
</tr>
<tr>
<td>EBS volumes</td>
<td>Availability Zone</td>
<td>An Amazon EBS volume is tied to its Availability Zone and can be attached only to instances in the same Availability Zone.</td>
</tr>
<tr>
<td>Instances</td>
<td>Availability Zone</td>
<td>An instance is tied to the Availability Zones in which you launched it. However, its instance ID is tied to the Region.</td>
</tr>
</tbody>
</table>

**Resource IDs**

When resources are created, we assign each resource a unique resource ID. A resource ID takes the form of a resource identifier (such as `snap` for a snapshot) followed by a hyphen and a unique combination of letters and numbers.

You can use resource IDs to find your resources in the Amazon EC2 console. If you are using a command line tool or the Amazon EC2 API to work with Amazon EC2, resource IDs are required for certain commands. For example, if you are using the `stop-instances` AWS CLI command to stop an instance, you must specify the instance ID in the command.

**Resource ID length**

Prior to January 2016, the IDs assigned to newly created resources of certain resource types used 8 characters after the hyphen (for example, `i-1a2b3c4d`). From January 2016 to June 2018, we changed the IDs of these resource types to use 17 characters after the hyphen (for example, `i-1234567890abcdef0`). Depending on when your account was created, you might have resources of the following resource types with short IDs, though any new resources of these types receive the longer IDs:

- bundle
- conversion-task
- customer-gateway
- dhcp-options
- elastic-ip-allocation
- elastic-ip-association
- export-task
- flow-log
- image
List and filter your resources

You can get a list of some types of resources using the Amazon EC2 console. You can get a list of each type of resource using its corresponding command or API action. If you have many resources, you can filter the results to include, or exclude, only the resources that match certain criteria.

Contents
- List and filter resources using the console (p. 1444)
- List and filter using the CLI and API (p. 1448)

List and filter resources using the console

Contents
- List resources using the console (p. 1444)
- Filter resources using the console (p. 1445)

List resources using the console

You can view the most common Amazon EC2 resource types using the console. To view additional resources, use the command line interface or the API actions.

To list EC2 resources using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose the option that corresponds to the resource type. For example, to list your instances, choose **Instances**.

The page displays all resources of the selected resource type.

**Filter resources using the console**

**To filter a list of resources**

1. In the navigation pane, select a resource type (for example, **Instances**).
2. Choose the search field.
3. Choose the filter from the list.
4. Choose a filter value.
5. When you are finished, remove the filter.

The search and filter functionality differs slightly between the *old* and *new* Amazon EC2 console.

**New console**

The new console supports two types of filtering.

- **API filtering** happens on the server side. The filtering is applied on the API call, which reduces the number of resources returned by the server. It allows for quick filtering across large sets of resources, and it can reduce data transfer time and cost between the server and the browser.
- **Client filtering** happens on the client side. It enables you to filter down on data that is already available in the browser (in other words, data that has already been returned by the API). Client filtering works well in conjunction with an API filter to filter down to smaller data sets in the browser.

The new Amazon EC2 console supports the following types of searches:

**Search by keyword**

Searching by keyword is a free text search that lets you search for a value across all of your resources’ attributes, without specifying an attribute to search.

**Note**

All keyword searches use **client filtering**.

To search by keyword, enter or paste what you’re looking for in the search field, and then choose **Enter**. For example, searching for `123` matches all instances that have `123` in any of their attributes, such as an IP address, instance ID, VPC ID, or AMI ID. If your free text search returns unexpected matches, apply additional filters.

**Search by attributes**

Searching by an attribute lets you search a specific attribute across all of your resources.

**Note**

Attribute searches use either **API filtering** or **client filtering**, depending on the selected attribute. When performing an attribute search, the attributes are grouped accordingly.

For example, you can search the **Instance state** attribute for all of your instances to return only instances that are in the **stopped** state. To do this:

1. In the search field on the **Instances** screen, start entering **Instance state**. As you enter the characters, the two types of filters appear for **Instance state: API filters** and **Client filters**.
2. To search on the server side, choose **Instance state** under **API filters**. To search on the client side, choose **Instance state (client)** under **Client filters**.
A list of possible values for the selected attribute appears.
3. Choose **stopped** from the list.

You can use the following techniques to enhance or refine your searches:

**Inverse search**

Inverse searches let you search for resources that do **not** match a specified value. Inverse searches are performed by prefixing the search keyword with the exclamation mark (!) character.

**Note**
Inverse search is supported with keyword searches and attribute searches on **client** filters only. It is not supported with attribute searches on API filters.

For example, you can search the **Instance state** attribute for all of your instances to exclude all instances that are in the **terminated** state. To do this:
1. In the search field on the **Instances** screen, start entering **Instance state**. As you enter the characters, the two types of filters appear for **Instance state: API filters** and **Client filters**.
2. Choose **Instance state (client)**. Inverse search is only supported on **client** filters.

A list of possible values for the selected attribute appears.
3. Enter ! (exclamation mark) to display the inverse filters.
4. Choose **!terminated** from the list.

To filter instances based on an instance state attribute, you can also use the search icons (
**state** column. The search icon with a plus sign (+) displays all the instances that match that attribute. The search icon with a minus sign (-) excludes all instances that match that attribute.

Here is another example of using the inverse search: To list all instances that are not assigned the security group named launch-wizard-1, search by the **Security group name** attribute, and for the keyword, enter !launch-wizard-1.

**Partial search**

With partial searches, you can search for partial string values. To perform a partial search, enter only a part of the keyword that you want to search for. For example, to search for all t2.micro, t2.small, and t2.medium instances, search by the **Instance Type** attribute, and for the keyword, enter t2.

**Note**

Partial search is supported with keyword searches and attribute searches on client filters only. It is not supported with attribute searches on API filters.

**Regular expression search**

To use regular expression searches, you must enable **Use regular expression matching** in the Preferences.

Regular expressions are useful when you need to match the values in a field with a specific pattern. For example, to search for a value that starts with a, search for ^a. To search for a value that ends with xyz, search for xyz$. Or to search for a value that starts with a number that is followed by one or more characters, search for [0-9]+.*. Regular expression searches are not case-sensitive.

**Note**

Regular expression search is supported with keyword searches and attribute searches on client filters only. It is not supported with attribute searches on API filters.

**Wildcard search**

Use the * wildcard to match zero or more characters. Use the ? wildcard to match zero or one character. For example, if you have a data set with the following values: prod, prods, and production; "prod*" matches all values, whereas "prod?" matches only prod and prods. To use the literals values, escape them with a backslash (\). For example, "prod\*" would match prod*.

**Note**

Wildcard search is supported with attribute searches on API filters only. It is not supported with keyword searches and attribute searches on client filters only.

**Combining searches**

In general, multiple filters with the same attribute are automatically joined with OR. For example, searching for **Instance State** : Running and **Instance State** : Stopped returns all instances that are either running OR stopped. To join search with AND, search across different attributes. For example, searching for **Instance State** : Running and **Instance Type** : c4.large returns only instances that are of type c4.large AND that are in the stopped state.

**Old console**

The old Amazon EC2 console supports the following types of searches:

**Search by keyword**

Searching by keyword is a free text search that lets you search for a value across all of your resources’ attributes. To search by keyword, enter or paste what you’re looking for in the search field, and then choose **Enter**. For example, searching for 123 matches all instances that have 123 in any of their attributes, such as an IP address, instance ID, VPC ID, or AMI ID. If your free text search returns unexpected matches, apply additional filters.
Search by attributes

Searching by an attribute lets you search a specific attribute across all of your resources. For example, you can search the **State** attribute for all of your instances to return only instances that are in the **stopped** state. To do this:

1. In the search field on the Instances screen, start entering **Instance State**. As you enter characters, a list of matching attributes appears.
2. Select **Instance State** from the list. A list of possible values for the selected attribute appears.
3. Select **Stopped** from the list.

You can use the following techniques to enhance or refine your searches:

Inverse search

Inverse searches let you search for resources that do **not** match a specified value. Inverse searches are performed by prefixing the search keyword with the exclamation mark (!) character. For example, to list all instances that are **not** terminated, search by the **Instance State** attribute, and for the keyword, enter **!Terminated**.

Partial search

With partial searches, you can search for partial string values. To perform a partial search, enter only a part of the keyword you want to search for. For example, to search for all **t2.micro**, **t2.small**, and **t2.medium** instances, search by the **Instance Type** attribute, and for the keyword, enter **t2**.

Regular expression search

Regular expressions are useful when you need to match the values in a field with a specific pattern. For example, to search for all instances that have an attribute value that starts with **s**, search for **^s**. Or to search for all instances that have an attribute value that ends with **xyz**, search for **xyz$**. Regular expression searches are not case-sensitive.

Combining searches

In general, multiple filters with the same attribute are automatically joined with **OR**. For example, searching for **Instance State : Running** and **Instance State : Stopped** returns all instances that are either running OR stopped. To join search with **AND**, search across different attributes. For example, searching for **Instance State : Running** and **Instance Type : c4.large** returns only instances that are of type **c4.large** AND that are in the stopped state.

List and filter using the CLI and API

Each resource type has a corresponding CLI command and API action that you use to list resources of that type. The resulting lists of resources can be long, so it can be faster and more useful to filter the results to include only the resources that match specific criteria.

Filtering considerations

- You can specify multiple filters and multiple filter values in a single request.
- You can use wildcards with the filter values. An asterisk (*) matches zero or more characters, and a question mark (?) matches zero or one character.
- Filter values are case sensitive.
- Your search can include the literal values of the wildcard characters; you just need to escape them with a backslash before the character. For example, a value of ```\*amazon\?\``` searches for the literal string ```*amazon?\```.
Supported filters

To see the supported filters for each Amazon EC2 resource, see the following documentation:

- AWS CLI: The describe commands in the AWS CLI Command Reference-Amazon EC2.
- Query API: The Describe API actions in the Amazon EC2 API Reference.

Example Example: Specify a single filter

You can list your Amazon EC2 instances using describe-instances. Without filters, the response contains information for all of your resources. You can use the following command to include only the running instances in your output.

```
aws ec2 describe-instances --filters Name=instance-state-name,Values=running
```

To list only the instance IDs for your running instances, add the --query parameter as follows.

```
aws ec2 describe-instances --filters Name=instance-state-name,Values=running --query "Reservations[*].Instances[*].InstanceId" --output text
```

The following is example output.

```
i-0ef1f57f78d4775a4
i-0626d4edd54f1286d
i-04a636d18e83cfacb
```

Example Example: Specify multiple filters or filter values

If you specify multiple filters or multiple filter values, the resource must match all filters to be included in the results.

You can use the following command to list all instances whose type is either m5.large or m5d.large.

```
aws ec2 describe-instances --filters Name=instance-type,Values=m5.large,m5d.large
```

You can use the following command to list all stopped instances whose type is t2.micro.

```
aws ec2 describe-instances --filters Name=instance-state-name,Values=stopped Name=instance-type,Values=t2.micro
```

Example Example: Use wildcards in a filter value

If you specify database as the filter value for the description filter when describing EBS snapshots using describe-snapshots, the command returns only the snapshots whose description is "database".

```
aws ec2 describe-snapshots --filters Name=description,Values=database
```

The * wildcard matches zero or more characters. If you specify *database* as the filter value, the command returns only snapshots whose description includes the word database.

```
aws ec2 describe-snapshots --filters Name=description,Values=*database*
```
Tag your resources

To help you manage your instances, images, and other Amazon EC2 resources, you can assign your own metadata to each resource in the form of tags. Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type—you can quickly identify a specific resource based on the tags that you've assigned to it. This topic describes tags and shows you how to create them.

Warning
Tag keys and their values are returned by many different API calls. Denying access to DescribeTags doesn't automatically deny access to tags returned by other APIs. As a best practice, we recommend that you do not include sensitive data in your tags.

Contents
- Tag basics (p. 1451)
- Tag your resources (p. 1451)
- Tag restrictions (p. 1454)
- Tags and access management (p. 1455)
- Tag your resources for billing (p. 1455)
- Work with tags using the console (p. 1455)
- Work with tags using the command line (p. 1459)
- Add tags to a resource using CloudFormation (p. 1462)
Tag basics

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value, both of which you define.

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. For example, you could define a set of tags for your account's Amazon EC2 instances that helps you track each instance's owner and stack level.

The following diagram illustrates how tagging works. In this example, you've assigned two tags to each of your instances—one tag with the key Owner and another with the key Stack. Each tag also has an associated value.

We recommend that you devise a set of tag keys that meets your needs for each resource type. Using a consistent set of tag keys makes it easier for you to manage your resources. You can search and filter the resources based on the tags you add. For more information about how to implement an effective resource tagging strategy, see the AWS whitepaper Tagging Best Practices.

Tags don't have any semantic meaning to Amazon EC2 and are interpreted strictly as a string of characters. Also, tags are not automatically assigned to your resources. You can edit tag keys and values, and you can remove tags from a resource at any time. You can set the value of a tag to an empty string, but you can't set the value of a tag to null. If you add a tag that has the same key as an existing tag on that resource, the new value overwrites the old value. If you delete a resource, any tags for the resource are also deleted.

Tag your resources

You can tag most Amazon EC2 resources that already exist in your account. The table (p. 1452) 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. For more information about enabling users to tag resources on creation, see [Grant permission to tag resources during creation](p. 1065).

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

<table>
<thead>
<tr>
<th>Resource</th>
<th>Supports tags</th>
<th>Supports tagging on creation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFI</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AMI</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Bundle task</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Capacity Reservation</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Carrier gateway</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Client VPN endpoint</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Client VPN route</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Customer gateway</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Dedicated Host</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Dedicated Host Reservation</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DHCP option</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>EBS snapshot</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>EBS volume</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>EC2 Fleet</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Egress-only internet gateway</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Elastic IP address</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Elastic Graphics accelerator</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Instance</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Instance store volume</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Internet gateway</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Resource</td>
<td>Supports tags</td>
<td>Supports tagging on creation</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>---------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>IP address pool (BYOIP)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Key pair</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Launch template</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Launch template version</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Local gateway</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Local gateway route table</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Local gateway virtual interface</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Local gateway virtual interface group</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Local gateway route table VPC association</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Local gateway route table virtual interface group association</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NAT gateway</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Network ACL</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Network interface</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Placement group</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Prefix list</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Reserved Instance</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Reserved Instance listing</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Route table</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Spot Fleet request</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Spot Instance request</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Security group</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Security group rule</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Subnet</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Traffic Mirror filter</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Traffic Mirror session</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Traffic Mirror target</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Transit gateway</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Transit gateway route table</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Transit gateway VPC attachment</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Tag restrictions

The following basic restrictions apply to tags:

- Maximum number of tags per resource – 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length – 128 Unicode characters in UTF-8
- Maximum value length – 256 Unicode characters in UTF-8
- Although EC2 allows for any character in its tags, other services are more restrictive. The allowed characters across services are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- The `aws:` prefix is reserved for AWS use. If a tag has a tag key with this prefix, then you can't edit or delete the tag's key or value. Tags with the `aws:` prefix do not count against your tags per resource limit.

You can't terminate, stop, or delete a resource based solely on its tags; you must specify the resource identifier. For example, to delete snapshots that you tagged with a tag key called `DeleteMe`, you...
must use the `DeleteSnapshots` action with the resource identifiers of the snapshots, such as `snap-1234567890abcdef0`.

When you tag public or shared resources, the tags you assign are available only to your AWS account; no other AWS account will have access to those tags. For tag-based access control to shared resources, each AWS account must assign its own set of tags to control access to the resource.

You can't tag all resources. For more information, see Tagging support for Amazon EC2 resources (p. 1452).

**Tags and access management**

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 Grant permission to tag resources during creation (p. 1065).

You can also use resource tags to implement attribute-based control (ABAC). You can create IAM policies that allow operations based on the tags for the resource. For more information, see Control access to EC2 resources using resource tags (p. 1067).

**Tag 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 Monthly cost allocation report in the AWS Billing and Cost Management User Guide. To see the cost of your combined resources, you can organize your billing information based on resources that have the same tag key values. For example, you can tag several resources with a specific application name, and then organize your billing information to see the total cost of that application across several services. For more information, see Using cost allocation tags in the AWS Billing and Cost Management User Guide.

**Note**

If you've just enabled reporting, data for the current month is available for viewing after 24 hours.

Cost allocation tags can indicate which resources are contributing to costs, but deleting or deactivating resources doesn't always reduce costs. For example, snapshot data that is referenced by another snapshot is preserved, even if the snapshot that contains the original data is deleted. For more information, see Amazon Elastic Block Store volumes and snapshots in the AWS Billing and Cost Management User Guide.

**Note**

Elastic IP addresses that are tagged do not appear on your cost allocation report.

**Work 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 List and filter your resources (p. 1444).

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 Tag Editor in Getting Started with the AWS Management Console.
Tasks

- Display tags (p. 1456)
- Add and delete tags on an individual resource (p. 1457)
- Add and delete tags to a group of resources (p. 1458)
- Add a tag when you launch an instance (p. 1458)
- Filter a list of resources by tag (p. 1459)

Display 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.

Display tags for individual resources

When you select a resource-specific page in the Amazon EC2 console, it displays a list of those resources. For example, if you select Instances from the navigation pane, the console displays your Amazon EC2 instances. When you select a resource from one of these lists (for example, an instance), if the resource supports tags, you can view and manage its tags. On most resource pages, you can view the tags by choosing the Tags tab.

You can add a column to the resource list that displays all values for tags with the same key. You can use this column sort and filter the resource list by the tag.

New console

- Choose the Preferences gear-shaped icon in the top right corner of the screen. In the Preferences dialog box, under Tag columns, select one or more tag keys, and then choose Confirm.

Old console

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.

Display 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.
Add and delete 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. 1442).
3. In the navigation pane, select a resource type (for example, Instances).
4. Select the resource from the resource list and choose the Tags tab.
5. Choose Manage tags, Add tag. Enter the key and value for the tag. When you are finished adding tags, choose Save.

**To delete a tag from an individual resource**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can’t. For more information, see Resource locations (p. 1442).
3. In the navigation pane, select a resource type (for example, Instances).
4. Select the resource from the resource list and choose the Tags tab.
5. Choose Manage tags. For each tag, choose Remove. When you are finished removing tags, choose Save.
Add and delete 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. 1442).
3. In the navigation pane, choose Tags.
4. At the top of the content pane, choose Manage Tags.
5. For Filter, select the type of resource (for example, instances).
6. In the resources list, select the check box next to each resource.
7. Under Add Tag, enter the tag key and value and choose Add Tag.

Note
If you add a new tag with the same tag key as an existing tag, the new tag overwrites the existing tag.

To remove a tag from a group of resources

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select the Region that meets your needs. This choice is important because some Amazon EC2 resources can be shared between Regions, while others can't. For more information, see Resource locations (p. 1442).
3. In the navigation pane, choose Tags, Manage Tags.
4. To view the tags in use, select the Show/Hide Columns gear-shaped icon, and in the Show/Hide Columns dialog box, select the tag keys to view and choose Close.
5. For Filter, select the type of resource (for example, instances).
6. In the resource list, select the check box next to each resource.
7. Under Remove Tag, enter the tag key and choose Remove Tag.

Add 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. 1442).
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, see Find a Windows AMI (p. 97).
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**.

**Filter a list of resources by tag**

You can filter your list of resources based on one or more tag keys and tag values.

**To filter a list of resources by tag**

1. In the navigation pane, select a resource type (for example, **Instances**).
2. Choose the search field.
3. Choose the tag key from in the list.
4. Choose the corresponding tag value from the list.
5. When you are finished, remove the filter.

For more information about filters, see **List and filter your resources (p. 1444)**.

**Work with tags using the command line**

You can add tags to many EC2 resource when you create them, using the tag specifications parameter for the create command. You can view the tags for a resource using the describe command for the resource. You can also add, update, or delete tags for your existing resources using the following commands.

<table>
<thead>
<tr>
<th>Task</th>
<th>AWS CLI</th>
<th>AWS Tools for Windows PowerShell</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add or overwrite one or more tags</td>
<td>create-tags</td>
<td>New-EC2Tag</td>
</tr>
<tr>
<td>Delete one or more tags</td>
<td>delete-tags</td>
<td>Remove-EC2Tag</td>
</tr>
<tr>
<td>Describe one or more tags</td>
<td>describe-tags</td>
<td>Get-EC2Tag</td>
</tr>
</tbody>
</table>

**Tasks**

- Add tags on resource creation (p. 1459)
- Add tags to an existing resource (p. 1460)
- Describe tagged resources (p. 1461)

**Add tags on resource creation**

The following examples demonstrate how to apply tags when you create resources.

The way you enter JSON-formatted parameters on the command line differs depending on your operating system. Linux, macOS, or Unix and Windows PowerShell use single quotes (') to enclose the JSON data structure. Omit the single quotes when using the commands with the Windows command line. For more information, see **Specifying parameter values for the AWS CLI**.
Work with tags using the command line

Example Example: Launch an instance and apply tags to the instance and volume

The following `run-instances` command launches an instance and applies a tag with the key `webserver` and the value `production` to the instance. The command also applies a tag with the key `cost-center` and the value `cc123` to any EBS volume that's created (in this case, the root volume).

```
aws ec2 run-instances \
--image-id ami-abc12345 \
--count 1 \
--instance-type t2.micro \
--key-name MyKeyPair \
--subnet-id subnet-6e7f829e \
--tag-specifications 'ResourceType=instance,Tags=[{Key=webserver,Value=production}],' 'ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
```

You can apply the same tag keys and values to both instances and volumes during launch. The following command launches an instance and applies a tag with a key of `cost-center` and a value of `cc123` to both the instance and any EBS volume that's created.

```
aws ec2 run-instances \
--image-id ami-abc12345 \
--count 1 \
--instance-type t2.micro \
--key-name MyKeyPair \
--subnet-id subnet-6e7f829e \
--tag-specifications 'ResourceType=instance,Tags=[{Key=cost-center,Value=cc123}]','ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
```

Example Example: Create a volume and apply a tag

The following `create-volume` command creates a volume and applies two tags: `purpose=production` and `cost-center=cc123`.

```
aws ec2 create-volume \
--availability-zone us-east-1a \
--volume-type gp2 \
--size 80 \
--tag-specifications 'ResourceType=volume,Tags=[{Key=purpose,Value=production},{Key=cost-center,Value=cc123}]'
```

Add tags to an existing resource

The following examples demonstrate how to add tags to an existing resource using the `create-tags` command.

Example Example: Add a tag to a resource

The following command adds the tag `Stack=production` to the specified image, or overwrites an existing tag for the AMI where the tag key is `Stack`. If the command succeeds, no output is returned.

```
aws ec2 create-tags \
--resources ami-78a54011 \
--tags Key=Stack,Value=production
```

Example Example: Add tags to multiple resources

This example adds (or overwrites) two tags for an AMI and an instance. One of the tags contains just a key (`webserver`), with no value (we set the value to an empty string). The other tag consists of a key (`stack`) and value (`Production`). If the command succeeds, no output is returned.
Work with tags using the command line

```
aws ec2 create-tags \
  --resources ami-1a2b3c4d i-1234567890abcdef0 \
  --tags Key=webserver,Value=  Key=stack,Value=Production
```

Example Example: Add tags with special characters

This example adds the tag \[Group\]=test to an instance. The square brackets ([ and ]) are special characters, which must be escaped.

If you are using Linux or OS X, to escape the special characters, enclose the element with the special character with double quotes ("), and then enclose the entire key and value structure with single quotes (').

```
aws ec2 create-tags \
  --resources i-1234567890abcdef0 \
  --tags 'Key=[Group],Value=test'
```

If you are using Windows, to escape the special characters, enclose the element that has special characters with double quotes ("), and then precede each double quote character with a backslash (\) as follows:

```
aws ec2 create-tags ^ \
  --resources i-1234567890abcdef0 ^ \
  --tags Key="[Group]\",Value=test
```

If you are using Windows PowerShell, to escape the special characters, enclose the value that has special characters with double quotes ("), precede each double quote character with a backslash (\), and then enclose the entire key and value structure with single quotes (') as follows:

```
aws ec2 create-tags ` \
  --resources i-1234567890abcdef0 ` \
  --tags 'Key="[Group]\",Value=test'
```

Describe tagged resources

The following examples show you how to use filters with the describe-instances to view instances with specific tags. All EC2 describe commands use this syntax to filter by tag across a single resource type. Alternatively, you can use the describe-tags command to filter by tag across EC2 resource types.

Example Example: Describe instances with the specified tag key

The following command describes the instances with a Stack tag, regardless of the value of the tag.

```
aws ec2 describe-instances \
  --filters Name=tag-key,Values=Stack
```

Example Example: Describe instances with the specified tag

The following command describes the instances with the tag Stack=production.

```
aws ec2 describe-instances \
  --filters Name=tag:Stack,Values=production
```

Example Example: Describe instances with the specified tag value

The following command describes the instances with a tag with the value production, regardless of the tag key.
aws ec2 describe-instances \ 
  --filters Name=tag-value,Values=production

Example Example: Describe all EC2 resources with the specified tag

The following command describes all EC2 resources with the tag `Stack=Test`.

aws ec2 describe-tags \ 
  --filters Name=key,Values=Stack Name=value,Values=Test

Add tags to a resource using CloudFormation

With Amazon EC2 resource types, you specify tags using either a `Tags` or `TagSpecifications` property.

The following examples add the tag `Stack=Production` to `AWS::EC2::Instance` using its `Tags` property.

Example Example: Tags in YAML

```
Tags:
  - Key: "Stack"
    Value: "Production"
```

Example Example: Tags in JSON

```
"Tags": [
  {
    "Key": "Stack",
    "Value": "Production"
  }
]
```

The following examples add the tag `Stack=Production` to `AWS::EC2::LaunchTemplate LaunchTemplateData` using its `TagSpecifications` property.

Example Example: TagSpecifications in YAML

```
TagSpecifications:
  - ResourceType: "instance"
    Tags:
      - Key: "Stack"
        Value: "Production"
```

Example Example: TagSpecifications in JSON

```
"TagSpecifications": [
  {
    "ResourceType": "instance",
    "Tags": [
      {
        "Key": "Stack",
        "Value": "Production"
      }
    ]
  }
]```
Amazon EC2 service quotas

Amazon EC2 provides different resources that you can use. These resources include images, instances, volumes, and snapshots. When you create your AWS account, we set default quotas (also referred to as limits) on these resources on a per-Region basis. For example, there is a maximum number of instances that you can launch in a Region. So if you were to launch an instance in the US West (Oregon) Region, for example, the request must not cause your usage to exceed your maximum number of instances in that Region.

The Amazon EC2 console provides limit information for the resources managed by the Amazon EC2 and Amazon VPC consoles. You can request an increase for many of these limits. Use the limit information that we provide to manage your AWS infrastructure. Plan to request any limit increases in advance of the time that you'll need them.

For more information, see Amazon EC2 endpoints and quotas in the Amazon Web Services General Reference. For information about Amazon EBS quotas, see Amazon EBS quotas (p. 1392).

View your current limits

Use the Limits page in the Amazon EC2 console to view the current limits for resources provided by Amazon EC2 and Amazon VPC, on a per-Region basis.

To view your current limits

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select a Region.
3. From the navigation pane, choose Limits.
4. Locate the resource in the list. You can use the search fields to filter the list by resource name or resource group. The Current limit column displays the current maximum for the resource for your account.

Request an increase

Use the Limits page in the Amazon EC2 console to request an increase in your Amazon EC2 or Amazon VPC resources, on a per-Region basis.

Alternatively, request an increase using Service Quotas. For more information, see Requesting a quota increase in the Service Quotas User Guide.

To request an increase using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/
2. From the navigation bar, select a Region.
3. From the navigation pane, choose Limits.
4. Select the resource in the list, and choose Request limit increase.
5. Complete the required fields on the limit increase form and choose Submit. We'll respond to you using the contact method that you specified.

Restriction on email sent using port 25

On all instances, Amazon EC2 restricts traffic on port 25 by default. You can request that this restriction be removed. For more information, see How do I remove the restriction on port 25 from my EC2 instance? in the AWS Knowledge Center.

Amazon EC2 usage reports

AWS provides a free reporting tool called AWS Cost Explorer that enables you to analyze the cost and usage of your EC2 instances and the usage of your Reserved Instances. You can view data up to the last 13 months, and forecast how much you are likely to spend for the next three months. You can use Cost Explorer to see patterns in how much you spend on AWS resources over time, identify areas that need further inquiry, and see trends that you can use to understand your costs. You also can specify time ranges for the data, and view time data by day or by month.

Here’s an example of some of the questions that you can answer when using Cost Explorer:

- How much am I spending on instances of each instance type?
- How many instance hours are being used by a particular department?
- How is my instance usage distributed across Availability Zones?
- How is my instance usage distributed across AWS accounts?
- How well am I using my Reserved Instances?
- Are my Reserved Instances helping me save money?

For more information about working with reports in Cost Explorer, including saving reports, see Analyzing your costs with Cost Explorer.
Troubleshoot EC2 Windows instances

The following procedures and tips can help you troubleshoot problems with your Amazon EC2 Windows instances.

Contents

- Troubleshoot instance launch issues (p. 1465)
- Troubleshoot connecting to your Windows instance (p. 1468)
- Troubleshoot an unreachable instance (p. 1475)
- Reset a lost or expired Windows administrator password (p. 1484)
- Troubleshoot stopping your instance (p. 1495)
- Troubleshoot instance termination (shutting down) (p. 1497)
- Troubleshoot Sysprep (p. 1498)
- Use EC2Rescue for Windows Server (p. 1499)
- EC2 Serial Console for Windows instances (p. 1511)
- Send a diagnostic interrupt (for advanced users) (p. 1525)
- Common issues with Windows instances (p. 1526)
- Common messages troubleshooting Windows instances (p. 1530)

To get additional information for troubleshooting problems with your instance, use Use EC2Rescue for Windows Server (p. 1499). For information about troubleshooting issues with PV drivers, see Troubleshoot PV drivers (p. 530).

Troubleshoot instance launch issues

The following issues prevent you from launching an instance.

Launch Issues

- Instance limit exceeded (p. 1465)
- Insufficient instance capacity (p. 1466)
- The requested configuration is currently not supported. Please check the documentation for supported configurations. (p. 1466)
- Instance terminates immediately (p. 1467)
- High CPU usage shortly after Windows starts (p. 1468)

Instance limit exceeded

Description

You get the InstanceLimitExceeded error when you try to launch a new instance or restart a stopped instance.
Causes

If you get an \texttt{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.

Solutions

You can request an instance limit increase on a per-region basis. For more information, see Amazon EC2 service quotas (p. 1463).

Insufficient instance capacity

Description

You get the \texttt{InsufficientInstanceCapacity} error when you try to launch a new instance or restart a stopped instance.

Cause

If you get this error when you try to launch an instance or restart a stopped instance, AWS does not currently have enough available On-Demand capacity to fulfill 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 Change the instance type (p. 231).
- 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. 978).

The requested configuration is currently not supported. Please check the documentation for supported configurations.

Description

You get the \texttt{Unsupported} error when you try to launch a new instance because the instance configuration is not supported.

Cause

The error message provides additional details. For example, an instance type or instance purchasing option might not be supported in the specified Region or Availability Zone.
Solution

Try a different instance configuration. To search for an instance type that meets your requirements, see Find an Amazon EC2 instance type (p. 230).

Instance terminates immediately

Description

Your instance goes from the pending state to the terminated state.

Cause

The following are a few reasons why an instance might immediately terminate:

- You've exceeded your EBS volume limits. For more information, see Instance volume limits (p. 1407).
- An EBS snapshot is corrupted.
- The root EBS volume is encrypted and you do not have permissions to access the KMS key for decryption.
- A snapshot specified in the block device mapping for the AMI is encrypted and you do not have permissions to access the KMS key for decryption or you do not have access to the KMS key to encrypt the restored volumes.
- The instance store-backed AMI that you used to launch the instance is missing a required part (an image.part.xx file).

For more information, get the termination reason using one of the following methods.

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. On the first tab, find the reason next to State transition reason.

To get the termination reason using the AWS Command Line Interface

1. Use the describe-instances command and specify the instance ID.

   ```bash
   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.

   ```json
   "StateReason": {
   "Code": "Server.InternalError"
   },
   ```

To get the termination reason using AWS CloudTrail
For more information, see Viewing events with CloudTrail event history in the AWS CloudTrail User Guide.

Solution

Depending on the termination reason, take one of the following actions:

- **Client.VolumeLimitExceeded**: Volume limit exceeded — Delete unused volumes. You can submit a request to increase your volume limit.
- **Client.InternalError**: Client error on launch — Ensure that you have the permissions required to access the AWS KMS keys used to decrypt and encrypt volumes. For more information, see Using key policies in AWS KMS in the AWS Key Management Service Developer Guide.

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 run this script, specify a value for /d. The default value is 3. Possible values include the following:

1. Never check for updates
2. Check for updates but let me choose whether to download and install them
3. Download updates but let me choose whether to install them
4. Install updates automatically

After you modify the user data for your instance, you can run it. For more information, see View and update the instance user data (p. 576) and User data execution (p. 574).

Troubleshoot connecting to your Windows instance

The following are possible problems you may have and error messages you may see while trying to connect to your Windows instance.

Contents

- Remote Desktop can't connect to the remote computer (p. 1469)
- Error using the macOS RDP client (p. 1471)
- RDP displays a black screen instead of the desktop (p. 1471)
- Unable to remotely log on to an instance with a user account that is not an administrator (p. 1472)
- Troubleshooting Remote Desktop issues using AWS Systems Manager (p. 1472)
- Enable Remote Desktop on an EC2 Instance With Remote Registry (p. 1474)
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. 926).
- 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. 6).
- 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 white space character when you copied the password.
- Verify that the instance has passed status checks. For more information, see Status checks for your instances (p. 809) and Troubleshoot 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 AWSSupport-TroubleshootRDP (p. 1472) to disable the Windows Firewall profiles using SSM Agent. To disable Windows Firewall on a Windows instance that is not configured for AWS Systems Manager, use AWSSupport-ExecuteEC2Rescue (p. 1474), 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 AWSSupport-TroubleshootRDP (p. 1472) to disable NLA).
- Verify that the Remote Desktop Service (TermService) Startup Type is Automatic and the service is started (use AWSSupport-TroubleshootRDP (p. 1472) 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 AWSSupport-TroubleshootRDP (p. 1472) to read the current RDP port and change it back to 3389).
- Verify that Remote Desktop connections are allowed on your instance (use AWSSupport-TroubleshootRDP (p. 1472) 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 Reset a lost or expired Windows administrator password (p. 1484).
- 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 the 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, and then choose Actions, Monitor and troubleshoot, 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 Change the instance type (p. 231).
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 instance with 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.

Troubleshooting Remote Desktop issues using AWS Systems Manager

You can use AWS Systems Manager to troubleshoot issues connecting to your Windows instance using RDP.

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 with EC2 instances, on-premises instances, and virtual machines (VMs) that are enabled for use with AWS Systems Manager (managed instances). In addition, it can also be used with EC2 instances for Windows Server that are not enabled for use with Systems Manager. For information about enabling instances for use with AWS Systems Manager, see AWS Systems Manager Managed Instances in the AWS Systems Manager User Guide.

To troubleshoot using the AWSSupport-TroubleshootRDP document

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. For Execution Mode, choose Simple execution.
5. For Input parameters, Instanceld, enable Show interactive instance picker.
6. Choose your Amazon EC2 instance.
7. Review the examples (p. 1472), then choose Execute.
8. To monitor the execution progress, for 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 an item from Step ID.

AWSSupport-TroubleshootRDP examples

The following examples show you how to accomplish common troubleshooting tasks using AWSSupport-TroubleshootRDP. You can use either the example AWS CLI start-automation-execution command or the provided link to the AWS Management Console.

Example Example: Check the current RDP status

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --parameters "InstanceId=instance_id" --region region_code
```
Example Example: Disable the Windows Firewall

AWS CLI:

```bash
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --parameters "InstanceId=instance_id,Firewall=Disable" --region region_code
```

Example Example: Disable Network Level Authentication

AWS CLI:

```bash
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --parameters "InstanceId=instance_id,NLASettingAction=Disable" --region region_code
```

Example Example: Set RDP Service Startup Type to Automatic and start the RDP service

AWS CLI:

```bash
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --parameters "InstanceId=instance_id,RDPServiceStartupType=Auto, RDPServiceAction=Start" --region region_code
```

Example Example: Restore the default RDP Port (3389)

AWS CLI:

```bash
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --parameters "InstanceId=instance_id,RDPPortAction=Modify" --region region_code
```
Enable Remote Desktop on an EC2 Instance With Remote Registry

If your unreachable instance is not managed by AWS Systems Manager Session Manager, then you can use remote registry to enable Remote Desktop.

1. From the EC2 console, stop the unreachable instance.
2. Attach the root volume of the unreachable instance to another instance in the same Availability Zone.

3. On the instance to which you attached the root volume, open Disk Management. To open Disk Management, run

```
diskmgmt.msc
```

4. Right click on the root volume of the affected instance and choose **Online**.

5. Open the Windows Registry Editor by running the following command:

```
regedit
```

6. In the Registry Editor console tree, choose **HKEY_LOCAL_MACHINE**, then select **File>Load Hive**.

7. Select the drive of the attached volume, navigate to `\Windows\System32\config\`, select **SYSTEM**, and then choose **Open**.

8. For **Key Name**, enter a unique name for the hive and choose **OK**.

9. Back up the registry hive before making any changes to the registry.

   a. In the Registry Editor console tree, select the hive that you loaded: **HKEY_LOCAL_MACHINE \your key name**.

   b. Choose **File>Export**.

   c. In the Export Registry File dialog box, choose the location to which you want to save the backup copy, and then type a name for the backup file in the **File name** field.

   d. Choose **Save**.

10. In the Registry Editor console tree, navigate to **HKEY_LOCAL_MACHINE\your key name\ControlSet001\Control\Terminal Server**, and then, in the details pane, double-click on **fDenyTSConnections**.

11. In the **Edit DWORD value** box, enter 0 in the **Value data** field.

12. Choose **OK**.

   **Note**

   If the value in the **Value data** field is 1, then the instance will deny remote desktop connections. A value of 0 allows remote desktop connections.

13. Close the Registry Editor and the Disk Management consoles.

14. From the EC2 console, detach the root volume from the instance to which you attached it and reattach it to the unreachable instance. When attaching the volume to the unreachable instance, enter `/dev/sda1` in the **device** field.

15. Restart the unreachable instance.

---

** Troubleshoot an unreachable instance **

If you are unable to reach your Windows instance through SSH or RDP, you can capture a screenshot of your instance and view it as an image. This provides visibility into the status of the instance, and allows for quicker troubleshooting. You can also use **EC2 Rescue** (p. 1499) on instances running Windows Server 2008 or later to gather and analyze date from offline instances.

- Get a screenshot of an unreachable instance (p. 1476)
- Common screenshots (p. 1477)

For information about troubleshooting an unreachable Linux instance, see Troubleshoot an unreachable instance.
Get a screenshot of an unreachable instance

You can get screenshots of an instance while it is running or after it has crashed. There is no data transfer cost for the screenshot. The image is generated in JPG format and is no larger than 100 kb. This feature is not supported when the instance is using an NVIDIA GRID driver, is on bare metal instances (instances of type *.metal*), or is powered by Arm-based Graviton or Graviton 2 processors. This feature is available in the following Regions:

- Asia Pacific (Hong Kong) Region
- Asia Pacific (Tokyo) Region
- Asia Pacific (Seoul) Region
- Asia Pacific (Singapore) Region
- Asia Pacific (Sydney) Region
- Asia Pacific (Mumbai) Region
- US East (N. Virginia) Region
- US East (Ohio) Region
- US West (Oregon) Region
- US West (N. California) Region
- Europe (Ireland) Region
- Europe (Frankfurt) Region
- Europe (Milan) Region
- Europe (London) Region
- Europe (Paris) Region
- Europe (Stockholm) Region
- Europe (Paris) Region
- South America (São Paulo) Region
- Canada (Central) Region
- Middle East (Bahrain) Region
- Africa (Capetown) Region
- China (Beijing) Region
- China (Ningxia) Region

To get a screenshot of a running instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose Instances.
3. Select the instance to capture.
4. Choose Actions, Monitor and troubleshoot, Get instance screenshot.
5. Choose Download, or right-click the image to download and save it.

To get a screenshot of a running instance 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 Access Amazon EC2 (p. 3).

- get-console-screenshot (AWS CLI)
- GetConsoleScreenshot (Amazon EC2 Query API)
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. 1477)
- Recovery console screen (p. 1479)
- Windows boot manager screen (p. 1481)
- Sysprep screen (p. 1481)
- Getting ready screen (p. 1482)
- Windows Update screen (p. 1483)
- Chkdsk (p. 1484)

Log on screen (Ctrl+Alt+Delete)

Console Screenshot Service returned the following.

If an instance becomes unreachable during logon, 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**

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Verify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security group configuration</td>
<td>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. 1469).</td>
</tr>
<tr>
<td>VPC configuration (Network ACLs)</td>
<td>Verify that the access control list (ACL) for your Amazon VPC is not blocking access. For information, see Network ACLs in the Amazon VPC User Guide.</td>
</tr>
<tr>
<td>VPN configuration</td>
<td>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?</td>
</tr>
</tbody>
</table>

**Windows network configuration**

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Verify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Firewall</td>
<td>Verify that Windows Firewall isn't blocking connections to your instance. Disable Windows Firewall as described in bullet 7 of the remote desktop troubleshooting section, Remote Desktop can't connect to the remote computer (p. 1469).</td>
</tr>
<tr>
<td>Advanced TCP/IP configuration (Use of static IP)</td>
<td>The instance may be unresponsive because you configured a static IP address. For a VPC, create a network interface (p. 948) and attach it to the instance (p. 950). For EC2 Classic, enable DHCP.</td>
</tr>
</tbody>
</table>

**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 issues**

If the instance can't be reached during logon, there could a problem with Remote Desktop Services (RDS) on the instance.

**Remote Desktop Services configuration**

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Verify</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDS is running</td>
<td>Verify that RDS is running on the instance. Connect to the instance using the Microsoft Management Console (MMC) Services snap-in (services.msc). In the list of services, verify that Remote</td>
</tr>
<tr>
<td>Configuration</td>
<td>Verify</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Desktop Services</td>
<td><strong>Verify</strong> 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 Detach an Amazon EBS volume from a Windows instance (p. 1204).</td>
</tr>
<tr>
<td>RDS is enabled</td>
<td>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 Enable Remote Desktop on an EC2 Instance With Remote Registry (p. 1474). When you are finished, reattach the root volume to the original instance. For more information, see Detach an Amazon EBS volume from a Windows instance (p. 1204).</td>
</tr>
</tbody>
</table>

**High CPU usage**

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, see Get Statistics for a Specific Resource in the Amazon CloudWatch User Guide. For additional troubleshooting tips, see High CPU usage shortly after Windows starts (p. 1468).

**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 Detach an Amazon EBS volume from a Windows instance (p. 1204).

   **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. 1469) to avoid a disk signature collision.

3. Log in to the instance and run the following command from a command prompt to change the `bootstatuspolicy` configuration to `ignoreallfailures`:

   ```
   bcdedit /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.

Window Boot Manager

Windows failed to start. A recent hardware or software change might be the cause. To fix the problem:

1. Insert your Windows installation disc and restart your computer.
2. Choose your language settings, and then click “Next.”
3. Click “Repair your computer.”

If you do not have this disc, contact your system administrator or computer manufacturer for assistance.

File: \Boot\BCD
Status: 0xc000000f
Info: The Boot Configuration Data for your PC is missing or contains errors.

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 Detach an Amazon EBS volume from a Windows instance (p. 1204).

Sysprep screen

Console Screenshot Service returned the following.
You may see this screen if you did not use the EC2Config Service to call Sysprep or if the operating system failed while running Sysprep. You can reset the password using EC2Rescue (p. 1499). Otherwise, Create a standardized Amazon Machine Image (AMI) using Sysprep (p. 40).

**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.

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.

Reset 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 Passwords and SSH Keys on Amazon EC2 Instances in the AWS Systems Manager User Guide.

The manual methods to reset the administrator password use EC2Launch v2, EC2Config, or EC2Launch.

• For all supported Windows AMIs that include the EC2Launch v2 service, use EC2Launch v2.
• For Windows AMIs before Windows Server 2016, use the EC2Config service.
• For Windows Server 2016 and later 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
Reset the Windows administrator password using EC2Launch v2

If you have lost your Windows administrator password and are using a supported Windows AMI that includes the EC2Launch v2 service, you can use EC2Launch v2 to generate a new password.

If you are using a Windows Server 2016 or later AMI that does not include the EC2Launch v2 service, see Reset the Windows administrator password using EC2Launch (p. 1492).

If you are using a Windows Server AMI earlier than Windows Server 2016 that does not include the EC2Launch v2 service, see Reset the Windows administrator password using EC2Config (p. 1488).

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 Passwords and SSH Keys on Amazon EC2 Instances in the AWS Systems Manager User Guide.

To reset your Windows administrator password using EC2Launch v2, you need to do the following:

• Step 1: Verify that the EC2Launch v2 service is running (p. 1485)
• Step 2: Detach the root volume from the instance (p. 1486)
• Step 3: Attach the volume to a temporary instance (p. 1486)
• Step 4: Delete the .run-once file (p. 1487)
• Step 5: Restart the original instance (p. 1487)

Step 1: Verify that the EC2Launch v2 service is running

Before you attempt to reset the administrator password, verify that the EC2Launch v2 service is installed and running. You use the EC2Launch v2 service to reset the administrator password later in this section.

To verify that the EC2Launch v2 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 select the instance that requires a password reset. This instance is referred to as the original instance in this procedure.
3. Choose Actions, Monitor and troubleshoot, Get system log.
4. Locate the EC2 Launch entry, for example, Launch: EC2Launch v2 service v2.0.124. If you see this entry, the EC2Launch v2 service is running.

If the system log output is empty, or if the EC2Launch v2 service is not running, troubleshoot the instance using the Instance Console Screenshot service. For more information, see Troubleshoot an unreachable instance (p. 1475).
Step 2: Detach the root volume from the instance

You can't use EC2Launch v2 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 requires a password reset and choose Actions, Instance state, Stop instance. After the status of the instance 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 and templates, Create image. Type a name and a description for the image and choose Create image. In the navigation pane, choose AMIs. After the image status changes to available, continue to the next step.
   d. Select the image and choose Actions, and then 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 instances, 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 2019, launch the temporary instance using the base AMI for Windows Server 2016.

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: Delete the `.run-once` file**

After you have attached the volume to the temporary instance as a secondary volume, delete the `.run-once` file from the instance, located at `%ProgramData%/Amazon/EC2Launch/state/.run-once`. This directs EC2Launch v2 to run all tasks with a frequency of once, which includes setting the administrator password.

**Important**
Any scripts set to run once will be triggered by this action.

**Step 5: Restart the original instance**

After you have deleted the `.run-once` 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 **Instance state, Start instance**. 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 **Connect to your Windows instance** (p. 413).

**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. 386).
4. (Optional) If you have no further use for the temporary instance, you can terminate it. Select the temporary instance, and choose **Instance State, Terminate instance**.

**Reset 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 or later AMI, see **Reset the Windows administrator password using EC2Launch** (p. 1492).

**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 Passwords and SSH Keys on Amazon EC2 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. 1488)
- **Step 2: Detach the root volume from the instance** (p. 1488)
- **Step 3: Attach the volume to a temporary instance** (p. 1489)
- **Step 4: Modify the configuration file** (p. 1490)
- **Step 5: Restart the original instance** (p. 1491)

**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/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Instances** and then select the instance that requires a password reset. This instance is referred to as the **original** instance in this procedure.
3. Choose **Actions, Monitor and troubleshoot, 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. 1475).

**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 requires a password reset and choose Actions, Instance state, Stop instance. After the status of the instance 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 and templates, Create image. Type a name and a description for the image and choose Create image. In the navigation pane, choose AMIs. After the image status changes to available, continue to the next step.
   d. Select the image and choose Actions, and then 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 instances, 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 2019, launch the temporary instance using the base AMI for Windows Server 2016.
   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 Instance state, Start instance. 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 Connect to your Windows instance (p. 413).
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. 386).

4. (Optional) If you have no further use for the temporary instance, you can terminate it. Select the temporary instance, and choose Instance State, Terminate instance.

Reset the Windows administrator password using EC2Launch

If you have lost your Windows administrator password and are using a Windows Server 2016 or later 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 Reset the Windows administrator password using EC2Config (p. 1488).

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 Passwords and SSH Keys on Amazon EC2 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. 1492)
- Step 2: Attach the volume to a temporary instance (p. 1493)
- Step 3: Reset the administrator password (p. 1494)
- Step 4: Restart the original instance (p. 1494)

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 requires a password reset and choose Actions, Instance state, Stop instance. After the status of the instance 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.
Reset using EC2Launch

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 and templates, Create image. Type a name and a description for the image and choose Create image. In the navigation pane, choose AMIs. After the image status changes to available, continue to the next step.

d. Select the image and choose Actions, and then 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.

5. Detach the root volume from the original instance as follows:

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 instances, 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 2019, launch the temporary instance using the base AMI for Windows Server 2016.

   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.


   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 (xvdf) volume from the temporary instance as follows:
   a. In the navigation pane, choose Instances and select the temporary instance.
   b. On the Storage tab for 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 **Instance state**, **Start instance**. 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 [Connect to your Windows instance](#) (p. 413).

4. (Optional) If you have no further use for the temporary instance, you can terminate it. Select the temporary instance, and choose **Instance State**, **Terminate instance**.

### Troubleshoot stopping your instance

If you have stopped your Amazon EBS-backed instance and it appears stuck in the **stopping** state, there may be an issue with the underlying host computer.

There is no cost for instance usage while an instance is in the **stopping** state or in any other state except **running**. You are only charged for instance usage when an instance is in the **running** state.

### Force stop the instance

Force the instance to stop using either the console or the AWS CLI.

**New console**

**To force stop the instance using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Instances** and select the stuck instance.
3. Choose **Instance state**, **Force stop instance**, **Stop**.

**Old console**

**To force stop the instance using the console**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose **Instances** and select the stuck instance.
3. Choose **Instance State**, **Stop**, **Yes, Forcefully Stop**.

**AWS CLI**

**To force stop the instance using the AWS CLI**

Use the `stop-instances` command and the `--force` option as follows:

```
aws ec2 stop-instances --instance-ids i-0123ab456c789d01e --force
```

If, after 10 minutes, the instance has not stopped, post a request for help in the Amazon EC2 forum. To help expedite a resolution, include the instance ID, and describe the steps that you've already taken. Alternatively, if you have a support plan, create a technical support case in the Support Center.
Create a replacement instance

To attempt to resolve the problem while you are waiting for assistance from the Amazon EC2 forum or the Support Center, create a replacement instance. Create an AMI of the stuck instance, and launch a new instance using the new AMI.

New console

To create a replacement instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances and select the stuck instance.
3. Choose Actions, Image and templates, Create image.
4. On the Create image page, do the following:
   a. Enter a name and description for the AMI.
   b. Choose No reboot.
   c. Choose Create image.

   For more information, see Create a Windows AMI from a running instance (p. 38).
5. Launch a new instance from the AMI and verify that the new instance is working.
6. Select the stuck instance, and choose Actions, Instance state, Terminate instance. If the instance also gets stuck terminating, Amazon EC2 automatically forces it to terminate within a few hours.

Old console

To create a replacement instance using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances and select the stuck instance.
4. In the Create Image dialog box, fill in the following fields, and then choose Create Image:
   a. Specify a name and description for the AMI.
   b. Choose No reboot.

   For more information, see Create a Windows AMI from a running instance (p. 38).
5. Launch a new instance from the AMI and verify that the new instance is working.
6. Select the stuck instance, and choose Actions, Instance State, Terminate. If the instance also gets stuck terminating, Amazon EC2 automatically forces it to terminate within a few hours.

AWS CLI

To create a replacement instance using the CLI

1. Create an AMI from the stuck instance using the create-image (AWS CLI) command and the --no-reboot option as follows:

   ```bash
   aws ec2 create-image --instance-id i-0123ab456c789d01e --name "AMI" --no-reboot
   ```
2. Launch a new instance from the AMI using the `run-instances` (AWS CLI) command as follows:

```
aws ec2 run-instances --image-id ami-1a2b3c4d --count 1 --instance-type c3.large --key-name MyKeyPair --security-groups MySecurityGroup
```

3. Verify that the new instance is working.

4. Terminate the stuck instance using the `terminate-instances` (AWS CLI) command as follows:

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

If you are unable to create an AMI from the instance as described in the previous procedure, you can set up a replacement instance as follows:

(Alternate) To create a replacement instance using the console

1. Select the instance and choose **Description, Block devices**. Select each volume and make note of its volume ID. Be sure to note which volume is the root volume.
2. In the navigation pane, choose **Volumes**. Select each volume for the instance, and choose **Actions, Create Snapshot**.
3. In the navigation pane, choose **Snapshots**. Select the snapshot that you just created, and choose **Actions, Create Volume**.
4. Launch an instance with the same operating system as the stuck instance. Note the volume ID and device name of its root volume.
5. In the navigation pane, choose **Instances**, select the instance that you just launched, and choose **Instance state, Stop instance**.
6. In the navigation pane, choose **Volumes**, select the root volume of the stopped instance, and choose **Actions, Detach Volume**.
7. Select the root volume that you created from the stuck instance, choose **Actions, Attach Volume**, and attach it to the new instance as its root volume (using the device name that you made note of). Attach any additional non-root volumes to the instance.
8. In the navigation pane, choose **Instances** and select the replacement instance. Choose **Instance state, Start instance**. Verify that the instance is working.
9. Select the stuck instance, choose **Instance state, Terminate instance**. If the instance also gets stuck terminating, Amazon EC2 automatically forces it to terminate within a few hours.

### Troubleshoot instance termination (shutting down)

You are not billed for any instance usage while an instance is not in the **running** state. In other words, when you terminate an instance, you stop incurring charges for that instance as soon as its state changes to **shutting-down**.

#### Instance terminates immediately

Several issues can cause your instance to terminate immediately on start-up. See [Instance terminates immediately (p. 1467)](#) for more information.

#### Delayed instance termination

If your instance remains in the **shutting-down** state longer than a few minutes, it might be delayed due to shutdown scripts being run by the instance.
Another possible cause is a problem with the underlying host computer. If your instance remains in the shutting-down state for several hours, Amazon EC2 treats it as a stuck instance and forcibly terminates it.

If it appears that your instance is stuck terminating and it has been longer than several hours, post a request for help to the Amazon EC2 forum. To help expedite a resolution, include the instance ID and describe the steps that you've already taken. Alternatively, if you have a support plan, create a technical support case in the Support Center.

**Terminated instance still displayed**

After you terminate an instance, it remains visible for a short while before being deleted. The state shows as terminated. If the entry is not deleted after several hours, contact Support.

**Instances automatically launched or terminated**

Generally, the following behaviors mean that you've used Amazon EC2 Auto Scaling, EC2 Fleet, or Spot Fleet to scale your computing resources automatically based on criteria that you've defined:

- You terminate an instance and a new instance launches automatically.
- You launch an instance and one of your instances terminates automatically.
- You stop an instance and it terminates and a new instance launches automatically.

To stop automatic scaling, see the Amazon EC2 Auto Scaling User Guide, EC2 Fleet (p. 664), or Create a Spot Fleet request (p. 725).

**Troubleshoot Sysprep**

If you experience problems or receive error messages during image preparations, review the following logs. Log location varies depending on whether you are running EC2Config or EC2Launch with Sysprep.

- %WINDIR%\Panther\Unattendgc (EC2Config and EC2Launch)
- %WINDIR%\System32\Sysprep\Panther (EC2Config and EC2Launch)
- C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt (EC2Config only)
- C:\ProgramData\Amazon\Ec2Config\Logs (EC2Config only)
- C:\ProgramData\Amazon\EC2-Windows\Launch\Log\EC2Launch.log (EC2Launch only)

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. For more information about the purpose of the log files by name, see Windows Setup-Related Log Files in the Microsoft documentation.

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]
```

1498
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.

---

**Use 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.

The AWSSupport-ExecuteEC2Rescue runbook uses the EC2Rescue tool to troubleshoot and, where possible, fix common connectivity issues with the specified EC2 instance. For more information, and to run this automation, see AWSSupport-ExecuteEC2Rescue.

**Note**

If you are using a Linux instance, see EC2Rescue for Linux.

**Contents**

- Use EC2Rescue for Windows Server GUI (p. 1499)
- Use EC2Rescue for Windows Server with the command line (p. 1503)
- Use EC2Rescue for Windows Server with Systems Manager Run Command (p. 1508)

**Use EC2Rescue for Windows Server GUI**

EC2Rescue for Windows Server can perform the following analysis on **offline instances**:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagnose and Rescue</td>
<td>EC2Rescue for Windows Server can detect and address issues with the following service settings:</td>
</tr>
<tr>
<td></td>
<td>- System Time</td>
</tr>
<tr>
<td></td>
<td>- <strong>RealTimeIsUniversal</strong> - 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.</td>
</tr>
</tbody>
</table>

1499
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Windows Firewall</td>
<td></td>
</tr>
<tr>
<td>• <strong>Domain networks</strong> - Detects whether this Windows Firewall profile is enabled or disabled.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Private networks</strong> - Detects whether this Windows Firewall profile is enabled or disabled.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Guest or public networks</strong> - Detects whether this Windows Firewall profile is enabled or disabled.</td>
<td></td>
</tr>
<tr>
<td>• Remote Desktop</td>
<td></td>
</tr>
<tr>
<td>• <strong>Service Start</strong> - Detects whether the Remote Desktop service is enabled.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Remote Desktop Connections</strong> - Detects whether this is enabled.</td>
<td></td>
</tr>
<tr>
<td>• <strong>TCP Port</strong> - Detects which port the Remote Desktop service is listening on.</td>
<td></td>
</tr>
<tr>
<td>• EC2Config (Windows Server 2012 R2 and earlier)</td>
<td></td>
</tr>
<tr>
<td>• <strong>Installation</strong> - Detects which EC2Config version is installed.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Service Start</strong> - Detects whether the EC2Config service is enabled.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Ec2SetPassword</strong> - Generates a new administrator password.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Ec2HandleUserData</strong> - Allows you to run a user data script on the next boot of the instance.</td>
<td></td>
</tr>
<tr>
<td>• EC2Launch (Windows Server 2016 and later)</td>
<td></td>
</tr>
<tr>
<td>• <strong>Installation</strong> - Detects which EC2Launch version is installed.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Ec2SetPassword</strong> - Generates a new administrator password.</td>
<td></td>
</tr>
<tr>
<td>• Network Interface</td>
<td></td>
</tr>
<tr>
<td>• <strong>DHCP Service Startup</strong> - Detects whether the DHCP service is enabled.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Ethernet detail</strong> - Displays information about the network driver version, if detected.</td>
<td></td>
</tr>
<tr>
<td>• <strong>DHCP on Ethernet</strong> - Detects whether DHCP is enabled.</td>
<td></td>
</tr>
</tbody>
</table>
# Use the GUI

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restore</td>
<td>Perform one of the following actions:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Last Known Good Configuration</strong> - Attempts to boot the instance into the last known bootable state.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Restore registry from backup</strong> - Restores the registry from <code>\Windows\System32\config\RegBack</code>.</td>
</tr>
<tr>
<td>Capture Logs</td>
<td>Allows you to capture logs on the instance for analysis.</td>
</tr>
</tbody>
</table>

EC2Rescue for Windows Server can collect the following data from **active and offline instances**:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Log</td>
<td>Collects application, system, and EC2Config event logs.</td>
</tr>
<tr>
<td>Registry</td>
<td>Collects <code>SYSTEM</code> and <code>SOFTWARE</code> hives.</td>
</tr>
<tr>
<td>Windows Update Log</td>
<td>Collects log files generated by Windows Update.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>In Windows Server 2016 and later, the log is collected in Event Tracing for Windows (ETW) format.</td>
</tr>
<tr>
<td>Sysprep Log</td>
<td>Collects log files generated by the Windows System Preparation tool.</td>
</tr>
<tr>
<td>Driver Setup Log</td>
<td>Collects Windows SetupAPI logs (<code>setupapi.dev.log</code> and <code>setupapi.setup.log</code>).</td>
</tr>
<tr>
<td>Boot Configuration</td>
<td>Collects <code>HKEY_LOCAL_MACHINE\BCD00000000</code> hive.</td>
</tr>
<tr>
<td>Memory Dump</td>
<td>Collects any memory dump files that exist on the instance.</td>
</tr>
<tr>
<td>EC2Config File</td>
<td>Collects log files generated by the EC2Config service.</td>
</tr>
<tr>
<td>EC2Launch File</td>
<td>Collects log files generated by the EC2Launch scripts.</td>
</tr>
<tr>
<td>SSM Agent File</td>
<td>Collects log files generated by SSM Agent and Patch Manager logs.</td>
</tr>
<tr>
<td>EC2 ElasticGPUs File</td>
<td>Collects event logs related to elastic GPUs.</td>
</tr>
<tr>
<td>ECS</td>
<td>Collects logs related to Amazon ECS.</td>
</tr>
<tr>
<td>CloudEndure</td>
<td>Collects log files related to CloudEndure Agent.</td>
</tr>
</tbody>
</table>
EC2Rescue for Windows Server can collect the following additional data from **active instances**:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Information</td>
<td>Collects MSInfo32.</td>
</tr>
<tr>
<td>Group Policy Result</td>
<td>Collects a Group Policy report.</td>
</tr>
</tbody>
</table>

**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?

**Analyze 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.

   You can run the following PowerShell command to download EC2Rescue without changing your Internet Explorer Enhanced Security Configuration (ESC):

   ```powershell
   PS C:\> Invoke-WebRequest https://s3.amazonaws.com/ec2rescue/windows/EC2Rescue_latest.zip -OutFile $env:USERPROFILE\Desktop\EC2Rescue_latest.zip
   ```

   This command will download the EC2Rescue .zip file to the desktop of the currently logged in user.

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.

**Collect 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.

   You can run the following PowerShell command to download EC2Rescue without changing your Internet Explorer Enhanced Security Configuration (ESC):
Use the command line

PS C:\> Invoke-WebRequest https://s3.amazonaws.com/ec2rescue/windows/EC2Rescue_latest.zip -OutFile $env:USERPROFILE\Desktop\EC2Rescue_latest.zip

This command will download the EC2Rescue .zip file to the desktop of the currently logged in user.

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.

Use 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 using the following command:

EC2RescueCmd.exe /help

EC2Rescue for Windows Server can perform the following actions on an Amazon EC2 Windows instance:

- Collect action (p. 1503)
- Rescue action (p. 1505)
- Restore action (p. 1507)

Collect action

Note
You can collect all logs, an entire log group, or an individual log within a group.

EC2Rescue for Windows Server can collect the following data from active and offline instances.

<table>
<thead>
<tr>
<th>Log group</th>
<th>Available logs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td></td>
<td>Collects all available logs.</td>
</tr>
<tr>
<td>eventlog</td>
<td>'Application'</td>
<td>Collects application, system, and EC2Config event logs.</td>
</tr>
</tbody>
</table>
### Log group | Available logs | Description
--- | --- | ---
memory-dump | • 'Memory Dump File' | Collects any memory dump files that exist on the instance.
• 'Mini Dump Files' | | |
ec2config | • 'Log Files' | Collects log files generated by the EC2Config service.
• 'Configuration Files' | | |
ec2launch | • 'Logs' | | |
• 'Config' | Collects log files generated by the EC2Launch scripts.
ssm-agent | • 'Log Files' | Collects log files generated by SSM Agent and Patch Manager logs.
• 'Patch Baseline Logs' | | |
• 'InstanceData' | | |
sysprep | • 'Log Files' | Collects log files generated by the Windows System Preparation tool.
\ | | |
driver-setup | • 'SetupAPI Log Files' | Collects Windows SetupAPI logs (setupapi.dev.log and setupapi.setup.log).
• 'DPInst Log File' | | |
• 'AWS PV Setup Log File' | | |
registry | • 'SYSTEM' | Collects SYSTEM and SOFTWARE hives.
• 'SOFTWARE' | | |
• 'BCD' | | |
egpu | • 'Event Log' | Collects event logs related to elastic GPUs.
• 'System Files' | | |
boot-config | • 'BCDEDIT Output' | Collects HKEY_LOCAL_MACHINE \BCD0000000 hive.
\ | | |
windows-update | • 'Log Files' | Collects log files generated by Windows Update.
\ | | |
\\ | | |
cloudendure | • 'Migrate Script Logs' | Collects log files related to CloudEndure Agent.
• 'Driver Logs' | | |
• 'CloudEndure File List' | | |

EC2Rescue for Windows Server can collect the following additional data from active instances.

### Log group | Available logs | Description
--- | --- | ---
system-info | • 'MSInfo32 Output' | Collects MSInfo32.
Log group | Available logs | Description
---|---|---
gpresult | 'GPResult Output' | Collects a Group Policy report.

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 can detect and address issues with the following service settings:

<table>
<thead>
<tr>
<th>Service group</th>
<th>Available actions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service group</td>
<td>Available actions</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| system-time  | 'RealTimeIsUniversal' | System Time  
- **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. |
| firewall     | 'Domain networks', 'Private networks', 'Guest or public networks' | Windows Firewall  
- **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          | 'Service Start', 'Remote Desktop Connections', 'TCP Port' | 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    | 'Service Start', 'Ec2SetPassword', 'Ec2HandleUserData' | EC2Config  
- **Service Start** - Detects whether the EC2Config service is enabled.  
- **Ec2SetPassword** - Generates a new administrator password.  
- **Ec2HandleUserData** - Allows you to run a user data script on the next boot of the instance. |
| ec2launch    | 'Reset Administrator Password' | Generates a new Windows administrator password. |
Use the command line

## Service group

<table>
<thead>
<tr>
<th>Network</th>
<th>'DHCP Service Startup'</th>
</tr>
</thead>
</table>

*DHCP Service Startup* - Detects whether the DHCP service is enabled.

### 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 can detect and address issues with the following service settings:

<table>
<thead>
<tr>
<th>Service Group</th>
<th>Available Actions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restore Last Known Good Configuration</td>
<td>lkgc</td>
<td>Last Known Good Configuration - Attempts to boot the instance into the last known bootable state.</td>
</tr>
</tbody>
</table>
Use Systems Manager

<table>
<thead>
<tr>
<th>Service Group</th>
<th>Available Actions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restore Windows registry from latest backup</td>
<td>regback</td>
<td>Restore registry from backup - Restores the registry from \Windows\System32\config\RegBack.</td>
</tr>
</tbody>
</table>

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
```

**Use 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:

```json
{
  "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:

```json
{
  "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**

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**.

1509
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 run 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 AWS 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
```
EC2 Serial Console for Windows instances

With the EC2 serial console, you have access to your Amazon EC2 instance's serial port, which you can use to troubleshoot boot, network configuration, and other issues. The serial console does not require your instance to have any networking capabilities. With the serial console, you can enter commands to an instance as if your keyboard and monitor are directly attached to the instance's serial port. The serial console session lasts during instance reboot and stop. During reboot, you can view all of the boot messages from the start.

Access to the serial console is not available by default. Your organization must grant account access to the serial console and configure IAM policies to grant your users access to the serial console. Serial console access can be controlled at a granular level by using instance IDs, resource tags, and other IAM levers. For more information, see Configure access to the EC2 Serial Console (p. 1511).

The serial console can be accessed by using the EC2 console or the AWS CLI.

The serial console is available at no additional cost.

If you are using a Linux instance, see EC2 Serial Console for Linux instances in the Amazon EC2 User Guide for Linux Instances.

Topics

- Configure access to the EC2 Serial Console (p. 1511)
- Connect to the EC2 Serial Console (p. 1516)
- Terminate an EC2 Serial Console session (p. 1520)
- Troubleshoot your Windows instance using the EC2 Serial Console (p. 1521)

Configure access to the EC2 Serial Console

To configure access to the serial console, you must grant serial console access at the account level and then configure IAM policies to grant access to your IAM users.

Topics

- Levels of access to the EC2 Serial Console (p. 1511)
- Manage account access to the EC2 Serial Console (p. 1512)
- Configure IAM policies for EC2 Serial Console access (p. 1514)

Levels of access to the EC2 Serial Console

By default, there is no access to the serial console at the account level. You need to explicitly grant access to the serial console at the account level. For more information, see Manage account access to the EC2 Serial Console (p. 1512).

You can use a service control policy (SCP) to allow access to the serial console within your organization. You can then have granular access control at the IAM user level by using an IAM policy to control access. By using a combination of SCP and IAM policies, you have different levels of access control to the serial console.
Organization level

You can use a service control policy (SCP) to allow access to the serial console for member accounts within your organization. For more information about SCPs, see Service control policies in the AWS Organizations User Guide.

Instance level

You can configure the serial console access policies by using IAM PrincipalTag and ResourceTag constructions and by specifying instances by their ID. For more information, see Configure IAM policies for EC2 Serial Console access (p. 1514).

IAM user level

You can configure access at the user level by configuring an IAM policy to allow or deny a specified user the permission to push the SSH public key to the serial console service of a particular instance. For more information, see Configure IAM policies for EC2 Serial Console access (p. 1514).

Manage account access to the EC2 Serial Console

By default, there is no access to the serial console at the account level. You need to explicitly grant access to the serial console at the account level.

Topics

- Grant permission to IAM users to manage account access (p. 1512)
- View account access status to the serial console (p. 1512)
- Grant account access to the serial console (p. 1513)
- Deny account access to the serial console (p. 1514)

Grant permission to IAM users to manage account access

To allow your IAM users to manage account access to the EC2 serial console, you need to grant them the required IAM permissions.

The following policy grants permissions to view the account status, and to allow and prevent account access to the EC2 serial console.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "ec2:GetSerialConsoleAccessStatus",
            "ec2:EnableSerialConsoleAccess",
            "ec2:DisableSerialConsoleAccess"
         ],
         "Resource": "*"
      }
   ]
}
```

For more information, see Creating IAM policies in the IAM User Guide.

View account access status to the serial console

To view account access status to the serial console (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the left navigation pane, choose **EC2 Dashboard**.
3. From **Account attributes**, choose **EC2 Serial Console**.

The **EC2 Serial Console access** field indicates whether account access is **Allowed** or **Prevented**.

The following screenshot shows that the account is prevented from using the EC2 serial console.

![EC2 Serial Console access screenshot](image)

**To view account access status to the serial console (AWS CLI)**

Use the `get-serial-console-access-status` command to view account access status to the serial console.

```bash
aws ec2 get-serial-console-access-status --region us-east-1
```

In the following output, `true` indicates that the account is allowed access to the serial console.

```
{
  "SerialConsoleAccessEnabled": true
}
```

**Grant account access to the serial console**

**To grant account access to the serial console (console)**

1. Open the Amazon EC2 console at [https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/).
2. On the left navigation pane, choose **EC2 Dashboard**.
3. From **Account attributes**, choose **EC2 Serial Console**.
4. Choose **Manage**.
5. To allow access to the EC2 serial console of all instances in the account, select the **Allow** check box.
6. Choose **Update**.

**To grant account access to the serial console (AWS CLI)**

Use the `enable-serial-console-access` command to allow account access to the serial console.

```bash
aws ec2 enable-serial-console-access --region us-east-1
```

In the following output, `true` indicates that the account is allowed access to the serial console.

```
{
  "SerialConsoleAccessEnabled": true
}
```
Deny account access to the serial console

To deny account access to the serial console (console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the left navigation pane, choose **EC2 Dashboard**.
3. From **Account attributes**, choose **EC2 Serial Console**.
4. Choose **Manage**.
5. To prevent access to the EC2 serial console of all instances in the account, clear the **Allow** check box.
6. Choose **Update**.

To deny account access to the serial console (AWS CLI)

Use the `disable-serial-console-access` command to prevent account access to the serial console.

```
aws ec2 disable-serial-console-access --region us-east-1
```

In the following output, `false` indicates that the account is denied access to the serial console.

```
{
    "SerialConsoleAccessEnabled": false
}
```

Configure IAM policies for EC2 Serial Console access

By default, your IAM users do not have access to the serial console. Your organization must configure IAM policies to grant your IAM users the required access. For more information, see Creating IAM policies in the IAM User Guide.

For serial console access, create a JSON policy document that includes the `ec2-instance-connect:SendSerialConsoleSSHPublicKey` action. This action grants an IAM user permission to push the public key to the serial console service, which starts a serial console session. We recommend restricting access to specific EC2 instances. Otherwise, all IAM users with this permission can connect to the serial console of all EC2 instances.

**Example IAM policies**

- Explicitly allow access to the serial console (p. 1514)
- Explicitly deny access to the serial console (p. 1515)
- Use resource tags to control access to the serial console (p. 1515)

**Explicitly allow access to the serial console**

By default, no one has access to the serial console. To grant access to the serial console, you need to configure a policy to explicitly allow access. We recommend configuring a policy that restricts access to specific instances.

The following policy allows access to the serial console of a specific instance, identified by its instance ID.

```json
{
    "Version": "2012-10-17",
    "Statement": [ 
```
Explicitly deny access to the serial console

The following IAM policy allows access to the serial console of all instances, denoted by the * (asterisk), and explicitly denies access to the serial console of a specific instance, identified by its ID.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DenySerialConsoleAccess",
            "Effect": "Deny",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "arn:aws:ec2:region:account-id:instance/i-0598c7d356eboa48d7"
        }
    ]
}
```

Use resource tags to control access to the serial console

You can use resource tags to control access to the serial console of an instance.

Attribute-based access control is an authorization strategy that defines permissions based on tags that can be attached to users and AWS resources. For example, the following policy allows an IAM user to initiate a serial console connection for an instance only if that instance's resource tag and the principal's tag have the same SerialConsole value for the tag key.

For more information about using tags to control access to your AWS resources, see Controlling access to AWS resources in the IAM User Guide.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowTagBasedSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "*",
            "Condition": {
            
            }
        }
    ]
}
```
Connect to the EC2 Serial Console

You can connect to the serial console of your EC2 instance by using the Amazon EC2 console or via SSH. After connecting to the serial console, you can use it for troubleshooting boot, network configuration, and other issues. For more information about troubleshooting, see Troubleshoot your Windows instance using the EC2 Serial Console (p. 1521).

Topics

• Considerations (p. 1516)
• Prerequisites (p. 1516)
• Connect to the EC2 Serial Console (p. 1517)
• EC2 Serial Console fingerprints (p. 1519)

Considerations

• Only one active serial console connection is supported per instance.
• The serial console connection typically lasts for one hour unless you terminate it. However, during system maintenance, Amazon EC2 will terminate the serial console session.
• It takes 30 seconds to tear down a session after you've disconnected from the serial console in order to allow a new session.
• Supported serial console port for Windows: COM1
• When you connect to the serial console, you might observe a slight drop in your instance's throughput.

Prerequisites

• Supported AWS Regions: US East (N. Virginia), US East (Ohio), US West (Oregon) Europe (Ireland), Europe (Frankfurt), Asia Pacific (Sydney), Asia Pacific (Tokyo), Asia Pacific (Singapore)
• Supported instance families:
  • A1
  • C5, C5a, C5ad, C5d, C5n, C6g, C6gd
  • M5, M5a, M5ad, M5d, M5dn, M5n, M5zn, M6g, M6gd
  • R5, R5a, R5ad, R5d, R5dn, R5n, R6, R6gd
  • T3, T3a, T4g
  • Z1d
• Configure access to the EC2 Serial Console, as follows:
  • Manage account access to the EC2 Serial Console (p. 1512).
  • Configure IAM policies for EC2 Serial Console access (p. 1514). All IAM users who will use the serial console must have the required permissions.
• To connect to the serial console using the browser-based client (p. 1517), your browser must support WebSocket. If your browser does not support WebSocket, connect to the serial console using your own key and an SSH client. (p. 1517)
Connect to the EC2 Serial Console

Connection options

- Connect using the browser-based client (p. 1517)
- Connect using your own key and SSH client (p. 1517)

Connect using the browser-based client

You can connect to your EC2 instance’s serial console by using the browser-based client. You do this by selecting the instance in the Amazon EC2 console and choosing to connect to the serial console. The browser-based client handles the permissions and provides a successful connection.

EC2 serial console works from most browsers, and supports keyboard and mouse input.

To connect to your instance’s serial port using the browser-based client (Amazon EC2 console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Instances.
3. Select the instance and choose Connect, EC2 Serial Console, Connect. Alternatively, you can select the instance and choose Actions, Monitor and troubleshoot, EC2 Serial Console, Connect.

   An in-browser terminal window opens.
4. Press Enter. If a login prompt returns, you are connected to the serial console.

   If the screen remains black, you can use the following information to help resolve issues with connecting to the serial console:
   - Check that you have configured access to the serial console. For more information, see Configure access to the EC2 Serial Console (p. 1511).
   - Reboot your instance. You can reboot your instance by using the EC2 console or the AWS CLI. For more information, see Reboot your instance (p. 438).

Connect using your own key and SSH client

You can use your own SSH key and connect to your instance from the SSH client of your choice while using the serial console API. This enables you to benefit from the serial console capability to push a public key to the instance.

To connect to an instance’s serial console using SSH

1. Push your SSH public key to the instance to start a serial console session
Use the `send-serial-console-ssh-public-key` command to push your SSH public key to the instance. This starts a serial console session.

If a serial console session has already been started for this instance, the command fails because you can only have one session open at a time. It takes 30 seconds to tear down a session after you've disconnected from the serial console in order to allow a new session.

```
C:\> aws ec2-instance-connect send-serial-console-ssh-public-key \
    --instance-id i-001234a4bf70dec41EXAMPLE \
    --serial-port 0 \
    --ssh-public-key file://my_rsa_key.pub \
    --region us-east-1
```

2. **Connect to the serial console using your private key**

Use the `ssh` command to connect to the serial console before the public key is removed from the serial console service. You have 60 seconds before it is removed.

Use the private key that corresponds to the public key.

The user name format is `instance-id.port0`, which comprises the instance ID and port 0. In the following example, the user name is `i-001234a4bf70dec41EXAMPLE.port0`.

**For all supported AWS Regions, except AWS GovCloud (US) Regions:**

The format of the public DNS name of the serial console service is `serial-console.ec2-instance-connect.region.aws`. In the following example, the serial console service is in the `us-east-1` Region.

```
C:\> ssh -i my_rsa_key i-001234a4bf70dec41EXAMPLE.port0@serial-console.ec2-instance-connect.us-east-1.aws
```

**For AWS GovCloud (US) Regions only:**

The format of the public DNS name of the serial console service in the AWS GovCloud (US) Regions is `serial-console.ec2-instance-connect.GovCloud-region.amazonaws.com`. In the following example, the serial console service is in the `us-gov-east-1` Region.

```
C:\> ssh -i my_rsa_key i-001234a4bf70dec41EXAMPLE.port0@serial-console.ec2-instance-connect.us-gov-east-1.amazonaws.com
```

3. **(Optional) Verify the fingerprint**

When you connect for the first time to the serial console, you are prompted to verify the fingerprint. You can compare the serial console fingerprint with the fingerprint that's displayed for verification. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, you can confidently connect to the serial console.

The following fingerprint is for the serial console service in the `us-east-1` Region. For the fingerprints for each Region, see EC2 Serial Console fingerprints (p. 1519).

```
SHA256:dXwn5ma/xadVMeBZGEru5l2gx+yI5LDiJaLUCz0FMnw
```

**Note**

The fingerprint only appears the first time you connect to the serial console.

4. **Press Enter.** If a prompt returns, you are connected to the serial console.
If the screen remains black, you can use the following information to help resolve issues with connecting to the serial console:

- **Check that you have configured access to the serial console.** For more information, see [Configure access to the EC2 Serial Console](p. 1511).
- **Reboot your instance.** You can reboot your instance by using the EC2 console or the AWS CLI. For more information, see [Reboot your instance](p. 438).

## EC2 Serial Console Fingerprints

The EC2 Serial Console fingerprint is unique for each AWS Region.

- **us-east-2** – US East (Ohio)
  
  SHA256:EhwFkTzRTtY7TRSzz26XbB0/HVv9jRM7mCZm0xw/d/0

- **us-east-1** – US East (N. Virginia)
  
  SHA256:dXwn5ma/xadVMebZGERu5l2gx+yI5LDiJaLUCz0FMmw

- **us-west-1** – US West (N. California)

  SHA256:OHldlcMET8u7QLSx3jmRTRAPFHVtqbyoLZBMUCqiH3Y

- **us-west-2** – US West (Oregon)

  SHA256:EMCIe23TqKaBI6yGHaingZcMwqNKdhhAVHaIo2JxVUC

- **ap-south-1** – Asia Pacific (Mumbai)

  SHA256:oBLXcYmplqHHEbliARxEgH8IsO51rezTPiSM35BsU40

- **ap-northeast-2** – Asia Pacific (Seoul)

  SHA256:FoqWXNX+DZ++GuNTztg9FK49WYmqBX+FrcZM2sRqrI

- **ap-southeast-1** – Asia Pacific (Singapore)

  SHA256:PLFNn7WnCQDHx3qwwLulGy/O0TUX7LQgZuaC6L45CoY

- **ap-southeast-2** – Asia Pacific (Sydney)

  SHA256:yFvMwUK91EUqjQTRoXXzuN+cW9/VSe9W984CF5Tgzo4

- **ap-northeast-1** – Asia Pacific (Tokyo)

  SHA256:RQfsDCZTOfQawetRvdV1t9Em/NMrFQe+CR1IOT5um4k

- **ca-central-1** – Canada (Central)

  SHA256:P2O2jOZwmpMwkpO6YW738FIOTHdUTyEv2gzYYMO78a

- **eu-central-1** – Europe (Frankfurt)
Amazon Elastic Compute Cloud  
User Guide for Windows Instances  
Terminate an EC2 Serial Console session

The way to terminate a serial console session depends on the client.

**Browser-based client**

To terminate the serial console session, close the serial console in-browser terminal window.

**Standard OpenSSH client**

To terminate the serial console session, use the following command to close the SSH connection. This command must be run immediately following a new line.

```
C:\> ~.
```

**Note**

The command that you use for closing an SSH connection might be different depending on the SSH client that you’re using.
Troubleshoot your Windows instance using the EC2 Serial Console

By using EC2 Serial Console, you can troubleshoot boot, network configuration, and other issues by connecting to your instance's serial port.

**Topics**
- Use SAC to troubleshoot your Windows instance (p. 1521)

For information about troubleshooting your Linux instance, see Troubleshoot your Linux instance using the EC2 Serial Console in the Amazon EC2 User Guide for Linux Instances.

Use SAC to troubleshoot your Windows instance

The Special Admin Console (SAC) capability of Windows provides a way to troubleshoot a Windows instance. By connecting to the instance's serial console and using SAC, you can interrupt the boot process and boot Windows in safe mode.

**Topics**
- Limitations (p. 1521)
- Prerequisites (p. 1521)
- Use SAC (p. 1522)
- Use the boot menu (p. 1524)

**Limitations**

If you launch an instance with an AMI that comes preconfigured with SAC, the EC2 services that rely on password retrieval will not work from the console.

**Prerequisites**

To use SAC for troubleshooting a Windows instance, you must first complete the following prerequisites:

1. Grant access to the serial console. For more information, see Configure access to the EC2 Serial Console (p. 1511).
2. Enable SAC and the boot menu. For more information, see Enable SAC and the boot menu (p. 1521).
3. Connect to the serial console. For more information, see Connect to the EC2 Serial Console (p. 1516).

**Enable SAC and the boot menu**

Use one of the following methods to enable SAC and the boot menu on an instance.

**PowerShell**

To enable SAC and the boot menu on a Windows instance

1. Connect (p. 413) to your instance and perform the following steps from an elevated PowerShell command line.
2. Enable SAC.
Troubleshoot your instance using the EC2 Serial Console

3. Enable the boot menu.

    bcdedit /set '{bootmgr}' displaybootmenu yes
    bcdedit /set '{bootmgr}' timeout 15
    bcdedit /set '{bootmgr}' bootems yes

4. Apply the updated configuration by rebooting the instance.

    shutdown -r -t 0

Command prompt

To enable SAC and the boot menu on a Windows instance

1. Connect (p. 413) to your instance and perform the following steps from the command prompt.
2. Enable SAC.

    bcdedit /ems '{current}' on
    bcdedit /emssettings EMSPORT:1 EMSBAUDRATE:115200

3. Enable the boot menu.

    bcdedit /set '{bootmgr}' displaybootmenu yes
    bcdedit /set '{bootmgr}' timeout 15
    bcdedit /set '{bootmgr}' bootems yes

4. Apply the updated configuration by rebooting the instance.

    shutdown -r -t 0

Use SAC

To use SAC

1. Connect to the serial console. (p. 1516)

   If SAC is enabled on the instance, the serial console displays the SAC> prompt.

2. To display the SAC commands, enter ?, and then press Enter.

   Expected output
3. To create a command prompt channel (such as cmd0001 or cmd0002), enter `cmd`, and then press Enter.

4. To view the command prompt channel, press ESC, and then press TAB.

Expected output

To switch channels, press ESC+TAB+channel number together. For example, to switch to the cmd0002 channel (if it has been created), press ESC+TAB+2.

6. Enter the credentials required by the command prompt channel.

The command prompt is the same full-featured command shell that you get on a desktop, but with the exception that it does not allow the reading of characters that were already output.

PowerShell can also be used from the command prompt.

Note that you might need to set the progress preference to silent mode.
Use the boot menu

If the instance has the boot menu enabled and is restarted after connecting via SSH, you should see the boot menu, as follows.

**Boot menu commands**

**ENTER**

Starts the selected entry of the operating system.

**TAB**

Switches to the Tools menu.

**ESC**

Cancels and restarts the instance.

**ESC followed by 8**

Equivalent to pressing F8. Shows advanced options for the selected item.

**ESC key + left arrow**

Goes back to the initial boot menu.

**Note**

The ESC key alone does not take you back to the main menu because Windows is waiting to see if an escape sequence is in progress.
Send a diagnostic interrupt (for advanced users)

Warning
Diagnostic interrupts are intended for use by advanced users. Incorrect usage could negatively impact your instance. Sending a diagnostic interrupt to an instance could trigger an instance to crash and reboot, which could lead to the loss of data.

You can send a diagnostic interrupt to an unreachable or unresponsive Windows instance to manually trigger a stop error. Stop errors are commonly referred to as blue screen errors.

In general, Windows operating systems crash and reboot when a stop error occurs, but the specific behavior depends on its configuration. A stop error can also cause the operating system to write debugging information, such as a kernel memory dump, to a file. You can then use this information to conduct root cause analysis to debug the instance.

The memory dump data is generated locally by the operating system on the instance itself.

Before sending a diagnostic interrupt to your instance, we recommend that you consult the documentation for your operating system and then make the necessary configuration changes.

Contents
- Supported instance types (p. 1525)
- Prerequisites (p. 1525)
- Send a diagnostic interrupt (p. 1526)

Supported instance types
Diagnostic interrupt is supported on all Nitro-based instance types, except A1. For more information, see Instances built on the Nitro System (p. 146).

Prerequisites
Before using a diagnostic interrupt, you should configure your instance's operating system to perform the actions you need when a stop error occurs.
To configure Windows to generate a memory dump when a stop error occurs

1. Connect to your instance.
2. Open the Control Panel and choose System, Advanced system settings.
3. In the System Properties dialog box, choose the Advanced tab.
4. In the Startup and Recovery section, choose Settings....
5. In the System failure section, configure the settings as needed, and then choose OK.

For more information about configuring Windows stop errors, see Overview of memory dump file options for Windows.

Send a diagnostic interrupt

After you have completed the necessary configuration changes, you can send a diagnostic interrupt to your instance using the AWS CLI or Amazon EC2 API.

To send a diagnostic interrupt to your instance (AWS CLI)

Use the send-diagnostic-interrupt command and specify the instance ID.

```
aws ec2 send-diagnostic-interrupt --instance-id i-1234567890abcdef0
```

To send a diagnostic interrupt to your instance (AWS Tools for Windows PowerShell)

Use the Send-EC2DiagnosticInterrupt cmdlet and specify the instance ID.

```
PS C:\> Send-EC2DiagnosticInterrupt-InstanceId i-1234567890abcdef0
```

Common issues with Windows instances

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 and later (p. 1526)
- Boot an EC2 Windows instance into Directory Services Restore Mode (DSRM) (p. 1527)
- Instance loses network connectivity or scheduled tasks don't run when expected (p. 1529)
- Unable to get console output (p. 1529)
- Windows Server 2012 R2 not available on the network (p. 1530)

EBS volumes don't initialize on Windows Server 2016 and later

Instances created from Amazon Machine Images (AMIs) for Windows Server 2016 and later 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:

```json
{
    "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:

```bash
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:

```bash
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 table shows which drivers are supported for DSRM.

<table>
<thead>
<tr>
<th>Driver Versions</th>
<th>DSRM Supported?</th>
<th>Next Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Citrix PV 5.9</td>
<td>No</td>
<td>Restore the instance from a backup. You cannot enable DSRM.</td>
</tr>
<tr>
<td>AWS PV 7.2.0</td>
<td>No</td>
<td>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).</td>
</tr>
<tr>
<td>AWS PV 7.2.2 and later</td>
<td>Yes</td>
<td>Detach the root volume, attach it to another instance, and enable DSRM (as described in this section).</td>
</tr>
</tbody>
</table>
Driver Versions | DSRM Supported? | Next Steps
--- | --- | ---
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 Upgrade PV drivers on Windows instances (p. 524).

**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.
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:

```
bcdedit /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 and select the affected instance. Choose Instance state, Stop instance.
4. Choose Launch instances 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, 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:
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 `RealTimeIsUniversal` 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 Upgrade PV drivers on Windows instances (p. 524).
2. Verify that the following registry key exists and is set to 1: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal`

Unable to get console output

For Windows instances, the instance console displays the output from 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. 497).

To get the console output for your instance using the Amazon EC2 console, select the instance, and then choose Actions, Monitor and troubleshoot, 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).

For instances running Windows Server 2012 R2 and earlier, 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 Install the latest version of EC2Config (p. 494).
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. 530).

Common messages troubleshooting Windows instances

This section includes tips to help you troubleshoot issues based on common messages.

Topics

- "Password is not available" (p. 1530)
- "Password not available yet" (p. 1531)
- "Cannot retrieve Windows password" (p. 1531)
- "Waiting for the metadata service" (p. 1531)
- "Unable to activate Windows" (p. 1534)
- "Windows is not genuine (0x80070005)" (p. 1535)
- "No Terminal Server License Servers available to provide a license" (p. 1535)
- "Some settings are managed by your organization" (p. 1536)

"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 Reset a lost or expired Windows administrator password (p. 1484).
"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 Unable to get console output (p. 1529).

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. 579).

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. 948) and attaching it to the instance (p. 950).
- **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 Install the latest version of EC2Config (p. 494).
   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 ...
```

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:

```
```

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 Install the latest version of EC2Config (p. 494).
- Check whether the Windows instance is running RedHat PV drivers. If so, update to Citrix PV drivers. For more information, see Upgrade PV drivers on Windows instances (p. 524).
"Unable to activate Windows"

Windows instances use Windows AWS 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 AWS KMS server. Windows must be activated every 180 days. EC2Config attempts to contact the AWS 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 Install the latest version of EC2Config (p. 494).
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 AWS 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. From a PowerShell prompt with administrative rights, import the EC2Launch module:

   ```powershell
   PS C:\> Import-Module "C:\ProgramData\Amazon\EC2-Windows\Launch\Module\Ec2Launch.psd1"
   ```

2. Call the Add-Routes function to see the list of new routes:

   ```powershell
   PS C:\> Add-Routes
   ```

3. Call the Set-ActivationSettings function:

   ```powershell
   PS C:\> Set-ActivationSettings
   ```

4. Then, run the following script to activate Windows:

   ```powershell
   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 AWS 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 AWS KMS servers are listed.

  \begin{verbatim}
  route print
  \end{verbatim}

- Verify that the AWS KMS client key is set. Run the following command and check the output.

  \begin{verbatim}
  C:\Windows\System32\slmgr.vbs /dlv
  \end{verbatim}

  If the output contains Error: product key not found, the AWS KMS client key isn't set. If the AWS KMS client key isn't set, look up the client key as described in this Microsoft article: AWS KMS Client Setup Keys, and then run the following command to set the AWS KMS client key.

  \begin{verbatim}
  C:\Windows\System32\slmgr.vbs /ipk client_key
  \end{verbatim}

- 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.

  \begin{verbatim}
  netsh advfirewall set allprofiles state off
  \end{verbatim}

"Windows is not genuine (0x80070005)"

Windows instances use Windows AWS 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. 1534).

"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 \texttt{/admin} parameter, for example:
For more information, see the following Microsoft article: [Access Remote Desktop Via Command Line](https://docs.microsoft.com/en-us/windows-server/administration/remote-administration/access-desktop-quickly).

- Stop the instance, detach its Amazon EBS volumes, and attach them to another instance in the same Availability Zone to recover your data.

"Some settings are managed by your organization"

Instances launched from the latest Windows Server AMIs might show a Windows Update dialog message stating "Some settings are managed by your organization." This message appears as a result of changes in Windows Server and does not impact the behavior of Windows Update or your ability to manage update settings.

**To remove the warning**

1. Open `gpedit.msc` and navigate to **Computer Configuration, Administrative Templates, Windows Components, Windows updates**. Edit **Configure Automatic Update**, and set it to **enabled**.
2. In a command prompt, update group policy using `gpupdate /force`.
3. Close and reopen the Windows Update Settings. You will see the above message about your settings being managed by your organization, followed by "We'll automatically download updates, except on metered connections (where charges may apply). In that case, we'll automatically download those updates required to keep Windows running smoothly."
4. Return to `gpedit.msc` and set the group policy back to **not configured**. Run `gpupdate /force` again.
5. Close the command prompt and wait a few minutes.
6. Reopen the Windows Update Settings. You should not see the message "Some settings are managed by your organization."
AWS Systems Manager for Microsoft System Center VMM

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

Get started

To get started, see the following documentation:

- Setting Up (p. 1538)
- Managing EC2 Instances (p. 1542)
- Troubleshooting (p. 1548)

Set 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. 1538)
- Set up access for users (p. 1539)
- Deploy the add-in (p. 1541)
- Provide your AWS credentials (p. 1541)

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

2. Follow the online instructions.

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

The first time that you use 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.
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

```json
{
  "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

```json
{
  "Version": "2012-10-17",
  "Statement": [ 
    {
      "Effect": "Allow",
      "Action": [ 
        "iam:ListInstanceProfiles", "iam:PassRole",
        "ec2:Describe*", "ec2:CreateKeyPair",
```
Set up access for users

```
"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", "s3:GetObject",
"s3:ListBucket", "s3:PutObject",
"ec2:DescribeTags", "ec2:CancelConversionTask",
"ec2:DescribeConversionTasks", "ec2:DescribeInstanceAttribute",
"ec2:CreateImage", "ec2:AttachVolume",
"ec2:ImportInstance", "ec2:ImportVolume",
"dynamodb:DescribeTable", "dynamodb:CreateTable",
"dynamodb:Scan", "dynamodb:PutItem", "dynamodb:UpdateItem"
],
"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 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.
Manage EC2 instances using AWS Systems Manager for Microsoft SCVMM

After you log in to the Systems Manager console using your AWS credentials, you can manage your EC2 instances.

Tasks
- Create an EC2 Instance (p. 1542)
- View your instances (p. 1544)
- Connect to your instance (p. 1544)
- Reboot your instance (p. 1545)
- Stop your instance (p. 1545)
- Start your instance (p. 1545)
- Terminate your instance (p. 1546)

Create 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
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 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. 1186)
- Associate an Elastic IP address with your instance (p. 930)
- Enable termination protection (p. 443)

**View 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 the AWS Systems Manager console.
2. From the list of Regions, 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.

<table>
<thead>
<tr>
<th>i-343ef95a (my-instance)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Virtual machine information</th>
<th>Networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance ID: i-343ef95a</td>
<td>Public DNS name:</td>
</tr>
<tr>
<td>Name: my-instance</td>
<td>Public IP address:</td>
</tr>
<tr>
<td>State: Running</td>
<td>Private DNS name: ip-10-0-0-147.us-west-2.compute.internal</td>
</tr>
<tr>
<td>Launch time: 1/20/2015 12:26:48 PM -0800 (1 minute ago)</td>
<td>Private IP address: 10.0.0.147</td>
</tr>
<tr>
<td>Instance type: m3.medium</td>
<td>Vpc ID: vpc-f1653d98</td>
</tr>
<tr>
<td>Tenancy: default</td>
<td>Subnet ID: subnet-c9663da0</td>
</tr>
<tr>
<td>Image ID: ami-29d18719</td>
<td>Network interfaces: eni-88b0b9ed0</td>
</tr>
<tr>
<td>Operating system: Windows</td>
<td></td>
</tr>
</tbody>
</table>

**Connect 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 Troubleshoot 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.

Reboot 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.

Stop 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.

Start 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](https://aws.amazon.com/ec2/instance-quotas/).

## Terminate 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**.

## Import 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. 1546)
- Import your virtual machine (p. 1547)
- Check the import task status (p. 1548)
- Back up your imported instance (p. 1548)

### Prerequisites

- Ensure that your VM is ready. For more information, see Prepare Your VM in the [VM Import/Export User Guide](https://aws.amazon.com/ec2/instance-quotas/).
- 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](https://docs.microsoft.com/en-us/systemcenter/virtualization-manager/installation/installation) in the Microsoft TechNet Library.
Import 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**.

### Check the import task status

The import task can take several hours to complete. To view the current status, open AWS Systems 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 virtual machine](p. 1549).

### Back 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 [Create a custom Windows AMI](p. 37).

### Troubleshoot AWS Systems Manager for Microsoft SCVMM

The following are common errors and troubleshooting steps.

#### Contents

- Error: Add-in cannot be installed (p. 1548)
- Installation errors (p. 1549)
- Check the log file (p. 1549)
- Errors importing a virtual machine (p. 1549)
- Uninstall the add-in (p. 1550)

#### 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](p. 1549). 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 by the 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 run 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.

**Check 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 virtual machine**

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.

**Uninstall 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

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. 1551)
- Overview of AWS Management Pack for System Center 2007 R2 (p. 1553)
- Download the AWS Management Pack (p. 1554)
- Deploy the AWS Management Pack (p. 1555)
- Use the AWS Management Pack (p. 1565)
- Upgrade the AWS Management Pack (p. 1585)
- Uninstall the AWS Management Pack (p. 1586)
- Troubleshoot the AWS Management Pack (p. 1587)

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.

<table>
<thead>
<tr>
<th>Item</th>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Operations Manager infrastructure</td>
<td>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.</td>
</tr>
<tr>
<td>2</td>
<td>Resource pool</td>
<td>One or more management servers used for communicating with AWS using the AWS SDK for .NET. These servers must have Internet connectivity.</td>
</tr>
<tr>
<td>3</td>
<td>AWS credentials</td>
<td>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 IAM User Guide.</td>
</tr>
<tr>
<td>4</td>
<td>EC2 instances</td>
<td>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.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Item</th>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Operations Manager infrastructure</td>
<td>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.</td>
</tr>
<tr>
<td>2</td>
<td>Watcher node</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Item | Component | Description
--- | --- | ---
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 privileges and use those credentials. For more information about creating an IAM user, see Adding a New User to Your AWS Account in the IAM User Guide.

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.

---

**Download 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 Deploy the AWS Management Pack (p. 1555).
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 Deploy the AWS Management Pack (p. 1555).

**Deploy the AWS Management Pack**

Before you can deploy the AWS Management Pack, you must download it. For more information, see Download the AWS Management Pack (p. 1554).

**Tasks**

- Step 1: Install the AWS Management Pack (p. 1555)
- Step 2: Configure the watcher node (p. 1557)
- Step 3: Create an AWS Run As account (p. 1557)
- Step 4: Run the Add Monitoring wizard (p. 1561)
- Step 5: Configure ports and endpoints (p. 1565)

**Step 1: Install 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.
Step 2: Configure 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.

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.
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.
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:

   a. Click Administration on the Go menu.
   b. Click the Resource Pools node.
   e. On the Pool Membership page, remove the management servers that should not monitor AWS resources.
Step 4: Run the Add Monitoring wizard

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
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.

Use the AWS Management Pack

You can use the AWS Management Pack to monitor the health of your AWS resources.

Contents

- Views (p. 1566)
- Discoveries (p. 1580)
- Monitors (p. 1581)
- Rules (p. 1582)
- Events (p. 1582)
- Health model (p. 1583)
- Customize the AWS Management Pack (p. 1585)
Views

The AWS Management Pack provides the following views, which are displayed in the Monitoring workspace of the Operations console.

Views

- EC2 Instances (p. 1566)
- Amazon EBS Volumes (p. 1568)
- Elastic Load Balancers (p. 1570)
- AWS Elastic Beanstalk applications (p. 1572)
- AWS CloudFormation stacks (p. 1574)
- Amazon performance views (p. 1576)
- Amazon CloudWatch metric alarms (p. 1577)
- AWS alerts (p. 1578)
- Watcher nodes (System Center Operations Manager 2007 R2) (p. 1579)

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.
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 EBS Volumes**

Shows the health state of all the Amazon EBS volumes for a particular AWS account from all Availability Zones and regions.
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.
Elastic Load Balancing Diagram View

Shows the Elastic Load Balancing relationship with other components. The following illustration shows an example:
AWS Elastic Beanstalk applications

Shows the state of all discovered AWS Elastic Beanstalk applications.
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.
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 AWS Services That Publish CloudWatch Metrics in the Amazon CloudWatch User Guide. The following illustration shows an example.
Amazon CloudWatch metric alarms

Shows Amazon CloudWatch alarms related to the discovered AWS resources.
**AWS alerts**

Shows the alerts that the AWS management pack produces when the health of an object is in a critical state.
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).

<table>
<thead>
<tr>
<th>Discovery</th>
<th>Interval (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Resources Discovery (SCOM 2012)</td>
<td>14400</td>
</tr>
<tr>
<td>Discovers EC2 instances, Amazon EBS volumes, load balancers, and CloudFront stacks.</td>
<td></td>
</tr>
<tr>
<td>AWS Elastic Beanstalk Discovery</td>
<td>14400</td>
</tr>
<tr>
<td>Discovers AWS Elastic Beanstalk and its relationship with environment, resources, and configuration.</td>
<td></td>
</tr>
<tr>
<td>CloudWatch Alarms Discovery</td>
<td>900</td>
</tr>
<tr>
<td>Discovers alarms generated using CloudWatch metrics.</td>
<td></td>
</tr>
<tr>
<td>Custom CloudWatch Metric Discovery</td>
<td>14400</td>
</tr>
<tr>
<td>Discovers custom CloudWatch metrics.</td>
<td></td>
</tr>
<tr>
<td>Watcher Node Discovery (SCOM 2007 R2)</td>
<td>14400</td>
</tr>
<tr>
<td>Targets the root management server and creates the watcher node objects.</td>
<td></td>
</tr>
</tbody>
</table>

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).
Rules

Rules create alerts (based on Amazon CloudWatch metrics) and collect data for analysis and reporting.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Interval (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Resource Discovery Rule (SCOM 2007 R2)</td>
<td>14400</td>
</tr>
<tr>
<td>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 <strong>Not Monitored</strong> state.</td>
<td></td>
</tr>
<tr>
<td>Amazon Elastic Block Store Volume Performance Metrics Data Collection Rule</td>
<td>900</td>
</tr>
<tr>
<td>Amazon EC2 Instance Performance Metrics Data Collection Rule</td>
<td>900</td>
</tr>
<tr>
<td>Elastic Load Balancing Balancing Performance Metrics Data Collection Rule</td>
<td>900</td>
</tr>
<tr>
<td>Custom CloudWatch Metric Data Collection Rule</td>
<td>900</td>
</tr>
</tbody>
</table>

Events

Events report on activities that involve the monitored resources. Events are written to the Operations Manager event log.

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4101</td>
<td>Amazon EC2 Instance Discovery (General Discovery) finished</td>
</tr>
<tr>
<td>4102</td>
<td>Elastic Load Balancing Metrics Discovery, Amazon EBS Volume Metrics Discovery, Amazon EC2 Instance Metrics Discovery finished</td>
</tr>
<tr>
<td>4103</td>
<td>Amazon CloudWatch Metric Alarms Discovery finished</td>
</tr>
</tbody>
</table>
The following illustration shows the health model defined by the AWS Management Pack.

![Health Model Diagram]

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.

<table>
<thead>
<tr>
<th>Health State</th>
<th>AWS CloudFormation Stack Status</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>CREATE_FAILED</td>
<td>Most likely usable</td>
</tr>
<tr>
<td></td>
<td>DELETE_IN_PROGRESS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DELETE_FAILED</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UPDATE_ROLLBACK_FAILED</td>
<td></td>
</tr>
<tr>
<td>Warning</td>
<td>UPDATE_ROLLBACK_IN_PROGRESS</td>
<td>Recovering after some problem</td>
</tr>
<tr>
<td></td>
<td>UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UPDATE_ROLLBACK_COMPLETE</td>
<td></td>
</tr>
<tr>
<td>Healthy</td>
<td>CREATE_COMPLETE</td>
<td>Usable</td>
</tr>
<tr>
<td></td>
<td>UPDATE_IN_PROGRESS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UPDATE_COMPLETE_CLEANUP_IN_PROGRESS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UPDATE_COMPLETE</td>
<td></td>
</tr>
</tbody>
</table>

The full health model for an AWS CloudFormation stack is as follows:
Customize 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](https://technet.microsoft.com).

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](https://technet.microsoft.com).

Upgrade 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 Uninstall the AWS Management Pack (p. 1586).

**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 Uninstall the AWS Management Pack (p. 1586).

**Uninstall 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.

---

**Troubleshoot the AWS Management Pack**

The following are common errors, events, and troubleshooting steps.

**Contents**

- Errors 4101 and 4105 (p. 1587)
- Error 4513 (p. 1587)
- Event 623 (p. 1588)
- Events 2023 and 2120 (p. 1588)
- Event 6024 (p. 1588)
- General troubleshooting for System Center 2012 — Operations Manager (p. 1588)
- General troubleshooting for System Center 2007 R2 (p. 1589)

---

**Errors 4101 and 4105**

If you receive one of the following errors, you must upgrade the AWS Management Pack. For more information, see Upgrade the AWS Management Pack (p. 1585).

**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 Upgrade the AWS Management Pack (p. 1585).

**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 run.

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 SCOM 2012 - Event ID 6024 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: Install the AWS Management Pack (p. 1555).
• 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. 1582) 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: Install the AWS Management Pack (p. 1555).
• 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: Configure the watcher node (p. 1557).
  • 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. 1566).
• 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. 1582) for a list of the events that the AWS Management Pack writes to the Operations Manager event log.
Related information

The following related resources can help you as you work with this service.

Windows on AWS

- **Windows on AWS** – Overview of Windows on AWS workloads and services.
- **Amazon Web Services and Microsoft: Frequently Asked Questions** – Frequently asked questions specific to running Microsoft software on AWS.
- **Microsoft Licensing on AWS: Options for using Microsoft software licenses on the AWS Cloud** – Options for using Microsoft software licenses on the AWS Cloud.
- **AWS Migration Acceleration Program for Windows** – AWS services, best practices, and tools to help you save costs and accelerate migrations of Windows workloads to AWS.
- **AWS Optimization and Licensing Assessment** – Evaluate your Windows environment to reduce costs and optimize compute.
- **EC2 Image Builder** – automate the creation, management, and deployment of customized, secure, and up-to-date server images that are pre-installed and pre-configured with software settings to meet specific IT standards.
- **AWS Launch Wizard** – AWS Launch Wizard guides you through the sizing, configuration, and deployment of applications on AWS following the AWS Well-Architected Framework.
- **Microsoft SQL Server on AWS** – Overview of Microsoft SQL Server on AWS workloads and services.

Forum

**Amazon EC2 forum** – AWS Discussion forum for Amazon EC2 to post questions and feedback.

Pricing

**Amazon EC2 pricing** – Pricing information for Amazon EC2.

Tutorials

**Hands-on tutorials** – Get started with step-by-step tutorials to launch your first application.

General AWS resources

The following related resources can help you as you work with AWS.

- **Classes & Workshops** – Links to role-based and specialty courses, in addition to self-paced labs to help sharpen your AWS skills and gain practical experience.
- **AWS Developer Tools** – Links to developer tools, SDKs, IDE toolkits, and command line tools for developing and managing AWS applications.
- **AWS Whitepapers** – Links to a comprehensive list of technical AWS whitepapers, covering topics such as architecture, security, and economics and authored by AWS Solutions Architects or other technical experts.
- **AWS Support Center** – The hub for creating and managing your AWS Support cases. Also includes links to other helpful resources, such as forums, technical FAQs, service health status, and AWS Trusted Advisor.
- **AWS Support** – The primary webpage for information about AWS Support, a one-on-one, fast-response support channel to help you build and run applications in the cloud.
- **Contact Us** – A central contact point for inquiries concerning AWS billing, account, events, abuse, and other issues.
- **AWS Site Terms** – Detailed information about our copyright and trademark; your account, license, and site access; and other topics.
Document history

The following table describes important additions to the Amazon EC2 documentation starting in 2019. We also update the documentation frequently to address the feedback that you send us.

<table>
<thead>
<tr>
<th>update-history-change</th>
<th>update-history-description</th>
<th>update-history-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMI deprecation support for</td>
<td>Amazon Data Lifecycle Manager EBS-backed AMI policies can deprecate AMIs. The AWSDataLifecycleManagerServiceRoleForAMIManagement AWS managed policy has been updated to support this feature.</td>
<td>August 23, 2021</td>
</tr>
<tr>
<td>Amazon Data Lifecycle Manager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hibernation support for C5d,</td>
<td>Hibernate your newly-launched instances running on C5d, M5d, and R5d instance types.</td>
<td>August 19, 2021</td>
</tr>
<tr>
<td>M5d, and R5d</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amazon EC2 key pairs</td>
<td>Amazon EC2 now supports ED25519 keys on Linux and Mac instances.</td>
<td>August 17, 2021</td>
</tr>
<tr>
<td>M6i instances (p. 1592)</td>
<td>New general purpose instances featuring third generation Intel Xeon Scalable processors (Ice Lake).</td>
<td>August 16, 2021</td>
</tr>
<tr>
<td>CloudWatch metrics for Amazon</td>
<td>You can monitor your Amazon Data Lifecycle Manager policies using Amazon CloudWatch.</td>
<td>July 28, 2021</td>
</tr>
<tr>
<td>Data Lifecycle Manager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New Local Zone added</td>
<td>Add Local Zone in Denver.</td>
<td>July 27, 2021</td>
</tr>
<tr>
<td>CloudTrail data events for EBS</td>
<td>The ListSnapshotBlocks, ListChangedBlocks, GetSnapshotBlock, and PutSnapshotBlock APIs can be logged data events in CloudTrail.</td>
<td>July 27, 2021</td>
</tr>
<tr>
<td>direct APIs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prefixes for network interfaces</td>
<td>You can assign a private IPv4 or IPv6 CIDR range, either automatically or manually, to your network interfaces.</td>
<td>July 22, 2021</td>
</tr>
<tr>
<td>io2 Block Express volumes</td>
<td>io2 Block Express volumes are now generally available in all Regions and Availability Zones that support R5b instances.</td>
<td>July 19, 2021</td>
</tr>
<tr>
<td>Event windows</td>
<td>You can define custom, weekly-recurring event windows for scheduled events that reboot, stop, or terminate your Amazon EC2 instances.</td>
<td>July 15, 2021</td>
</tr>
</tbody>
</table>

1592
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource IDs and tagging support for security group rules (p. 1592)</td>
<td>You can refer to security group rules by resource ID. You can also add tags to your security group rules.</td>
<td>July 7, 2021</td>
</tr>
<tr>
<td>New Local Zones added</td>
<td>Add Local Zones in Dallas and Philadelphia.</td>
<td>July 7, 2021</td>
</tr>
<tr>
<td>Deprecate an AMI</td>
<td>You can now specify when an AMI is deprecated.</td>
<td>June 11, 2021</td>
</tr>
<tr>
<td>Windows per-second billing (p. 1592)</td>
<td>Amazon EC2 charges for Windows- and SQL Server-based usage by the second, with a one-minute minimum charge.</td>
<td>June 10, 2021</td>
</tr>
<tr>
<td>Capacity Reservations on AWS Outposts</td>
<td>You can now use Capacity Reservations on AWS Outposts.</td>
<td>May 24, 2021</td>
</tr>
<tr>
<td>Capacity Reservation sharing</td>
<td>You can now share Capacity Reservations created in Local Zones and Wavelength Zones.</td>
<td>May 24, 2021</td>
</tr>
<tr>
<td>High memory virtualized instances (p. 1592)</td>
<td>Virtualized high memory instances purpose-built to run large in-memory databases. The new types are u-6tb1.56xlarge, u-6tb1.112xlarge, u-9tb1.112xlarge, and u-12tb1.112xlarge.</td>
<td>May 11, 2021</td>
</tr>
<tr>
<td>Root volume replacement</td>
<td>You can now use root volume replacement tasks to replace the root EBS volume for running instances.</td>
<td>April 22, 2021</td>
</tr>
<tr>
<td>Store and restore an AMI using S3</td>
<td>Store EBS-backed AMIs in S3 and restore them from S3 to enable cross-partition copying of AMIs.</td>
<td>April 6, 2021</td>
</tr>
<tr>
<td>EC2 Serial Console</td>
<td>Troubleshoot boot and network connectivity issues by establishing a connection to the serial port of an instance.</td>
<td>March 30, 2021</td>
</tr>
<tr>
<td>Boot modes</td>
<td>Amazon EC2 now supports UEFI boot on selected AMD- and Intel-based EC2 instances.</td>
<td>March 22, 2021</td>
</tr>
<tr>
<td>Amazon EBS local snapshots on Outposts</td>
<td>You can now use Amazon EBS local snapshots on Outposts to store snapshots of volumes on an Outpost locally in Amazon S3 on the Outpost itself.</td>
<td>February 4, 2021</td>
</tr>
<tr>
<td>Create a reverse DNS record</td>
<td>You can now set up reverse DNS lookup for your Elastic IP addresses.</td>
<td>February 3, 2021</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Amazon Data Lifecycle Manager</td>
<td>Use Amazon Data Lifecycle Manager to automate the process of sharing snapshots and copying them across AWS accounts.</td>
<td>December 17, 2020</td>
</tr>
<tr>
<td>G4ad instances (p. 1592)</td>
<td>New instances powered by AMD Radeon Pro V520 GPUs and AMD 2nd Generation EPYC processors.</td>
<td>December 9, 2020</td>
</tr>
<tr>
<td>Tag AMIs and snapshots on AMI creation</td>
<td>When you create an AMI, you can tag the AMI and the snapshots with the same tags, or you can tag them with different tags.</td>
<td>December 4, 2020</td>
</tr>
<tr>
<td>io2 Block Express preview</td>
<td>You can opt in to the io2 Block Express volumes preview. io2 Block Express volumes provide sub-millisecond latency, and support higher IOPS, higher throughput, and larger capacity than io2 volumes.</td>
<td>December 1, 2020</td>
</tr>
<tr>
<td>gp3 volumes (p. 1592)</td>
<td>A new Amazon EBS General Purpose SSD volume type. You can specify provisioned IOPS and throughput when you create or modify the volume.</td>
<td>December 1, 2020</td>
</tr>
<tr>
<td>D3, D3en, M5zn, and R5b instances (p. 1592)</td>
<td>New instance types built on the Nitro System.</td>
<td>December 1, 2020</td>
</tr>
<tr>
<td>Throughput Optimized HDD and Cold HDD volume sizes</td>
<td>Throughput Optimized HDD (st1) and Cold HDD (sc1) volumes can range in size from 125 GiB to 16 TiB.</td>
<td>November 30, 2020</td>
</tr>
<tr>
<td>Use Amazon EventBridge to monitor Spot Fleet events</td>
<td>Create EventBridge rules that trigger programmatic actions in response to Spot Fleet state changes and errors.</td>
<td>November 20, 2020</td>
</tr>
<tr>
<td>Use Amazon EventBridge to monitor EC2 Fleet events</td>
<td>Create EventBridge rules that trigger programmatic actions in response to EC2 Fleet state changes and errors.</td>
<td>November 20, 2020</td>
</tr>
<tr>
<td>Delete instant fleets</td>
<td>Delete an EC2 Fleet of type instant and terminate all the instances in the fleet in a single API call.</td>
<td>November 18, 2020</td>
</tr>
<tr>
<td>Hibernation support for T3 and T3a</td>
<td>Hibernate your newly-launched instances running on T3 and T3a instance types.</td>
<td>November 17, 2020</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Amazon Data Lifecycle Manager</td>
<td>You can use Amazon Data Lifecycle Manager to automate the creation, retention, and deletion of EBS-backed AMIs.</td>
<td>November 9, 2020</td>
</tr>
<tr>
<td>Instance metadata category: events/recommendations/rebalance</td>
<td>The approximate time, in UTC, when the EC2 instance rebalance recommendation notification is emitted for the instance.</td>
<td>November 4, 2020</td>
</tr>
<tr>
<td>EC2 instance rebalance recommendation</td>
<td>A signal that notifies you when a Spot Instance is at elevated risk of interruption.</td>
<td>November 4, 2020</td>
</tr>
<tr>
<td>Capacity Reservations in Wavelength Zones</td>
<td>Capacity Reservations can now be created and used in Wavelength Zones.</td>
<td>November 4, 2020</td>
</tr>
<tr>
<td>Capacity Rebalancing</td>
<td>Configure Spot Fleet or EC2 Fleet to launch a replacement Spot Instance when Amazon EC2 emits a rebalance recommendation.</td>
<td>November 4, 2020</td>
</tr>
<tr>
<td>Hibernation support for I3, M5ad, and R5ad</td>
<td>Hibernate your newly-launched instances running on I3, M5ad, and R5ad instance types.</td>
<td>October 21, 2020</td>
</tr>
<tr>
<td>Spot Instance vCPU limits</td>
<td>Spot Instance limits are now managed in terms of the number of vCPUs that your running Spot Instances are either using or will use pending the fulfillment of open requests.</td>
<td>October 1, 2020</td>
</tr>
<tr>
<td>Capacity Reservations in Local Zones</td>
<td>Capacity Reservations can now be created and used in Local Zones.</td>
<td>September 30, 2020</td>
</tr>
<tr>
<td>Amazon Data Lifecycle Manager</td>
<td>Amazon Data Lifecycle Manager policies can be configured with up to four schedules.</td>
<td>September 17, 2020</td>
</tr>
<tr>
<td>Hibernation support for M5a and R5a</td>
<td>Hibernate your newly-launched instances running on M5a and R5a instance types.</td>
<td>August 28, 2020</td>
</tr>
<tr>
<td>Provisioned IOPS SSD (io2) volumes for Amazon EBS</td>
<td>Provisioned IOPS SSD (io2) volumes are designed to provide 99.999 percent volume durability with an AFR no higher than 0.001 percent.</td>
<td>August 24, 2020</td>
</tr>
<tr>
<td>Instance metadata provides instance location and placement information</td>
<td>New instance metadata fields under the <code>placement</code> category: Region, placement group name, partition number, host ID, and Availability Zone ID.</td>
<td>August 24, 2020</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td><strong>CSad instances (p. 1592)</strong></td>
<td>New compute optimized instances featuring second-generation AMD EYPC processors.</td>
<td>August 13, 2020</td>
</tr>
<tr>
<td><strong>Wavelength Zones</strong></td>
<td>A Wavelength Zone is an isolated zone in the carrier location where the Wavelength infrastructure is deployed.</td>
<td>August 6, 2020</td>
</tr>
<tr>
<td><strong>Capacity Reservation groups</strong></td>
<td>You can use AWS Resource Groups to create logical collections of Capacity Reservations, and then target instance launches into those groups.</td>
<td>July 29, 2020</td>
</tr>
<tr>
<td><strong>Fast snapshot restore</strong></td>
<td>You can enable fast snapshot restore for snapshots that are shared with you.</td>
<td>July 21, 2020</td>
</tr>
<tr>
<td><strong>EC2Launch v2 (p. 449)</strong></td>
<td>You can use EC2Launch v2 to perform tasks during instance startup, if an instance is stopped and later started, if an instance is restarted, and on demand. EC2Launch v2 supports all versions of Windows Server and replaces EC2Launch and EC2Config.</td>
<td>June 30, 2020</td>
</tr>
<tr>
<td><strong>Bare metal instances for G4dn (p. 1592)</strong></td>
<td>New instances that provide your applications with direct access to the physical resources of the host server.</td>
<td>June 5, 2020</td>
</tr>
<tr>
<td><strong>C5a instances (p. 1592)</strong></td>
<td>New compute optimized instances featuring second-generation AMD EYPC processors.</td>
<td>June 4, 2020</td>
</tr>
<tr>
<td><strong>Bring your own IPv6 addresses</strong></td>
<td>You can bring part or all of your IPv6 address range from your on-premises network to your AWS account.</td>
<td>May 21, 2020</td>
</tr>
<tr>
<td><strong>Launch instances using a Systems Manager parameter</strong></td>
<td>You can specify a AWS Systems Manager parameter instead of an AMI when you launch an instance.</td>
<td>May 5, 2020</td>
</tr>
<tr>
<td><strong>Customize scheduled event notifications</strong></td>
<td>You can customize scheduled event notifications to include tags in the email notification.</td>
<td>May 4, 2020</td>
</tr>
<tr>
<td><strong>Windows Server on Dedicated Hosts</strong></td>
<td>You can use Windows Server AMIs provided by Amazon to run the latest versions of Windows Server on Dedicated Hosts.</td>
<td>April 7, 2020</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Stop and start a Spot Instance</td>
<td>Stop your Spot Instances backed by Amazon EBS and start them at will, instead of relying on the stop interruption behavior.</td>
<td>January 13, 2020</td>
</tr>
<tr>
<td>Resource tagging (p. 1592)</td>
<td>You can tag egress-only internet gateways, local gateways, local gateway route tables, local gateway virtual interfaces, local gateway virtual interface groups, local gateway route table VPC associations, and local gateway route table virtual interface group associations.</td>
<td>January 10, 2020</td>
</tr>
<tr>
<td>Connect to your instance using Session Manager</td>
<td>You can start a Session Manager session with an instance from the Amazon EC2 console.</td>
<td>December 18, 2019</td>
</tr>
<tr>
<td>Dedicated Hosts and host resource groups</td>
<td>Dedicated Hosts can now be used with host resource groups.</td>
<td>December 2, 2019</td>
</tr>
<tr>
<td>Dedicated Host sharing</td>
<td>You can now share your Dedicated Hosts across AWS accounts.</td>
<td>December 2, 2019</td>
</tr>
<tr>
<td>Default credit specification at the account level</td>
<td>You can set the default credit specification per burstable performance instance family at the account level per AWS Region.</td>
<td>November 25, 2019</td>
</tr>
<tr>
<td>Instance type discovery</td>
<td>You can find an instance type that meets your needs.</td>
<td>November 22, 2019</td>
</tr>
<tr>
<td>Dedicated Hosts (p. 1592)</td>
<td>You can now configure a Dedicated Host to support multiple instance types in an instance family.</td>
<td>November 21, 2019</td>
</tr>
<tr>
<td>Amazon EBS fast snapshot restores</td>
<td>You can enable fast snapshot restores on an EBS snapshot to ensure that EBS volumes created from the snapshot are fully-initialized at creation and instantly deliver all of their provisioned performance.</td>
<td>November 20, 2019</td>
</tr>
<tr>
<td>Instance Metadata Service Version 2</td>
<td>You can use Instance Metadata Service Version 2, which is a session-oriented method for requesting instance metadata.</td>
<td>November 19, 2019</td>
</tr>
<tr>
<td>Hibernation support for On-Demand Windows instances</td>
<td>You can hibernate On-Demand Windows instances.</td>
<td>October 14, 2019</td>
</tr>
<tr>
<td>Queued purchases of Reserved Instances</td>
<td>You can queue the purchase of a Reserved Instance up to three years in advance.</td>
<td>October 4, 2019</td>
</tr>
<tr>
<td>Feature</td>
<td>Details</td>
<td>Date</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td><strong>G4dn instances (p. 1592)</strong></td>
<td>New instances featuring NVIDIA Tesla GPUs.</td>
<td>September 19, 2019</td>
</tr>
<tr>
<td><strong>Diagnostic interrupt</strong></td>
<td>You can send a diagnostic interrupt to an unreachable or unresponsive instance to trigger a blue screen/stop error.</td>
<td>August 14, 2019</td>
</tr>
<tr>
<td><strong>Capacity optimized allocation strategy</strong></td>
<td>Using EC2 Fleet or Spot Fleet, you can launch Spot Instances from Spot pools with optimal capacity for the number of instances that are launching.</td>
<td>August 12, 2019</td>
</tr>
<tr>
<td><strong>On-Demand Capacity Reservation sharing</strong></td>
<td>You can now share your Capacity Reservations across AWS accounts.</td>
<td>July 29, 2019</td>
</tr>
<tr>
<td><strong>Resource tagging (p. 1592)</strong></td>
<td>Launch templates on creation.</td>
<td>July 24, 2019</td>
</tr>
<tr>
<td><strong>Host recovery</strong></td>
<td>Automatically restart your instances on a new host in the event of an unexpected hardware failure on a Dedicated Host.</td>
<td>June 5, 2019</td>
</tr>
<tr>
<td><strong>Amazon EBS multi-volume snapshots</strong></td>
<td>You can take exact point-in-time, data coordinated, and crash-consistent snapshots across multiple EBS volumes attached to an EC2 instance.</td>
<td>May 29, 2019</td>
</tr>
<tr>
<td><strong>Resource tagging (p. 1592)</strong></td>
<td>You can tag Dedicated Host Reservations.</td>
<td>May 27, 2019</td>
</tr>
<tr>
<td><strong>Amazon EBS encryption by default</strong></td>
<td>After you enable encryption by default in a Region, all new EBS volumes you create in the Region are encrypted using the default KMS key for EBS encryption.</td>
<td>May 23, 2019</td>
</tr>
<tr>
<td><strong>VSS application-consistent snapshots</strong></td>
<td>Take application-consistent snapshots of all Amazon EBS volumes attached to your Windows instances using AWS Systems Manager Run Command.</td>
<td>May 13, 2019</td>
</tr>
<tr>
<td><strong>Resource tagging (p. 1592)</strong></td>
<td>You can tag VPC endpoints, endpoint services, and endpoint service configurations.</td>
<td>May 13, 2019</td>
</tr>
<tr>
<td><strong>Windows to Linux Replatforming Assistant for Microsoft SQL Server Databases</strong></td>
<td>Move existing Microsoft SQL Server workloads from a Windows to a Linux operating system.</td>
<td>May 8, 2019</td>
</tr>
</tbody>
</table>
### History for previous years

The following table describes important additions to the Amazon EC2 documentation in 2018 and earlier years.

<table>
<thead>
<tr>
<th>Feature</th>
<th>API version</th>
<th>Description</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition placement groups</td>
<td>2016-11-15</td>
<td>Partition placement groups spread instances across logical partitions, ensuring that instances in one partition do not share underlying hardware with instances in other partitions. For more information, see Partition placement groups (p. 976).</td>
<td>20 December 2018</td>
</tr>
<tr>
<td>p3dn.24xlarge instances</td>
<td>2016-11-15</td>
<td>New p3dn.24xlarge instances provide 100 Gbps of network bandwidth.</td>
<td>7 December 2018</td>
</tr>
<tr>
<td>Instances featuring 100 Gbps of network bandwidth</td>
<td>2016-11-15</td>
<td>New C5n instances can utilize up to 100 Gbps of network bandwidth.</td>
<td>26 November 2018</td>
</tr>
<tr>
<td>Spot console recommends a fleet of instances</td>
<td>2016-11-15</td>
<td>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 Create a Spot Fleet request (p. 725).</td>
<td>20 November 2018</td>
</tr>
<tr>
<td>New EC2 Fleet request type: instant</td>
<td>2016-11-15</td>
<td>EC2 Fleet now supports a new request type, instant, that you can use to synchronously provision capacity across instance types and</td>
<td>14 November 2018</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Instances featuring AMD EYPC processors</td>
<td>2016-11-15</td>
<td>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.</td>
<td>6 November 2018</td>
</tr>
<tr>
<td>Spot savings information</td>
<td>2016-11-15</td>
<td>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. 290).</td>
<td>5 November 2018</td>
</tr>
<tr>
<td>Console support for optimizing CPU options</td>
<td>2016-11-15</td>
<td>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 Optimize CPU options (p. 540).</td>
<td>31 October 2018</td>
</tr>
<tr>
<td>Console support for creating a launch template from an instance</td>
<td>2016-11-15</td>
<td>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 Create a launch template (p. 399).</td>
<td>30 October 2018</td>
</tr>
<tr>
<td>On-Demand Capacity Reservations</td>
<td>2016-11-15</td>
<td>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. 366).</td>
<td>25 October 2018</td>
</tr>
<tr>
<td>Bring Your Own IP Addresses (BYOIP)</td>
<td>2016-11-15</td>
<td>You can bring part or all of your public IPv4 address range from your on-premises network to your AWS account. After you bring the address range to AWS, it appears in your account as an address pool. You can create an Elastic IP address from your address pool and use it with your AWS resources. For more information, see Bring your own IP addresses (BYOIP) in Amazon EC2 (p. 908).</td>
<td>23 October 2018</td>
</tr>
<tr>
<td>g3s.xlarge instances</td>
<td>2016-11-15</td>
<td>Expands the range of the accelerated-computing G3 instance family with the introduction of g3s.xlarge instances.</td>
<td>11 October 2018</td>
</tr>
<tr>
<td>Dedicated Host tag on create and console support</td>
<td>2016-11-15</td>
<td>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 Allocate Dedicated Hosts (p. 333).</td>
<td>08 October 2018</td>
</tr>
</tbody>
</table>

1600
<table>
<thead>
<tr>
<th>Feature</th>
<th>API version</th>
<th>Description</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>High memory instances</td>
<td>2016-11-15</td>
<td>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. 199).</td>
<td>27 September 2018</td>
</tr>
<tr>
<td>f1.4xlarge instances</td>
<td>2016-11-15</td>
<td>Expands the range of the accelerated-computing F1 instance family with the introduction of f1.4xlarge instances.</td>
<td>25 September 2018</td>
</tr>
<tr>
<td>Console support for scheduled scaling for Spot Fleet</td>
<td>2016-11-15</td>
<td>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. 744).</td>
<td>20 September 2018</td>
</tr>
<tr>
<td>T3 instances</td>
<td>2016-11-15</td>
<td>T3 instances are 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. 160).</td>
<td>21 August 2018</td>
</tr>
<tr>
<td>Allocation strategies for EC2 Fleets</td>
<td>2016-11-15</td>
<td>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. 685).</td>
<td>26 July 2018</td>
</tr>
<tr>
<td>Allocation strategies for Spot Fleets</td>
<td>2016-11-15</td>
<td>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. 712).</td>
<td>26 July 2018</td>
</tr>
<tr>
<td>R5 and R5d instances</td>
<td>2016-11-15</td>
<td>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. 199).</td>
<td>25 July 2018</td>
</tr>
<tr>
<td>z1d instances</td>
<td>2016-11-15</td>
<td>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. 199).</td>
<td>25 July 2018</td>
</tr>
<tr>
<td>Automate snapshot lifecycle</td>
<td>2016-11-15</td>
<td>You can use Amazon Data Lifecycle Manager to automate creation and deletion of snapshots for your EBS volumes. For more information, see Amazon Data Lifecycle Manager (p. 1272).</td>
<td>12 July 2018</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Launch template CPU options</td>
<td>2016-11-15</td>
<td>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 Create a launch template (p. 399).</td>
<td>11 July 2018</td>
</tr>
<tr>
<td>Tag Dedicated Hosts</td>
<td>2016-11-15</td>
<td>You can tag your Dedicated Hosts. For more information, see Tag Dedicated Hosts (p. 344).</td>
<td>3 July 2018</td>
</tr>
<tr>
<td>i3.metal instances</td>
<td>2016-11-15</td>
<td>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. 210).</td>
<td>17 May 2018</td>
</tr>
<tr>
<td>Get latest console output</td>
<td>2016-11-15</td>
<td>You can retrieve the latest console output for some instance types when you use the get-console-output AWS CLI command.</td>
<td>9 May 2018</td>
</tr>
<tr>
<td>Optimize CPU options</td>
<td>2016-11-15</td>
<td>When you launch an instance, you can optimize the CPU options to suit specific workloads or business needs. For more information, see Optimize CPU options (p. 540).</td>
<td>8 May 2018</td>
</tr>
<tr>
<td>EC2 Fleet</td>
<td>2016-11-15</td>
<td>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 EC2 Fleet (p. 664).</td>
<td>2 May 2018</td>
</tr>
<tr>
<td>On-Demand Instances in Spot Fleets</td>
<td>2016-11-15</td>
<td>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 Spot Fleet (p. 711).</td>
<td>2 May 2018</td>
</tr>
<tr>
<td>Tag EBS snapshots on creation</td>
<td>2016-11-15</td>
<td>You can apply tags to snapshots during creation. For more information, see Create Amazon EBS snapshots (p. 1211).</td>
<td>2 April 2018</td>
</tr>
<tr>
<td>Change placement groups</td>
<td>2016-11-15</td>
<td>You can move an instance in or out of a placement group, or change its placement group. For more information, see Change the placement group for an instance (p. 985).</td>
<td>1 March 2018</td>
</tr>
<tr>
<td>Longer resource IDs</td>
<td>2016-11-15</td>
<td>You can enable the longer ID format for more resource types. For more information, see Resource IDs (p. 1443).</td>
<td>9 February 2018</td>
</tr>
<tr>
<td>Network performance improvements</td>
<td>2016-11-15</td>
<td>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. 147).</td>
<td>24 January 2018</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Tag Elastic IP addresses</td>
<td>2016-11-15</td>
<td>You can tag your Elastic IP addresses. For more information, see [Tag an Elastic IP address](p. 929).</td>
<td>21 December 2017</td>
</tr>
<tr>
<td>Amazon Time Sync Service</td>
<td>2016-11-15</td>
<td>You can use the Amazon Time Sync Service to keep accurate time on your instance. For more information, see [Set the time for a Windows instance](p. 559).</td>
<td>29 November 2017</td>
</tr>
<tr>
<td>T2 Unlimited</td>
<td>2016-11-15</td>
<td>T2 Unlimited instances can burst above the baseline for as long as required. For more information, see [Burstable performance instances](p. 160).</td>
<td>29 November 2017</td>
</tr>
<tr>
<td>Launch templates</td>
<td>2016-11-15</td>
<td>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 [Launch an instance from a launch template](p. 397).</td>
<td>29 November 2017</td>
</tr>
<tr>
<td>Spread placement</td>
<td>2016-11-15</td>
<td>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. 977).</td>
<td>29 November 2017</td>
</tr>
<tr>
<td>H1 instances</td>
<td>2016-11-15</td>
<td>H1 instances are designed for high-performance big data workloads. For more information, see [Storage optimized instances](p. 210).</td>
<td>28 November 2017</td>
</tr>
<tr>
<td>M5 instances</td>
<td>2016-11-15</td>
<td>M5 instances are general purpose compute instances. They provide a balance of compute, memory, storage, and network resources.</td>
<td>28 November 2017</td>
</tr>
<tr>
<td>Spot Instance hibernation</td>
<td>2016-11-15</td>
<td>The Spot service can hibernate Spot Instances in the event of an interruption. For more information, see [Hibernate interrupted Spot Instances](p. 319).</td>
<td>28 November 2017</td>
</tr>
<tr>
<td>Spot Fleet target tracking</td>
<td>2016-11-15</td>
<td>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. 742).</td>
<td>17 November 2017</td>
</tr>
<tr>
<td>Spot Fleet integrates with Elastic Load Balancing</td>
<td>2016-11-15</td>
<td>You can attach one or more load balancers to a Spot Fleet.</td>
<td>10 November 2017</td>
</tr>
<tr>
<td>X1e instances</td>
<td>2016-11-15</td>
<td>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. 199).</td>
<td>28 November 2017</td>
</tr>
<tr>
<td>C5 instances</td>
<td>2016-11-15</td>
<td>C5 instances are designed for compute-heavy applications. For more information, see [Compute optimized instances](p. 193).</td>
<td>6 November 2017</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Merge and split Convertible Reserved Instances</td>
<td>2016-11-15</td>
<td>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 Exchange Convertible Reserved Instances (p. 277).</td>
<td>6 November 2017</td>
</tr>
<tr>
<td>P3 instances</td>
<td>2016-11-15</td>
<td>P3 instances are compute-optimized GPU instances. For more information, see Windows accelerated computing instances (p. 216).</td>
<td>25 October 2017</td>
</tr>
<tr>
<td>Modify VPC tenancy</td>
<td>2016-11-15</td>
<td>You can change the instance tenancy attribute of a VPC from dedicated to default. For more information, see Change the tenancy of a VPC (p. 365).</td>
<td>16 October 2017</td>
</tr>
<tr>
<td>Stop on interruption</td>
<td>2016-11-15</td>
<td>You can specify whether Amazon EC2 should stop or terminate Spot Instances when they are interrupted. For more information, see Interruption behaviors (p. 318).</td>
<td>18 September 2017</td>
</tr>
<tr>
<td>Tag NAT gateways</td>
<td>2016-11-15</td>
<td>You can tag your NAT gateway. For more information, see Tag your resources (p. 1451).</td>
<td>7 September 2017</td>
</tr>
<tr>
<td>Security group rule descriptions</td>
<td>2016-11-15</td>
<td>You can add descriptions to your security group rules. For more information, see Security group rules (p. 1136).</td>
<td>31 August 2017</td>
</tr>
<tr>
<td>Elastic Graphics</td>
<td>2016-11-15</td>
<td>Attach Elastic Graphics accelerators to your instances to accelerate the graphics performance of your applications. For more information, see Amazon Elastic Graphics (p. 793).</td>
<td>29 August 2017</td>
</tr>
<tr>
<td>Recover Elastic IP addresses</td>
<td>2016-11-15</td>
<td>If you release an Elastic IP address for use in a VPC, you might be able to recover it. For more information, see Recover an Elastic IP address (p. 933).</td>
<td>11 August 2017</td>
</tr>
<tr>
<td>Tag Spot Fleet instances</td>
<td>2016-11-15</td>
<td>You can configure your Spot Fleet to automatically tag the instances that it launches.</td>
<td>24 July 2017</td>
</tr>
<tr>
<td>G3 instances</td>
<td>2016-11-15</td>
<td>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. 216).</td>
<td>13 July 2017</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Tag resources during creation</td>
<td>2016-11-15</td>
<td>You can apply tags to instances and volumes during creation. For more information, see [Tag your resources](p. 1451). In addition, you can use tag-based resource-level permissions to control the tags that are applied. For more information see, [Grant permission to tag resources during creation](p. 1065).</td>
<td>28 March 2017</td>
</tr>
<tr>
<td>I3 instances</td>
<td>2016-11-15</td>
<td>I3 instances are storage optimized instances. For more information, see [Storage optimized instances](p. 210).</td>
<td>23 February 2017</td>
</tr>
<tr>
<td>Perform modifications on attached EBS volumes</td>
<td>2016-11-15</td>
<td>With most EBS volumes attached to most EC2 instances, you can modify volume size, type, and IOPS without detaching the volume or stopping the instance. For more information, see [Amazon EBS Elastic Volumes](p. 1315).</td>
<td>13 February 2017</td>
</tr>
<tr>
<td>Attach an IAM role</td>
<td>2016-11-15</td>
<td>You can attach, detach, or replace an IAM role for an existing instance. For more information, see [IAM roles for Amazon EC2](p. 1114).</td>
<td>9 February 2017</td>
</tr>
<tr>
<td>Dedicated Spot Instances</td>
<td>2016-11-15</td>
<td>You can run Spot Instances on single-tenant hardware in a virtual private cloud (VPC). For more information, see [Specify a tenancy for your Spot Instances](p. 293).</td>
<td>19 January 2017</td>
</tr>
<tr>
<td>IPv6 support</td>
<td>2016-11-15</td>
<td>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. 893).</td>
<td>1 December 2016</td>
</tr>
<tr>
<td>R4 instances</td>
<td>2016-09-15</td>
<td>R4 instances are 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. 199)</td>
<td>30 November 2016</td>
</tr>
<tr>
<td>New t2.xlarge and t2.2xlarge instance types</td>
<td>2016-09-15</td>
<td>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. 160).</td>
<td>30 November 2016</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>P2 instances</td>
<td>2016-09-15</td>
<td>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. 216).</td>
<td>29 September 2016</td>
</tr>
<tr>
<td>m4.16xlarge instances</td>
<td>2016-04-01</td>
<td>Expands the range of the general-purpose M4 family with the introduction of m4.16xlarge instances, with 64 vCPUs and 256 GiB of RAM.</td>
<td>6 September 2016</td>
</tr>
<tr>
<td>Automatic scaling for Spot Fleet</td>
<td></td>
<td>You can now set up scaling policies for your Spot Fleet. For more information, see Automatic scaling for Spot Fleet (p. 740).</td>
<td>1 September 2016</td>
</tr>
<tr>
<td>Elastic Network Adapter (ENA)</td>
<td>2016-04-01</td>
<td>You can now use ENA for enhanced networking. For more information, see Enhanced networking support (p. 961).</td>
<td>28 June 2016</td>
</tr>
<tr>
<td>Enhanced support for viewing and modifying longer IDs</td>
<td>2016-04-01</td>
<td>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. 1443).</td>
<td>23 June 2016</td>
</tr>
<tr>
<td>Copy encrypted Amazon EBS snapshots between AWS accounts</td>
<td>2016-04-01</td>
<td>You can now copy encrypted EBS snapshots between AWS accounts. For more information, see Copy an Amazon EBS snapshot (p. 1229).</td>
<td>21 June 2016</td>
</tr>
<tr>
<td>Capture a screenshot of an instance console</td>
<td>2015-10-01</td>
<td>You can now obtain additional information when debugging instances that are unreachable. For more information, see Troubleshoot an unreachable instance (p. 1475).</td>
<td>24 May 2016</td>
</tr>
<tr>
<td>X1 instances</td>
<td>2015-10-01</td>
<td>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. 199).</td>
<td>18 May 2016</td>
</tr>
<tr>
<td>Two new EBS volume types</td>
<td>2015-10-01</td>
<td>You can now create Throughput Optimized HDD (st1) and Cold HDD (sc1) volumes. For more information, see Amazon EBS volume types (p. 1163).</td>
<td>19 April 2016</td>
</tr>
<tr>
<td>Added new NetworkPacketsIn and NetworkPacketsOut metrics for Amazon EC2</td>
<td></td>
<td>Added new NetworkPacketsIn and NetworkPacketsOut metrics for Amazon EC2. For more information, see Instance metrics (p. 842).</td>
<td>23 March 2016</td>
</tr>
<tr>
<td>CloudWatch metrics for Spot Fleet</td>
<td></td>
<td>You can now get CloudWatch metrics for your Spot Fleet. For more information, see CloudWatch metrics for Spot Fleet (p. 737).</td>
<td>21 March 2016</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Scheduled Instances</td>
<td>2015-10-01</td>
<td>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. 281).</td>
<td>13 January 2016</td>
</tr>
<tr>
<td>Longer resource IDs</td>
<td>2015-10-01</td>
<td>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. 1443).</td>
<td>13 January 2016</td>
</tr>
<tr>
<td>ClassicLink DNS support</td>
<td>2015-10-01</td>
<td>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 Enable ClassicLink DNS support (p. 1036).</td>
<td>11 January 2016</td>
</tr>
<tr>
<td>New t2.nano instance type</td>
<td>2015-10-01</td>
<td>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. 160).</td>
<td>15 December 2015</td>
</tr>
<tr>
<td>Dedicated hosts</td>
<td>2015-10-01</td>
<td>An Amazon EC2 Dedicated host is a physical server with instance capacity dedicated for your use. For more information, see Dedicated Hosts (p. 329).</td>
<td>23 November 2015</td>
</tr>
<tr>
<td>Spot Instance duration</td>
<td>2015-10-01</td>
<td>You can now specify a duration for your Spot Instances. For more information, see Define a duration for your Spot Instances (p. 293).</td>
<td>6 October 2015</td>
</tr>
<tr>
<td>Spot Fleet modify request</td>
<td>2015-10-01</td>
<td>You can now modify the target capacity of your Spot Fleet request. For more information, see Modify a Spot Fleet request (p. 735).</td>
<td>29 September 2015</td>
</tr>
<tr>
<td>Spot Fleet diversified allocation strategy</td>
<td>2015-04-15</td>
<td>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. 712).</td>
<td>15 September 2015</td>
</tr>
<tr>
<td>Spot Fleet instance weighting</td>
<td>2015-04-15</td>
<td>You can now define the capacity units that each instance type contributes to your application's performance, and adjust the amount you are willing to pay for Spot Instances for each Spot pool accordingly. For more information, see Spot Fleet instance weighting (p. 717).</td>
<td>31 August 2015</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>New reboot alarm action and new IAM role for use with alarm actions</td>
<td></td>
<td>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. 863).</td>
<td>23 July 2015</td>
</tr>
<tr>
<td>New t2.large instance type</td>
<td></td>
<td>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. 160).</td>
<td>16 June 2015</td>
</tr>
<tr>
<td>M4 instances</td>
<td></td>
<td>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.</td>
<td>11 June 2015</td>
</tr>
<tr>
<td>Spot Fleets</td>
<td>2015-04-15</td>
<td>You can manage a collection, or fleet, of Spot Instances instead of managing separate Spot Instance requests. For more information, see Spot Fleet (p. 711).</td>
<td>18 May 2015</td>
</tr>
<tr>
<td>Migrate Elastic IP addresses to EC2-Classic</td>
<td>2015-04-15</td>
<td>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 Migrate an Elastic IP Address from EC2-Classic (p. 1027).</td>
<td>15 May 2015</td>
</tr>
<tr>
<td>Importing VMs with multiple disks as AMIs</td>
<td>2015-03-01</td>
<td>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 VM Import/Export User Guide.</td>
<td>23 April 2015</td>
</tr>
<tr>
<td>New g2.8xlarge instance type</td>
<td></td>
<td>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.</td>
<td>7 April 2015</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>D2 instances</td>
<td></td>
<td>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. For more information, see Storage optimized instances (p. 210).</td>
<td>24 March 2015</td>
</tr>
<tr>
<td>Systems Manager</td>
<td></td>
<td>Systems Manager enables you to configure and manage your EC2 instances.</td>
<td>17 February 2015</td>
</tr>
<tr>
<td>Systems Manager for Microsoft SCVMM 1.5</td>
<td></td>
<td>You can now use Systems Manager for Microsoft SCVMM to launch an instance and to import a VM from SCVMM to Amazon EC2. For more information, see Create an EC2 Instance (p. 1542) and Import your virtual machine (p. 1547).</td>
<td>21 January 2015</td>
</tr>
<tr>
<td>Automatic recovery for EC2 instances</td>
<td></td>
<td>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. For more information, see Recover your instance (p. 447).</td>
<td>12 January 2015</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>C4 instances</td>
<td></td>
<td>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. For more information, see Compute optimized instances (p. 193).</td>
<td>11 January 2015</td>
</tr>
<tr>
<td>ClassicLink</td>
<td>2014-10-01</td>
<td>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. 1030).</td>
<td>7 January 2015</td>
</tr>
<tr>
<td>Spot Instance termination notices</td>
<td></td>
<td>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. For more information, see Spot Instance interruption notices (p. 322).</td>
<td>5 January 2015</td>
</tr>
<tr>
<td>Systems Manager for Microsoft SCVMM</td>
<td></td>
<td>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. 1537).</td>
<td>29 October 2014</td>
</tr>
<tr>
<td>DescribeVolumes pagination support</td>
<td>2014-09-01</td>
<td>The DescribeVolumes API call now supports the pagination of results with the MaxResults and NextToken parameters. For more information, see DescribeVolumes in the Amazon EC2 API Reference.</td>
<td>23 October 2014</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Added support for Amazon CloudWatch Logs</td>
<td></td>
<td>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.</td>
<td>10 July 2014</td>
</tr>
<tr>
<td>T2 instances</td>
<td>2014-06-15</td>
<td>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. 160).</td>
<td>30 June 2014</td>
</tr>
<tr>
<td>New EC2 Service Limits page</td>
<td></td>
<td>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.</td>
<td>19 June 2014</td>
</tr>
<tr>
<td>Amazon EBS General Purpose SSD Volumes</td>
<td>2014-05-01</td>
<td>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 volumes (gp2) (p. 1166).</td>
<td>16 June 2014</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td></td>
<td>AMIs for Windows Server 2012 R2 use the new AWS PV drivers. For more information, see AWS PV drivers (p. 520).</td>
<td>3 June 2014</td>
</tr>
<tr>
<td>AWS Management Pack</td>
<td></td>
<td>AWS Management Pack now supports for System Center Operations Manager 2012 R2. For more information, see AWS Management Pack for Microsoft System Center (p. 1551).</td>
<td>22 May 2014</td>
</tr>
<tr>
<td>Amazon EBS encryption</td>
<td>2014-05-01</td>
<td>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 AWS 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. 1327).</td>
<td>21 May 2014</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>R3 instances</td>
<td>2014-02-01</td>
<td>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.</td>
<td>9 April 2014</td>
</tr>
<tr>
<td>Amazon EC2 Usage Reports</td>
<td></td>
<td>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. 1464).</td>
<td>28 January 2014</td>
</tr>
<tr>
<td>Additional M3 instances</td>
<td>2013-10-15</td>
<td>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.</td>
<td>20 January 2014</td>
</tr>
<tr>
<td>I2 instances</td>
<td>2013-10-15</td>
<td>These instances provide very high IOPS. I2 instances also support enhanced networking that delivers improve inter-instance latencies, lower network jitter, and significantly higher packet per second (PPS) performance. For more information, see Storage optimized instances (p. 210).</td>
<td>19 December 2013</td>
</tr>
<tr>
<td>Updated M3 instances</td>
<td>2013-10-15</td>
<td>The M3 instance sizes <code>m3.xlarge</code> and <code>m3.2xlarge</code> now support instance store with SSD volumes.</td>
<td>19 December 2013</td>
</tr>
<tr>
<td>Resource-level permissions for RunInstances</td>
<td>2013-10-15</td>
<td>You can now create policies in AWS Identity and Access Management to control resource-level permissions for the Amazon EC2 RunInstances API action. For more information and example policies, see Identity and access management for Amazon EC2 (p. 1058).</td>
<td>20 November 2013</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>C3 instances</td>
<td>2013-10-15</td>
<td>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. For more information about the hardware specifications for each Amazon EC2 instance type, see Amazon EC2 Instance Types.</td>
<td>14 November 2013</td>
</tr>
<tr>
<td>Launching an instance from the AWS Marketplace</td>
<td></td>
<td>You can now launch an instance from the AWS Marketplace using the Amazon EC2 launch wizard. For more information, see Launch an AWS Marketplace instance (p. 412).</td>
<td>11 November 2013</td>
</tr>
<tr>
<td>G2 instances</td>
<td>2013-10-01</td>
<td>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. 216).</td>
<td>4 November 2013</td>
</tr>
<tr>
<td>New launch wizard</td>
<td></td>
<td>There is a new and redesigned EC2 launch wizard. For more information, see Launch an instance using the Launch Instance Wizard (p. 392).</td>
<td>10 October 2013</td>
</tr>
<tr>
<td>Modifying Amazon EC2 Reserved Instances</td>
<td>2013-08-15</td>
<td>You can now modify Reserved Instances in a Region.</td>
<td>11 September 2013</td>
</tr>
<tr>
<td>Assigning a public IP address</td>
<td>2013-07-15</td>
<td>You can now assign a public IP address when you launch an instance in a VPC. For more information, see Assign a public IPv4 address during instance launch (p. 898).</td>
<td>20 August 2013</td>
</tr>
<tr>
<td>Granting resource-level permissions</td>
<td>2013-06-15</td>
<td>Amazon EC2 supports new Amazon Resource Names (ARNs) and condition keys. For more information, see IAM policies for Amazon EC2 (p. 1060).</td>
<td>8 July 2013</td>
</tr>
<tr>
<td>Incremental Snapshot Copies</td>
<td>2013-02-01</td>
<td>You can now perform incremental snapshot copies. For more information, see Copy an Amazon EBS snapshot (p. 1229).</td>
<td>11 June 2013</td>
</tr>
</tbody>
</table>
## History for previous years

<table>
<thead>
<tr>
<th>Feature</th>
<th>API version</th>
<th>Description</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Management Pack</td>
<td></td>
<td>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. 1551).</td>
<td>8 May 2013</td>
</tr>
<tr>
<td>New Tags page</td>
<td></td>
<td>There is a new Tags page in the Amazon EC2 console. For more information, see [Tag your Amazon EC2 resources](p. 1450).</td>
<td>04 April 2013</td>
</tr>
<tr>
<td>Additional EBS-optimized instance types</td>
<td>2013-02-01</td>
<td>The following instance types can now be launched as EBS-optimized instances: c1.xlarge, m2.2xlarge, m3.xlarge, and m3.2xlarge. For more information, see [Amazon EBS-optimized instances](p. 1344).</td>
<td>19 March 2013</td>
</tr>
<tr>
<td>PV Drivers</td>
<td></td>
<td>To learn how to upgrade the paravirtualized (PV) drivers on your Windows AMI, see [Upgrade PV drivers on Windows instances](p. 524).</td>
<td>March 2013</td>
</tr>
<tr>
<td>Copy an AMI from one Region to another</td>
<td>2013-02-01</td>
<td>You can copy an AMI from one Region to another, enabling you to launch consistent instances in more than one AWS Region quickly and easily. For more information, see [Copy an AMI](p. 113).</td>
<td>11 March 2013</td>
</tr>
<tr>
<td>Launch instances into a default VPC</td>
<td>2013-02-01</td>
<td>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.</td>
<td>11 March 2013</td>
</tr>
<tr>
<td>High-memory cluster (cr1.8xlarge) instance type</td>
<td>2012-12-01</td>
<td>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.</td>
<td>21 January 2013</td>
</tr>
<tr>
<td>High storage (hs1.8xlarge) instance type</td>
<td>2012-12-01</td>
<td>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.</td>
<td>20 December 2012</td>
</tr>
<tr>
<td>EBS snapshot copy</td>
<td>2012-12-01</td>
<td>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 [Copy an Amazon EBS snapshot](p. 1229).</td>
<td>17 December 2012</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Updated EBS metrics and status checks for Provisioned IOPS SSD volumes</td>
<td>2012-10-01</td>
<td>Updated the EBS metrics to include two new metrics for Provisioned IOPS SSD volumes. For more information, see Amazon CloudWatch metrics for Amazon EBS (p. 1375). Also added new status checks for Provisioned IOPS SSD volumes. For more information, see EBS volume status checks (p. 1197).</td>
<td>20 November 2012</td>
</tr>
<tr>
<td>Support for Windows Server 2012</td>
<td></td>
<td>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: • 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</td>
<td>19 November 2012</td>
</tr>
<tr>
<td>M3 instances</td>
<td>2012-10-01</td>
<td>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.</td>
<td>31 October 2012</td>
</tr>
<tr>
<td>Spot Instance request status</td>
<td>2012-10-01</td>
<td>Spot Instance request status makes it easy to determine the state of your Spot requests.</td>
<td>14 October 2012</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Amazon EC2 Reserved Instance Marketplace</td>
<td>2012-08-15</td>
<td>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.</td>
<td>11 September 2012</td>
</tr>
<tr>
<td>Provisioned IOPS SSD for Amazon EBS</td>
<td>2012-07-20</td>
<td>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. 1163).</td>
<td>31 July 2012</td>
</tr>
<tr>
<td>High I/O instances for Amazon EC2</td>
<td>2012-06-15</td>
<td>High I/O instances provides very high, low latency, disk I/O performance using SSD-based local instance storage.</td>
<td>18 July 2012</td>
</tr>
<tr>
<td>IAM roles on Amazon EC2 instances</td>
<td>2012-06-01</td>
<td>IAM roles for Amazon EC2 provide:</td>
<td>11 June 2012</td>
</tr>
</tbody>
</table>
| • 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. | | |
| Spot Instance features that make it easier to get started and handle the potential of interruption. | 2012-05-01  | You can now manage your Spot Instances as follows:                                                                                             | 7 June 2012         |
| • Specify the amount you are willing to pay for Spot Instances using Auto Scaling launch configurations, and set up a schedule for specifying the amount you are willing to pay for Spot Instances. For more information, see Launching Spot Instances in Your Auto Scaling Group in the Amazon EC2 Auto Scaling User Guide.  
| • Get notifications when instances are launched or terminated.  
<p>| • Use AWS CloudFormation templates to launch Spot Instances in a stack with AWS resources. | | |
| 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.                                                                                                              | 25 May 2012         |
| | | Added support for timestamps on instance status and system status to indicate the date and time that a status check failed. | |</p>
<table>
<thead>
<tr>
<th>Feature</th>
<th>API version</th>
<th>Description</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC2 instance export, and timestamps in instance and system status checks for Amazon VPC</td>
<td>2012-05-01</td>
<td>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.</td>
<td>25 May 2012</td>
</tr>
<tr>
<td>Cluster Compute Eight Extra Large instances</td>
<td>2012-04-01</td>
<td>Added support for <code>cc2.8xlarge</code> instances in a VPC.</td>
<td>26 April 2012</td>
</tr>
<tr>
<td>AWS Marketplace AMIs</td>
<td>2012-04-01</td>
<td>Added support for AWS Marketplace AMIs.</td>
<td>19 April 2012</td>
</tr>
<tr>
<td>Medium instances, support for 64-bit on all AMIs</td>
<td>2011-12-15</td>
<td>Added support for a new instance type and 64-bit information.</td>
<td>7 March 2012</td>
</tr>
<tr>
<td>Reserved Instance pricing tiers</td>
<td>2011-12-15</td>
<td>Added a new section discussing how to take advantage of the discount pricing that is built into the Reserved Instance pricing tiers.</td>
<td>5 March 2012</td>
</tr>
<tr>
<td>Elastic Network Interfaces (ENIs) for EC2 instances in Amazon Virtual Private Cloud</td>
<td>2011-12-01</td>
<td>Added new section about elastic network interfaces (ENIs) for EC2 instances in a VPC. For more information, see <a href="https://example.com">Elastic network interfaces (p. 934)</a>.</td>
<td>21 December 2011</td>
</tr>
<tr>
<td>New offering types for Amazon EC2 Reserved Instances</td>
<td>2011-11-01</td>
<td>You can choose from a variety of Reserved Instance offerings that address your projected use of the instance.</td>
<td>16 November 2011</td>
</tr>
<tr>
<td>Amazon EC2 instance status</td>
<td>2011-11-01</td>
<td>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 Monitor the status of your instances (p. 809).</td>
<td>16 November 2011</td>
</tr>
<tr>
<td>Amazon EC2 Cluster Compute Instance Type</td>
<td></td>
<td>Added support for Cluster Compute Eight Extra Large (<code>cc2.8xlarge</code>) to Amazon EC2.</td>
<td>14 November 2011</td>
</tr>
<tr>
<td>Spot Instances in Amazon VPC</td>
<td>2011-07-15</td>
<td>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.</td>
<td>11 October 2011</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Simplified VM import process for users of the CLI tools</td>
<td>2011-07-15</td>
<td>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.</td>
<td>15 September 2011</td>
</tr>
<tr>
<td>Support for importing in VHD file format</td>
<td></td>
<td>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.</td>
<td>24 August 2011</td>
</tr>
<tr>
<td>Support for Windows Server 2003 R2</td>
<td></td>
<td>VM Import now supports Windows Server 2003 (R2). With this release, VM Import supports all versions of Windows Server supported by Amazon EC2.</td>
<td>24 August 2011</td>
</tr>
<tr>
<td>Update to the Amazon EC2 VM Import Connector for VMware vCenter</td>
<td></td>
<td>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.</td>
<td>27 June 2011</td>
</tr>
<tr>
<td>Spot Instances Availability Zone pricing changes</td>
<td>2011-05-15</td>
<td>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.</td>
<td>26 May 2011</td>
</tr>
<tr>
<td>AWS Identity and Access Management</td>
<td></td>
<td>Added information about AWS Identity and Access Management (IAM), which enables users to specify which Amazon EC2 actions a user can use with Amazon EC2 resources in general. For more information, see Identity and access management for Amazon EC2 (p. 1058).</td>
<td>26 April 2011</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Dedicated instances</td>
<td></td>
<td>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. 360).</td>
<td>27 March 2011</td>
</tr>
<tr>
<td>Reserved Instances updates to the AWS Management Console</td>
<td></td>
<td>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.</td>
<td>27 March 2011</td>
</tr>
<tr>
<td>Support for Windows Server 2008 R2</td>
<td></td>
<td>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.</td>
<td>15 March 2011</td>
</tr>
<tr>
<td>Metadata information</td>
<td>2011-01-01</td>
<td>Added information about metadata to reflect changes in the 2011-01-01 release. For more information, see Instance metadata and user data (p. 579) and Instance metadata categories (p. 596).</td>
<td>11 March 2011</td>
</tr>
<tr>
<td>Amazon EC2 VM Import Connector for VMware vCenter</td>
<td></td>
<td>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.</td>
<td>3 March 2011</td>
</tr>
<tr>
<td>Force volume detachment</td>
<td></td>
<td>You can now use the AWS Management Console to force the detachment of an Amazon EBS volume from an instance. For more information, see Detach an Amazon EBS volume from a Windows instance (p. 1204).</td>
<td>23 February 2011</td>
</tr>
<tr>
<td>Instance termination protection</td>
<td></td>
<td>You can now use the AWS Management Console to prevent an instance from being terminated. For more information, see Enable termination protection (p. 443).</td>
<td>23 February 2011</td>
</tr>
<tr>
<td>VM Import</td>
<td>2010-11-15</td>
<td>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.</td>
<td>15 December 2010</td>
</tr>
<tr>
<td>Feature</td>
<td>API version</td>
<td>Description</td>
<td>Release date</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Basic monitoring for instances</td>
<td>2010-08-31</td>
<td>Added information about basic monitoring for EC2 instances.</td>
<td>12 December 2010</td>
</tr>
<tr>
<td>Filters and Tags</td>
<td>2010-08-31</td>
<td>Added information about listing, filtering, and tagging resources. For more information, see List and filter your resources (p. 1444) and Tag your Amazon EC2 resources (p. 1450).</td>
<td>19 September 2010</td>
</tr>
<tr>
<td>Idempotent Instance Launch</td>
<td>2010-08-31</td>
<td>Added information about ensuring idempotency when running instances.</td>
<td>19 September 2010</td>
</tr>
<tr>
<td>Micro instances</td>
<td>2010-06-15</td>
<td>Amazon EC2 offers the t1.micro instance type for certain types of applications. For more information, see Burstable performance instances (p. 160).</td>
<td>8 September 2010</td>
</tr>
<tr>
<td>AWS Identity and Access Management for Amazon EC2</td>
<td></td>
<td>Amazon EC2 now integrates with AWS Identity and Access Management (IAM). For more information, see Identity and access management for Amazon EC2 (p. 1058).</td>
<td>2 September 2010</td>
</tr>
<tr>
<td>Cluster instances</td>
<td>2010-06-15</td>
<td>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.</td>
<td>12 July 2010</td>
</tr>
<tr>
<td>Amazon VPC IP Address Designation</td>
<td>2010-06-15</td>
<td>Amazon VPC users can now specify the IP address to assign an instance launched in a VPC.</td>
<td>12 July 2010</td>
</tr>
<tr>
<td>Amazon CloudWatch monitoring for Amazon EBS Volumes</td>
<td></td>
<td>Amazon CloudWatch monitoring is now automatically available for Amazon EBS volumes. For more information, see Amazon CloudWatch metrics for Amazon EBS (p. 1375).</td>
<td>14 June 2010</td>
</tr>
<tr>
<td>High-memory extra large instances</td>
<td>2009-11-30</td>
<td>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.</td>
<td>22 February 2010</td>
</tr>
<tr>
<td>Reserved Instances with Windows</td>
<td></td>
<td>Amazon EC2 now supports Reserved Instances with Windows.</td>
<td>22 February 2010</td>
</tr>
</tbody>
</table>